Piecewise Truckload Network Procurement

by

Jefferson Huang

B.S. Civil and Environmental Engineering, University of California, Berkeley (2008)

Submitted to the Department of Civil and Environmental Engineering in partial fulfillment of the requirements for the degree of

Master of Science in Transportation

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2011

Author ....
Department of Civil and Environmental Engineering
August 5, 2011

Certified by ....
Chris Caplice
Executive Director, Center for Transportation and Logistics
Thesis Supervisor

Accepted by ....
Heidi M. Nepf
Chair, Departmental Committee for Graduate Students

Piecewise Truckload Network Procurement

by

Jefferson Huang

Submitted to the Department of Civil and Environmental Engineering
on August 5, 2011, in partial fulfillment of the
requirements for the degree of
Master of Science in Transportation

Abstract

Faced with procuring transportation over its freight network, a shipper can either bid out all of its lanes at once, or somehow divide up the network and bid it out in pieces. For large shippers, practical concerns such as attendant manpower requirements and exposure to financial/operational risks can make the former undesirable or even infeasible. Such a shipper therefore needs to determine how to best allocate the lanes in its freight network to different bids to be run at different times. This thesis addresses this allocation problem.

Two related approaches are presented. The first focuses on explicitly preserving the synergies that arise in truckload network operations while attempting to balance the sizes of each bid, and is framed as a graph partitioning problem. The second treats lanes as independent entities and frames network allocation as a bin-packing problem, with constraints that attempt to achieve both balance and, implicitly, synergy preservation. These two approaches are illustrated and evaluated using a small subnetwork consisting of lanes from a large shipper. While the graph partitioning approach works in theory, the as yet unresolved question of what constitutes a "correct" synergy definition for network partitioning purposes, and the practical significance of the constraints considered in the bin-packing approach, make this second approach more attractive. The development of a lane allocation model that can explicitly consider inter-lane synergies as well as the kinds of constraints addressed in the second approach is left for future work.

Thesis Supervisor: Chris Caplice
Title: Executive Director, Center for Transportation and Logistics

Acknowledgments

I would of course like to thank Team Walmart; in particular, my thesis advisor Dr. Chris Caplice for his support and guidance, Dr. Francisco Jauffred for encouraging me to pursue mathematics, and former MST/MLOG student Ali Lokhandwala for his help and advice during my first year. I would also like to thank Patty Glidden, Kris Kipp, and Mary Gibson for their help in all things administrative.

In addition, I would like to thank Walmart, whose funding helped make my time here at MIT and this resultant thesis possible.

I'd also like to thank my classmates, from MST and otherwise, who helped make the last two years memorable and worthwhile, and my roommate Joseph for his great taste in movies, music, and books.

Finally, I'd like to thank my parents for their unwavering encouragement, and Aric for the hilarious work anecdotes.

THIS PAGE INTENTIONALLY LEFT BLANK

Contents

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 1

Introduction

The problem that this thesis is concerned with comes from Walmart's recent initiative to ultimately take full control of freight inbound from vendors to its distribution centers (DCs). Presently about two-thirds of the retailer's inbound loads are under vendor control [24], i.e. the vendor is responsible for the transportation of its goods to Walmart. This is typically fulfilled either with the vendor's private fleet, by directly contracting with a third-party carrier, or through a freight broker. Walmart is in turn charged for the freight either explicitly on the invoice, or implicitly through the price paid for the good. A transition from the present "prepaid" freight terms to "collect", in which the retailer takes ownership of the freight at the vendor's dock, has the potential to drive down freight costs for Walmart and hence the prices of goods in its stores. For example, the addition of new inbound lanes expands the space of possible tours, and therefore opportunities for reducing empty miles, available to the 6,650 trucks [36] currently comprising Walmart's private fleet. Improved fleet utilization could also provide increased leverage in negotiating fuel prices [4].

This large scale conversion in freight terms, however, could potentially lead to a significant increase in the size of the retailer's for-hire (contract carrier) network. This is because the private fleet is used primarily to fulfill deliveries outbound from DCs to stores, which are typically characterized by both high volume and temporal regularity and hence are desirable from a planning perspective. These lanes also tend to have stricter delivery time and arrival window requirements [37] that can be more easily managed through the increased control over dispatch time and routing associated with private fleet usage. The fact that private fleet drivers are domiciled at certain DCs, and must be routed back to their home domicile at regular intervals, also contributes to the bias towards the private fleet in outbound lane fulfillment [31]. This leaves the typically lower volume and less consistent inbound lanes to be serviced primarily by for-hire carriers. While private fleet tours will certainly account for some of the added inbound loads, the imbalance in inbound and outbound volume (after conversion, about 4 to 4.5 million inbound loads per year, versus 2 to 2.5 million outbound [24]) present in Walmart's network and the bias towards fulfilling outbound loads via the private fleet mean a large increase in the number of loads Walmart must bid out regularly is inevitable. For example, in a given month [24] Walmart receives on the order of 315,000 inbound loads, with 115,000 loads under the retailer's control and 200,000 under vendor control, at its DCs, and delivers 120,000 loads to its stores. Hence converting the freight terms of all vendor controlled inbound freight could potentially increase the number of for-hire loads bid out by a factor of two or more.

1.1 To Partition or Not to Partition

Such an increase in the size of the for-hire network begs the question of how transportation over this network should be procured. In particular, is it to Walmart's, or in general a large shipper's, advantage to bid out the entire network at once, or to somehow divide it up and bid out each piece separately? Three important factors relevant to this determination are TL economics, fixed vs. variable bid costs, and risk exposure.

TL Economics The economics of truckload (TL) transportation, which are reviewed in Chapter 2, favor bidding out all lanes at once. In short, certain groups of lanes can be served at lower cost by a single carrier than with multiple carriers, i.e. the cost of serving a lane is often conditional on what other lanes the carrier is serving. This is where the recent interest in the application of combinatorial auctions, in which bidders are allowed to submit bids for items conditional on winning certain other items, to TL procurement comes from. The concern with partitioning the for-hire network, then, is that it will prevent certain conditional bids from being formed that could have resulted in a lower cost allocation of lanes to carriers for the shipper. In general, including more lanes in a TL network translates into more opportunities for bidding carriers to find groups of lanes that complement each other and/or their own network, which can potentially translate into lower bids due to the predominantly cost-plus nature of the TL market. This is the motivation behind collaborative logistics [11].

Fixed vs. Variable Bid Costs The prominence of fixed versus variable costs involved in the transportation procurement process, from both the shipper and carrier perspective, also affects the desirability of bidding the entire network out at once. This process generally consists of three steps [8]:

(1) **Pre-Auction** The shipper prepares the information that carriers will use to bid on the network. This includes defining lanes, forecasting lane demand over the contract period, deciding which carriers to invite, and specifying the form of the bids. The form of the bids includes the desired rate definition (e.g. rate per move, rate per mile), service-related details (e.g. equipment type, capacity availability), and allowable bid types (e.g. simple, conditional).

(2) **Auction** The shipper's network and related data is sent to the participating carriers for analysis. The carriers then determine the rates with which they will bid, and submit their bids to the shipper. If the auction consists of multiple rounds, the carriers then receive feedback from the shipper and can update their bids.

(3) **Post-Auction** The shipper determines the winning set of bids for each lane by solving the Winner Determination Problem (WDP). Besides striving for a low cost set of bids for its lanes, the shipper will often have other side constraints, such as guaranteeing a certain amount of business to certain carriers or restricting

the number of different winning carriers. The results of the WDP are then sent to the shipper's planning, execution, auditing, and payment systems.

The fixed costs are defined here as those that do not depend on the number of lanes being bid out. The costs to the shipper associated with defining lanes, inviting carriers, specifying the bid format, and communicating the network to carriers can all be considered fixed costs. The costs of procuring a bidding tool and changing contracts are also fixed. Variable costs, on the other hand, are those whose magnitude does depend on the number of lanes bid out. From the shipper's perspective, these include the costs of forecasting lane demand, solving the WDP, and uploading final rates, while for carriers these include the costs associated with determining what rates to submit (e.g. analyzing lanes, forming conditional bids).

If the shipper's fixed costs are more important than its variable costs, it would be advantageous to bid out the entire network at once. This situation indicates the presence of potential economies of scale for the shipper in carrier procurement, as the shipper can reduce its procurement expenditure per lane by increasing the number of lanes bid out at once. On the other hand, if variable costs are more important, one needs to examine the behavior of the bid cost as a function of the number of lanes contained in the bid. In general, there are four possibilities. The bid cost could, with an increasing number of lanes:

(1) **Increase linearly** There is no cost advantage to not partitioning vs. partitioning.

(2) **Accelerate** (superadditive costs) The total cost of running several smaller bids will be less than running one large bid; partitioning is desirable.

(3) **Decelerate** (subadditive costs) The cost of running one large bid will be less than the total cost of running several smaller bids; partitioning is undesirable.

(4) **Behave in some other nonlinear fashion** The desirability of partitioning depends on the total number of lanes to be bid out and how many lanes are in each partition.

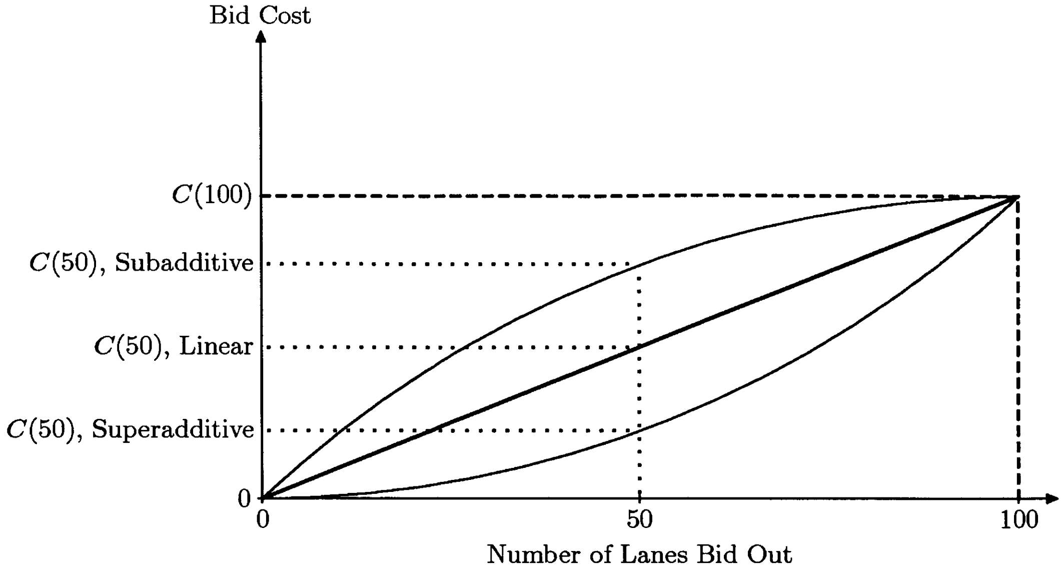


Figure 1-1: Comparison of Variable Cost Behavior

Figure 1-1 illustrates the first three cases. In particular, comparing the cost of bidding out all 100 lanes at once with running two bids with 50 lanes each, in the linear case $2 \cdot C(50) = 100$; in the superadditive case, $2 \cdot C(50) < C(100)$, and in the subadditive case $2 \cdot C(50) > C(100)$.

Risk Exposure Increasing the fraction of a shipper's network up for bid at once can create financial risks for both the shipper and carriers. The risk here is largely related to "putting all of your eggs in one basket"; the shipper risks both picking the wrong market environment and having to transition to a new set of carriers on a large scale, while carriers risk either losing a significant portion of their business with the shipper if they are incumbent, gaining morė business than they can handle within a short period, or missing out on opportunities to haul the shipper's freight.

Over time, the TL market can "tighten" or "loosen". The former means that there is more freight to be hauled than carriers to haul it, implying that carriers have more leverage in this case and that rates will tend to be higher. The latter means that there is plenty of capacity; here shippers have more leverage in negotiating rates since they have plenty of options. Hence in this case rates will tend to be lower. As the market fluctuates over time between these two states, the rates that carriers will bid on the shipper's lanes will fluctuate as well, assuming that carriers respond to bids based on contemporaneous market conditions. In bidding out its entire network at once, the shipper is hence exposed to the risk of poor market timing in that the market may loosen after the bid, leaving the shipper locked into higher rates. Here bidding out the network in pieces at regular intervals can mitigate the risk of poor market timing when the market fluctuates. The reasoning is that, over time, the well-timed bids (i.e. bids that happen before the market tightens) will tend to cancel out the poorly timed ones.

In addition to the risk of poor market timing, letting the entire network out for bid at once means that, absent constraints limiting the number of new carriers allowed to win, the shipper can potentially move to a new set of carriers over its network. This implies that there will be a period over which many of the shipper's lanes will be served by carriers not necessarily familiar with the shipper's operations, e.g. routes and transactional systems, and that many lanes will run an increased risk of service failure.

From the carrier's perspective, having to bid on the shipper's full network can present risks for both incumbent and new carriers. For incumbents, especially those that are heavily invested in a shipper's network and for which the shipper's business represents a significant portion of their revenue, having the shipper's entire network up for bid at once means that such carriers can potentially lose a large fraction of their business within a short period of time. At worst, this could mean bankruptcy for these carriers. For non-incumbents, there are risks for both carriers having the ability to bid on lanes across the shipper's entire network and those that are limited by available resources in bid response. In the former case, the carrier may end up winning a large number of its bids because the costs of hauling the shipper's freight were underestimated (i.e. the "winner's curse" [5]), with the number of lanes that can potentially be won in this way increasing as the number of lanes bid out at once increases. Such a carrier would then risk incurring losses on a large scale in the short term if it must default on certain lanes, as well as in the long term if the impact to service level causes the carrier to not be invited to future bids. On the other hand, some carriers may lose the opportunity to serve certain lanes in the shipper's network if they are unable to bid on those lanes due to resource limitations.

1.2 The Allocation Problem

If the shipper ultimately decides that bidding out all lanes at once is to its advantage, outside of preliminaries such as determining contract duration it is essentially ready to begin the procurement process. However, if bidding out the network in pieces emerges as a more desirable course of action, the shipper is faced with both determining how many bids to run and how to allocate its lanes to each bid. This thesis addresses the latter question. In particular, we consider two approaches to the allocation problem – the first attempts to account for inter-lane synergies explicitly, while the second considers business constraints that may be relevant to a large shipper.

1.2.1 Inter-Lane Synergies

The allocation problem can be approached from the perspective of maximizing the preservation of network synergies that arise from the economics of truckload transportation.

We do not know *a priori* what a bidding carrier's network will look like during an arbitrary procurement event, and hence how well a given lane in the shipper's network complements it. One can however identify groups of the shipper's lanes that, based on the economics of TL operations, appear attractive and hence have the potential to constitute a conditional bid. Of course, different groupings can have different degrees of attractiveness. For example, while all round trips are desirable, one consisting of 25 loads per week both ways is more attractive than one with 49 one way and 1 in the other direction, since the former can better preserve equipment balance. This degree of attractiveness is defined here as the synergy existing between the lanes in question. Since assigning any of the lanes in a grouping to different bids precludes bidding carriers from forming a conditional bid with those lanes, the synergy associated with a grouping can be interpreted as the cost of bidding out one or more of the constituent lanes separately.

An additional objective in allocating the shipper's lanes to different bids is that the bids should ultimately be about the same "size". Some measures of bid size include the number of lanes, number of loads, and dollar value of the lanes being bid out. This objective is useful in avoiding ending up with bids containing a single or very few lanes. Also, we do not assume any reason to make certain bids larger than others.

Hence the present partitioning problem involves finding an allocation of lanes to a given number of bids such that the cost of the allocation, defined as the total amount of synergy forfeited, is as small as possible while keeping the distribution of lanes between bids as even as possible. The synergies between lanes or, as will be pursued in this thesis, groups of lanes associated with a given DC can be represented by a synergy network (see Chapter 3.1). If only pairwise synergies are considered, this is a simple graph in which vertices represent the element (lane, DC) being assigned, edges the existence of synergy between two elements, and edge weights the magnitude of synergy. In general, if groupings containing more than two elements are considered, the synergy network is a hypergraph. Viewed in this manner, the partitioning problem becomes a graph/hypergraph partitioning problem.

Here we will only be concerned with preserving opportunities for follow-on loads, round trips, and origin/destination packages. The latter involve bundling multiple outbound/inbound lanes at a given location, and can arise when a carrier has inbound/outbound volume at that location in its own network that it needs to balance [34]. While tours involving more than two lanes can certainly be beneficial to carriers, the added value of identifying and attempting to preserve such groupings from among the shipper's lanes is unclear. In practice, constraints on manpower, time, and bid support tools often mean that when conditional bids are actually submitted they remain relatively simple (e.g. out and backs) . Even when the carrier has access to the resources necessary to create larger and more complex conditional bids, shipper constraints can prevent many of these larger bids from being awarded . For example if the shipper has a preferred carrier for a lane, any bids from other carriers containing that lane will be unusable. Clearly, as the number of lanes in the conditional bid increases the more likely that such a constraint will apply. In addition, even if such a package is actually awarded, demand and timing variability between the constituent lanes often mean that the tour rarely ends up being executed at the routing guide level . Again, as the number of lanes in the proposed tour increases the likelihood of execution in practice decreases.

1.2.2 Business Constraints

Alternatively, the allocation problem can be formulated as essentially a bin-packing problem, i.e. we're given a set of bids (bins) to which we want to allocate lanes in a manner consistent with certain shipper-defined constraints. The constraints that will be dealt with in this thesis are:

(1) **Bid Value Balance** The total value of the lanes in each bid should be roughly the same.

(2) **Location Balance** Depending on the location type, either

   (a) evenly distribute the location's volume between bids (applies to fleet domiciles, the inbound side of all distribution centers, and some ZIP clusters), or

   (b) keep all volume inbound to/outbound from the location in the same bid, and evenly allocate locations of that type between bids (applies to center points, import facilities, and some ZIP clusters).

(3) **Lane Quality Balance** Each bid should contain roughly the same number of desirable and undesirable lanes[1].

(4) **Region-to-Region Balance** Each pair of regions should have roughly the same volume, for both directions, assigned to each bid.

[1] Note that the concept of "desirability" depends on the particular shipper in question, and is highly subjective. This will be considered in more detail in Chapter 4.

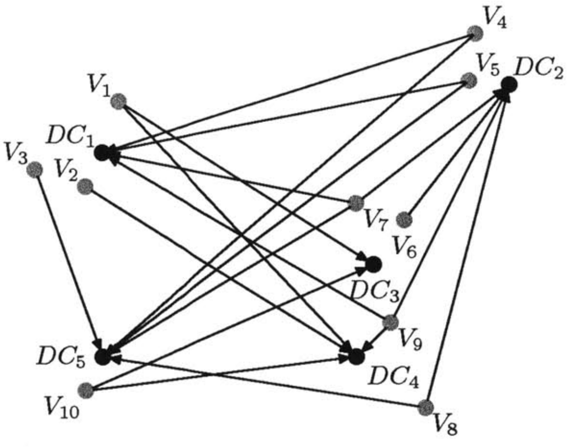


Figure 1-2: The Toy Subnetwork

While this approach does not consider synergies explicitly, the Location Balance (in particular (b)) and Region-to-Region Balance constraints have the effect of helping to preserve certain synergies inherent in truckload operations – in particular follow-on opportunities at a given location in the former, and out-and-backs in the latter.

1.3 The Toy Subnetwork

The two lane allocation approaches described in general in Section 1.2, and in more detail in Chapters 3 and 4, were evaluated using a subset of lanes taken from a large shipper's inbound freight network. This subnetwork, shown in Figure 1-2 above, consists of 19 lanes inbound to 5 distribution centers (each indicated by DC_i, $i = 1, \ldots, 5$), producing a total average flow of 1,623 loads per week. The distances depicted between the 10 vendor clusters (each indicated by V_j in the figure, $j = 1, \ldots, 10$) and DCs in Figure 1-2 indicate how far their associated actual vendor clusters are from the associated actual DCs. This depiction will be useful in constructing the toy subnetwork's associated neighborhood network in Chapter 3.

1.4 Thesis Summary

The remainder of this thesis is organized as follows. Chapter 2 provides some background on the truckload (TL) transportation industry and operations, and explains how inter-lane synergies arise in this setting. Chapter 3 presents the graph partitioning approach to network allocation. In particular, several measures of inter-lane synergy are proposed, and allocations of the toy subnetwork lanes are obtained by applying different graph partitioning heuristics to the graph representation of the lanes and these synergies. Chapter 4 in turn presents the bin-packing approach. A representation of the allocation problem as an integer program is proposed, with constraints corresponding to those listed in Section 1.2.2. This chapter also explores some potential methods to automate the ranking of lanes according to desirability, and likewise presents sample allocations of the toy subnetwork lanes. Chapter 5 then compares the approaches presented in Chapters 3 and 4. Finally, Chapter 6 summarizes the thesis, provides a recommendation for large shippers faced with the network allocation problem, and suggests directions for further research.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 2

TL Transportation Background

This chapter begins by providing a brief overview of the truckload (TL) transportation industry. Next, Section 2.2 reviews practical considerations associated with this mode in terms of both operations and lane pricing. Finally, in Sections 2.3 and 2.4 we identify, based on these concerns, the primary sources of cost efficiencies for carriers operating in this mode.

2.1 The TL Trucking Industry

According to Standard & Poor's [25], the US commercial freight transportation market's aggregate revenue, which includes the trucking, rail, air, water, and pipeline modes, was approximately $665 billion in 2009, or about 4.7 percent of the US gross domestic product. Trucking was the dominant segment, accounting for approximately $545 billion (82 percent) of the market, and is itself comprised of private and for-hire carriers. The private carriage industry is estimated to be valued at around $260 billion, or about 47.7 percent of the trucking market, with the for-hire industry accounting for the remaining $285 billion.

We are concerned here with TL for-hire carriage, which accounted for about $246 billion (86%) of the for-hire market, with the remainder attributed to less-than-truckload (LTL) and ground package delivery companies. The TL market is highly competitive; barriers to entry and exit are low, operators (with the exception of specialty equipment carriers) compete primarily on cost, and switching costs for shippers are usually low [8]. This market is also highly fragmented; about 30,000 of an estimated 45,000 TL companies had annual revenues of less than $1 million [25].

2.2 TL Operations & Lane Pricing

TL, which is used for shipments in excess of 10,000 pounds, is a direct mode. This means goods are shipped from origin to destination with no intermediate pickup/drop off stops. By contrast, a package shipped via LTL is combined with others with different origins and/or destinations and passes through one or more consolidation terminals en route to its destination. Hence TL trucking is usually likened to taxi operations, while LTL is analogous to buses or airlines [8].

Operationally, TL appears at first to be fairly straightforward. The process generally begins with a request from the shipper to pick up a certain load. If the carrier has the capacity and is willing, it dispatches a truck and a driver to the load's origin. Once the load is picked up, the driver drives directly to its destination, where the trailer is either unloaded or dropped off. The driver then either holds at a local terminal for a follow-on load from the same area, travels empty (deadheads) to a region where loads will potentially become available, or is sent directly to pick up another load.

Deciding which driver to assign to which load, however, is often not trivial. As the carrier dispatches trucks across its network, its primary concerns are to (1) minimize idling time , (2) minimize the number of empty miles driven, and (3) route drivers back to their home locations and trucks to maintenance facilities at regular intervals. Idling, or dwell time, is incurred when a driver is waiting to be dispatched to a load's pickup point, or while waiting for a shipment to be loaded or unloaded. Loading and unloading times can be reduced or eliminated through the use of trailer pools, where the shipper pre-loads a tendered shipment prior to the dispatched driver's arrival, and in general through improved shipper/carrier coordination. The second and third concerns, however, are more difficult to address due to the high degree of uncertainty in load locations over time. Shippers typically do not notify carriers of future loads before they need to be hauled; hence carriers may not be certain a load will occur until as little as 24 hours in advance [22]. Even when a load is known, there still remains the possibility that a more desirable load may materialize in the near future. There is therefore considerable uncertainty in whether, for a given empty truck and driver at a given location, it is better to have the driver wait for a load at its current location, move empty to pick up a known load, or move empty to another area. In addition, since the available dispatch options depend on the current state of the network, the carrier must balance the immediate costs/benefits of a given dispatch possibility with longer term costs/benefits of changing the state of the network with that dispatch. For example, always taking the next available or most valuable load may lead to a situation where there are too many trucks in areas with not enough loads, or where some driver requires excessive empty miles to return home on time.

In pricing a given lane in the face of often considerable uncertainty, the carrier can protect itself by adjusting the quoted price according to the amount of perceived uncertainty (i.e. "hedge" against unfavorable outcomes [9]). Here the value associated with having a truck available at a certain location or region can serve as a guide for lane pricing. Such "regional potentials" reflect the expected future availability of freight originating from a given region and hence the possibility of having to deadhead out of that region for a follow-on load. One measure of a region's potential is the total value of all outbound loads from that region [8]. The potential P_i of region i is found by summing, over all shipments (indexed by m) and all destinations (indexed by j), the difference between the revenue R_{ij}^m from shipment m destined to region j from i and the direct cost[1] C_{ij} of hauling from location i to j.

$$P_i = \sum_m \sum_j (R_{ij}^m - C_{ij})$$

The minimum rate r_{ij}^* for a lane from region i to j (not including profit markups) can then be taken to be the rate at which the carrier, based on the regional potentials for

[1] This includes fuel, driver wages, tire wear, etc.

i and j defined above, sees no net benefit/disbenefit:

$$r_{ij}^{*} = C_{ij} - P_j + P_i.$$

One must keep in mind, however, that this rate does not take the variability of actual shipment occurrences into account, and hence is not known with certainty.

2.3 Economies of Scope

Ultimately, the cost of serving a lane is highly dependent on the likelihood that a follow-on load will be available at the lane's destination for arriving trucks, as this determines the likelihood that empty miles will have to be incurred in finding another load and/or routing the driver back to his home location. This in turn means that the cost of serving a lane depends on what other lanes the carrier is serving. For example, suppose a carrier is serving a single lane originating at location A and destined to location B, and drivers are domiciled at A. Let c_{AB} be the cost to travel from A to B, r_{AB} the revenue generated from the A to B delivery, and c_{BA} the cost of moving empty from B back to A. The carrier's total operating cost is then $C(AB) = c_{AB} + c_{BA} - r_{AB}$. But if the carrier also serves the reverse direction, its total cost is $C(AB, BA) = c_{AB} + c_{BA} - r_{AB} - r_{BA} < C(AB)$. Here the added revenue generated from the B to A movement, which the carrier must execute in both cases, effectively reduces the cost of making the A to B delivery. Also, $C(AB, BA) < c_{AB} + c_{BA} - r_{BA} = C(BA)$, i.e. serving both lanes mitigates the cost of moving empty in order to either get the driver home or to pick up a load (see Figure 2-1).

One can view the "output", or product, of a carrier as the set of lanes it is serving, where each lane is defined as the movement of a commodity between a certain origin - destination pair during a certain time period [21]. The carrier's output can then be expressed as an n-vector $\mathbf{y}$, where n is the number of possible lanes that the carrier can serve and each entry denotes the volume served on the corresponding lane. Under

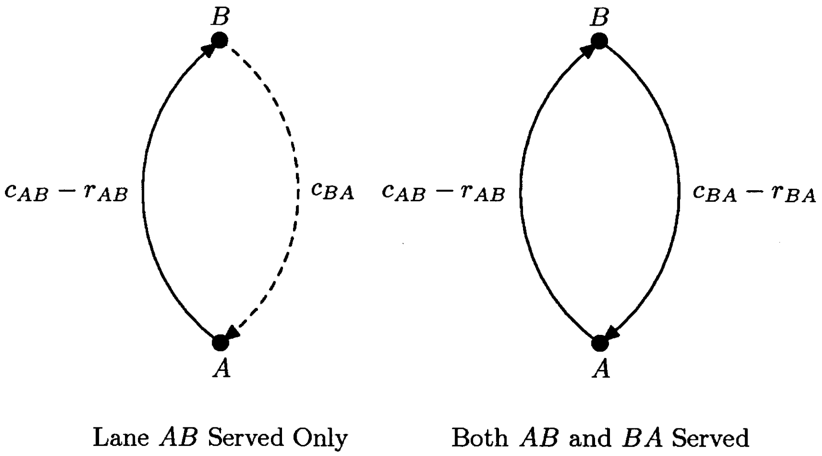


Figure 2-1: Economies of Scope

this definition the carrier's cost function $C(\mathbf{y})$ satisfies the definition of subadditivity and, in particular, of economies of scope. A cost function is subadditive with respect to $\mathbf{y}$ if, for any set of output vectors $\mathbf{y}_1, \ldots, \mathbf{y}_p$,

$$\sum_{i=1}^{p} C(\mathbf{y}_i) > C(\mathbf{y})$$

where

$$\sum_{i=1}^{p} \mathbf{y}_i = \mathbf{y}, \ ||\mathbf{y}|| > ||\mathbf{y}_i|| > 0.$$

Here $||\mathbf{u}|| = \sqrt{\mathbf{u} \cdot \mathbf{u}}$. Economies of scope exist if the vectors $\mathbf{y}_1, \ldots, \mathbf{y}_p$ are pairwise orthogonal, meaning for any $\mathbf{y}_i$ and $\mathbf{y}_j$, $1 \leq i < j \leq p$, $\mathbf{y}_i \cdot \mathbf{y}_j = 0$. For the present TL freight example this corresponds to the constraint that each lane be served by exactly one carrier, i.e. the demand on a lane cannot be satisfied by more than one carrier. This implies that $\mathbf{y}_1, \ldots, \mathbf{y}_p$ are pairwise orthogonal because, for each entry position from 1 to n, only one of the p vectors can have a nonzero entry. For example, $C([4,0]) + C([0,4]) > C([4,4])$ implies economies of scope, while $C([1,2]) + C([3,2]) > C([4,4])$ only implies subadditivity. For the two - location example above, denoting the lane from A to B as AB and the lane from B to A as BA, and letting V_i be the demanded volume on lane i,

$$\sum_{i=1}^{p} C(y_i) = C([V_{AB}, 0]) + C([0, V_{BA}]) > C(y) = C([V_{AB}, V_{BA}]),$$

where

$$[V_{AB}, 0] + [0, V_{BA}] = [V_{AB}, V_{BA}]$$

and

$$||[V_{AB}, V_{BA}]|| > ||[V_{AB}, 0]|| = ||[0, V_{BA}]|| > 0.$$

An industry that has a subadditive cost function at the industry's output level is one which would be more efficiently served cost-wise by a single firm than by two or more [3]. The existence of economies of scope for a set of products means in particular that it would be more cost-effective for a single firm to produce all of these products than if more than one firm each specialized in only certain products. Here one can interpret the scope economies present for the carrier's costs as indicating that, given two carriers whose service network consists of locations A and B and a shipper bidding out lanes AB and BA, it is more efficient cost-wise to have a single carrier serve both lanes than to have each carrier serve only one of them.

In general, the driving force behind the economies of scope that carriers experience for certain groups of lanes is the reduction of empty miles. This is an important cost element for truckload carriers and, as mentioned above, can be incurred either in repositioning a truck to a new load origin or in routing drivers back to their home domiciles. Indeed, the reduction of such non-revenue-generating miles, along with customer service, is a primary performance metric by which truckload dispatchers are evaluated [35]. While bundles of the shipper's lanes such as headhaul/backhaul pairs, lanes inbound and outbound from the same facility, and tours can all potentially help in reducing empty miles, it is important to keep in mind that the carrier's valuation of a given group of the shipper's lanes depends both on what lanes the carrier wins and on the carrier's existing network [8]. Hence it is difficult to tell in advance how valuable a given bundle of lanes will be; for instance, a headhaul/backhaul pair may actually create or exacerbate equipment imbalances, while a set of disjoint lanes may perfectly complement a carrier's existing flows.

2.4 Economies of Scale

The above implies that simply offering more freight for a TL carrier on a given lane may not result in cost efficiencies. For example, increasing volume on a lane whose destination does not typically see much outbound freight will only increase the likelihood that the carrier's trucks will have to deadhead out of that destination. Hence the importance of spatial (as well as temporal - an outbound lane from a facility is useless as a follow-on load if the inbound truck arrives too early or too late) relationships between lanes, and the absence of significant fixed costs [9], mean that economies of scale, i.e. cost efficiencies from increasing volume, are largely absent from TL operations.

Indeed, recent research has indicated that scope analysis, rather than scale analysis, is more appropriate in the context of transportation industries. One limitation of scale analysis is that it is based on looking at the behavior of a producer's cost function as outputs are increased proportionally. In the case of a single scalar output, the producer exhibits economies of scale if a proportional increase in inputs (e.g. resources needed for production) can result in a greater than proportional increase in output. In terms of costs, assuming that the total cost of production is linear in the inputs (e.g. total cost is simply the sum over all inputs of the input level times the unit cost of that input), economies of scale exist if a proportional increase in costs can result in a greater than proportional increase in output.

The generalization of scale analysis from a single scalar product to multiple products is carried out by looking at "bundles" of outputs, where the proportions of individual outputs within each bundle are fixed [21]. Aggregating the firm's outputs in this way allows essentially the same analysis as the single product case to be applied, since output can now be viewed in terms of the scalar amount of a given unit bundle produced. The applicability of such an analysis to transportation, however, is dubious. For instance, the two indices used to evaluate scale economies in transport industries have been returns to density (RTD) and network returns to scale (RTS), which are defined as follows:

$$\text{RTD} = \frac{1}{\sum_{i=1}^{K} \eta_i}$$

$$\text{RTS} = \frac{1}{\sum_{i=1}^{K} \eta_i + \eta_N}$$

Here η_i is the elasticity of the cost of producing output i with respect to the amount of output i produced, and η_N is the elasticity of the cost of production with respect to N, a measure of network size. In particular, RTD, which is equivalent to the definition of the degree of multiproduct scale economies [33], captures the effect of increasing volume proportionally on all of a carrier's existing lanes on the carrier's average cost , while RTS captures the effect on average cost of expanding both the volume on all lanes and the network size proportionately [32].

Intuitively, based on the need to balance equipment over its network, a scenario in which increasing all existing volumes proportionally does not benefit a TL carrier, since existing imbalances are exacerbated, but where adding certain lanes can improve balance and hence can reduce costs, is plausible. This applies, for instance, to the two-location example network in the previous section. Such a situation would imply that we have constant or decreasing returns to density and increasing returns to network scale, i.e. RTD $\leq$ RTS. But since all empirical studies have indicated that $\eta_N > 0$ [1], under the above definitions RTD $>$ RTS. In addition, it has been shown that RTS is inherently ambiguous, and that scope analysis should be employed instead in analyzing the behavior of costs with changing network size [2].

Returns to density have been found to exist at the corridor level for many truckload carriers, however [7]. That is, if one aggregates all of the lanes originating in one geographic area (e.g. Chicago) that are destined to another area (e.g. Atlanta), one will often find that as this corridor-wide volume increases carriers will find it more desirable to ship along the given corridor. The increase in desirability is most pronounced for corridor volumes between one and ten loads. A possible explanation is that the increased consistency that comes with increased corridor volume allows the carrier to better manage its network. In addition, if the destination area also has significant outbound volume, increasing the inbound volume to this area would be helpful in load balancing.

2.5 Summary

This chapter reviewed truckload trucking and its driving economics. In particular, economies of scope were identified as the primary source of cost efficiencies in TL operations. Such economies originate from the fact that carriers can derive an added benefit of empty mile reduction from serving certain groups of lanes that cannot be realized if the lanes are served by different carriers. On the other hand, the need for equipment balance, i.e. to have trucks where they are needed, and the absence of significant fixed costs mean that, beyond the corridor level, economies of scale are practically nonexistent in truckload trucking [7].

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 3

Explicit Synergy (Graph Partitioning) Approach

This chapter describes how the allocation problem can be formulated as a graph partitioning problem, and defines several measures of synergy (Section 3.1.1). The graph partitioning problem and several solution heuristics are subsequently described in Section 3.2, and solutions for the toy subnetwork using these methods are presented and analyzed in Section 3.3.

3.1 Modeling the Allocation Problem

Prior to allocation, the inbound network is redefined as a "neighborhood network". The idea behind the redefined network is that trucks that have unloaded freight at a particular DC can potentially pick up loads originating from vendors close to that DC. To incorporate this, for each DC a neighborhood is created that consists of the given DC and all vendor centroids such that the given DC is its nearest DC, with all loads inbound to the DC and outbound from the associated vendor centroids now incident to a single node representing the neighborhood. Hence each neighborhood contains exactly one DC; for example, a given neighborhood A corresponds to the neighborhood containing DC A. Additionally, instead of assigning individual loads to each bid, lanes are associated with their destination DC and all lanes associated

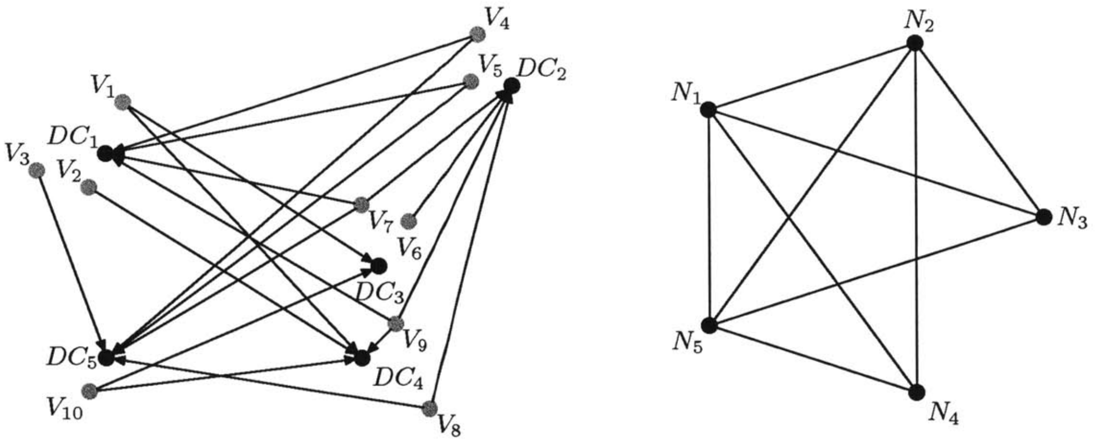


Figure 3-1: The Toy Subnetwork and Its Associated Neighborhood Network

with a DC are assigned together to a bid. Thus we are in effect using DCs as the element to be assigned to each bid. This has the benefit of preserving carriers' ability to package inbound lanes at each DC location while reducing the number of synergies to be calculated. In particular, where we previously would have had to calculate synergies between lanes, we now only need to effectively consider synergies between DCs.

Figure 3-1 above shows the toy subnetwork's corresponding neighborhood network, which is the graph that we will be partitioning. Section 3.1.1 proposes three ways in which synergies can be represented in this graph, and Section 3.1.2 indicates how a given allocation obtained via graph partitioning will be evaluated.

3.1.1 Synergy Measures

Here the type of synergy considered is that arising from the existence of follow-on opportunities. Only pairwise synergies, representing follow-ons and out-and-backs, are considered presently. Note that as the number of legs increases the continuous move or tour becomes less likely to occur in practice due to both resource limitations at many medium to smaller carriers preventing them from evaluating or even considering such packages, as well as difficulties in executing such moves once they are awarded (for example due to mismatches in actual lane volume occurrences or timing issues). It is not clear at the moment where the "cutoff" for the number of legs considered should be, i.e. the number beyond which the potential benefit of increased empty mile reduction is outweighed by low probability of success for such a package bid. Hence while hypergraph partitioning gives the option of considering more complex moves, it seems reasonable to focus on packages derived from two-legged moves, as they are both the most likely packages to materialize as bids and the most likely to actually be executed in practice.

The three synergy measures considered here are all based upon the aggregate volume flowing between neighborhoods. Let F_{ij} be the number of loads flowing from neighborhood i to neighborhood j, and Mk_{ij} be the amount of synergy between neighborhoods i and j (note that $Mk_{ij} = Mk_{ji}$) using measure k. Each attempts to capture the benefit associated with high volume between pairs, balanced volume between pairs, or both.

Measure 1: Reward High Volume

$$M1_{ij} = F_{ij} + F_{ji}$$

This measure is simply the total volume flowing between neighborhoods i and j. The idea is essentially to use outbound lane volume from a neighborhood as a proxy for the likelihood that volume will be available for trucks inbound to that neighborhood to use as a follow-on, where a higher outbound volume implies a higher likelihood that such volume will be available at that location. For example, suppose we have three neighborhoods A, B, and C, where the only flows are 5 loads per week outbound from A to C, and 50 loads per week outbound from B to C. Then if we require two subsets, we would prefer to keep B with C over A with C and A with B, since keeping B with C means the large number of opportunities for trucks inbound to B to find follow-on loads are preserved.

Recall that each flow is an aggregation of all lanes originating from vendors in the origin neighborhood to the given destination. Then assuming two bids are to be run, according to the synergy measure B and C should be assigned to one bid and A to the other. Since all lanes are associated with their destination DC, if A and C are separated then no continuous move packages can be created using lanes inbound to A and outbound to C via a vendor close to A. The same applies to B and C if B is separated from C; the idea is then to preserve the ability of carriers to create follow-on packages where it is likely that a follow-on load will be available.

Measure 2: Reward Volume Balance

$$M2_{ij} = \mathrm{MIN}(F_{ij}, F_{ji})$$

Here $\mathrm{MIN}(a, b)$ indicates the minimum of a and b. This measure also reflects the degree of balance between two neighborhoods, since the minimum of the i to j and j to i flow is the number of truckloads that are perfectly balanced between i and j. Here greater values of the measure are more desirable, since more balanced flow is better than less.

Measure 3: Reward Both High Volume & Balance

$$M3_{ij} = 2 \cdot M1_{ij} - |F_{ij} - F_{ji}|$$

This measure is essentially a modification of $M1$ that attempts to take both the volume and the degree of volume balance between neighborhoods into account. The idea is to scale up the total flow between a given pair of neighborhoods according to the degree of volume balance between those neighborhoods.

Using $M1$, to the graph partitioner the cost associated with cutting a pair of neighborhoods A and B with 2 loads per week in one direction and 0 in the other, and another pair C and D with 1 in both directions, is 2 in both cases. Therefore to make the latter more attractive we can scale up the flow between that pair in a way that reflects its degree of volume balance. One way to do this is with the formula

$$M3_{ij} = (F_{ij} + F_{ji}) \left[1 + \alpha \frac{T - \delta}{T}\right],$$

where

$T \equiv$ Max. allowable difference between F_{ij} and F_{ji},

$\delta \equiv |F_{ij} - F_{ji}|$,

$\alpha \equiv$ parameter determining how much to scale up for a certain amt. of balance.

For example, letting $\alpha = 1$ and $T = F_{ij} + F_{ji} = 200$, in the example above the flow between neighborhoods C and D is doubled,

$$M3_{CD} = (1+1)\left[1 + 1 \cdot \frac{2-0}{2}\right] = 4,$$

while the flow between neighborhoods A and B remains the same,

$$M3_{AB} = (2+0)\left[1 + 1 \cdot \frac{2-2}{2}\right] = 2.$$

Here we will take $\alpha = 1$ and $T = F_{ij} + F_{ji}$ as defaults; hence recalling that $M1_{ij} = F_{ij} + F_{ji}$ we have

$$\begin{aligned} M3_{ij} &= M1_{ij}\left[1 + \frac{M1_{ij} - |F_{ij} - F_{ji}|}{M1_{ij}}\right] \\ &= M1_{ij} + M1_{ij} - |F_{ij} - F_{ji}| \\ &= 2 \cdot M1_{ij} - |F_{ij} - F_{ji}| \end{aligned}$$

3.1.2 Partition Performance Measures

Given a synergy measure, the graph $G = (V, E)$, where each vertex corresponds to a neighborhood, vertex weights the total number of loads inbound to the neighborhood's DC, each edge the existence of synergy between two neighborhoods, and edge weights the magnitude of synergy, is partitioned using one of the methods described in Chapter 3.2.3. The two partition performance metrics considered here are the

(1) **Difference Between Subset Sizes**, corresponding to the number of loads bid out in each bid, and the

(2) **Total Weight of Crossing Edges**, corresponding to the "loss" in synergy, according to the measure used, associated with bidding out the network in those pieces.

It may not be possible to optimize both of these performance measures simultaneously. This can be seen for the graph associated with the parallel sparse matrix-vector multiplication example in Section 3.2.2 (see Figure 3-2); the smallest cut size can only be achieved by increasing the imbalance in subset sizes from the optimal size, and vice versa (see Table 3.1).

3.2 Graph Partitioning

This section provides background on graph partitioning. We begin by describing a practical problem that motivated the development of graph partitioning. Section 3.2.2 then defines the graph partitioning problem in general. Finally, Section 3.2.3 describes the three solution heuristics that were used to generate the toy subnetwork lane allocations presented in Section 3.3.

3.2.1 Motivation

For some partitioning problems, a measure of the quality of a given partition may be how "evenly" the objects are distributed. An example is the fair allocation of players to teams. If each player is assigned a number indicating his/her skill level, a fair allocation would be one where the sums of the players' skill levels for each team are equal. If the numbers used to capture the players' skill levels are nonnegative integers, this problem is known in general as the number partitioning problem. Other applications of number partitioning include evenly distributing tasks among workers or computer processors, very-large-scale integrated (VLSI) circuit design [29], and public key cryptography [28]. Number partitioning is also known to be NP-complete [15], and is often used as the basis for proving the NP-completeness of other number-based problems, such as bin packing, multiprocessor scheduling, quadratic programming, and knapsack-type problems [29].

The quality of a partition may also depend on relationships between the objects being allocated. For example, suppose we have a number of tasks for a computer with distributed memory and multiple processors, or a set of connected computers, to complete, and that the tasks have certain pairwise data dependencies. If communication between processors/computers is expensive, a good allocation of these tasks would not only give each processing unit about the same amount of work, but would also minimize the units' need to talk to one another.

Examples of parallel computation problems include the solution of partial differential equations, sparse Gaussian elimination, and sparse matrix-vector multiplication. The latter, in particular, can be formulated as follows [10]. Suppose we are given a vector $\mathbf{x}$ and a sparse[1] matrix $\mathbf{A}$, and wish to compute the vector $\mathbf{y} = \mathbf{Ax}$. One way to compute $\mathbf{y}$ in parallel is to define the calculation of the i^{th} element of $\mathbf{y}$, $y_i = \sum_j A_{ij}x_j$, as a single task, and to let the computer calculating y_i store the value of x_i and all nonzero values in the i^{th} row of $\mathbf{A}$. Under this definition, the computer assigned to calculate y_i needs to get the value of x_j, for all j corresponding to a nonzero A_{ij}, from the computer assigned to calculate y_j.

3.2.2 Problem Definition

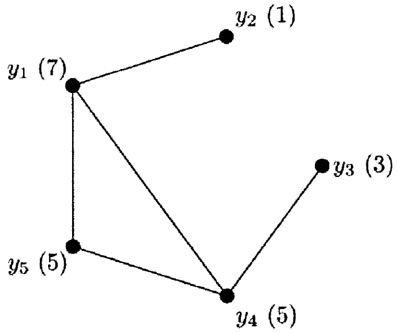
The tasks to be assigned, and the data dependencies between them, can be represented by an undirected graph $G = (V, E)$, where each vertex in V corresponds to a task and each edge in E corresponds to a data dependency between two tasks. Each vertex v and edge e can also have a weight, w_v and w_e, corresponding to task v's workload and data dependency e's inter-processor/computer communication burden, respectively. An example of a sparse matrix-vector multiplication problem, and the corresponding graph, are shown in Figure 3-2. Here vertex weights (in parentheses) are taken to be the number of addition/multiplication operations required to calculate the corresponding element of $\mathbf{y}$, and edge weights are all equal to 1.

A solution to the task assignment problem for p processors/computers, or more

[1] Mostly zeros – only non-zero entries are stored in memory

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & & A_{14} & A_{15} \\ A_{21} & & & & \\ & & A_{33} & A_{34} & \\ A_{41} & & A_{43} & & A_{45} \\ A_{51} & & & A_{54} & A_{55} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}$$

Problem



Graph Representation

Figure 3-2: Parallel Sparse Matrix-Vector Muliplication Example

generally the ***graph partitioning problem***, amounts to an allocation of vertices in G to p disjoint sets P_i such that $\bigcup_{i=1}^{p} P_i = V$. An optimal solution is one where the sizes of each set $|P_i| = \sum_{v \in P_i} w_v$, i.e. the sum of the weights of the vertices assigned to that set, are exactly or approximately equal and the sum of the weights of all edges crossing between sets, $\sum_{e \in E \cap (P_i \times P_j),\ \forall i \neq j} w_e$, is as small as possible.

For the example problem in Figure 3-2, for $p = 2$ there are $5 + \binom{5}{2} = 15$ possible solutions. This is because we can either have one vertex in one set and four in the other, or two in one set and three in the other. Given the former, we have 5 choices for the isolated vertex; given the latter, there are $\binom{5}{2}$ ways to select two vertices to isolate from the rest. The possible solutions are enumerated in Table 3.1. Since the sum of the vertex weights is odd (21), a "perfectly balanced" partition here is one where the subset sums differ by 1. There is one such partition for this problem, $\{\{y_1, y_3\}, \{y_2, y_4, y_5\}\}$. We can, however, reduce the cut size by 1 by allowing the subset sum imbalance to increase by 1 ($\{\{y_1, y_5\}, \{y_2, y_3, y_4\}\}$), or reduce the cut size by 2 by allowing the subset sum imbalance to increase by 2 ($\{\{y_1, y_2\}, \{y_3, y_4, y_5\}\}$ or $\{\{y_3, y_4\}, \{y_1, y_2, y_5\}\}$).

3.2.3 Partitioning Heuristics

A variety of heuristics have been developed to find approximate solutions to the graph partitioning problem, which is known to be NP-complete [13]. The implementations

Solution	$\|P_1\|$	$\|P_2\|$	Cut Size
$\{\{y_1\},\{y_2,y_3,y_4,y_5\}\}$	7	14	3
$\{\{y_2\},\{y_1,y_3,y_4,y_5\}\}$	1	20	1
$\{\{y_3\},\{y_1,y_2,y_4,y_5\}\}$	3	18	1
$\{\{y_4\},\{y_1,y_2,y_3,y_5\}\}$	5	16	3
$\{\{y_5\},\{y_1,y_2,y_3,y_4\}\}$	5	16	2
$\{\{y_1,y_2\},\{y_3,y_4,y_5\}\}$	8	13	2
$\{\{y_1,y_3\},\{y_2,y_4,y_5\}\}$	10	11	4
$\{\{y_1,y_4\},\{y_2,y_3,y_5\}\}$	12	9	4
$\{\{y_1,y_5\},\{y_2,y_3,y_4\}\}$	12	9	3
$\{\{y_2,y_3\},\{y_1,y_4,y_5\}\}$	4	17	2
$\{\{y_2,y_4\},\{y_1,y_3,y_5\}\}$	6	15	4
$\{\{y_2,y_5\},\{y_1,y_3,y_4\}\}$	6	15	3
$\{\{y_3,y_4\},\{y_1,y_2,y_5\}\}$	8	13	2
$\{\{y_3,y_5\},\{y_1,y_2,y_4\}\}$	8	13	3
$\{\{y_4,y_5\},\{y_1,y_2,y_3\}\}$	8	13	3

Table 3.1: Parallel Sparse Matrix-Vector Muliplication Example: Solution Space

of several of these heuristics in the partitioning software Chaco [17] – inertial, spectral, and multilevel Kernighan-Lin – were used to generate the toy subnetwork lane allocations presented in Section 3.3. These heuristics are described in this section.

Inertial

The idea behind the inertial heuristic is that cutting a graph perpendicularly to the direction in which it is elongated the most will likely give a small cut. This of course means that the vertices must be assigned fixed coordinates. Given this, a partition into two subsets can be obtained as follows. First, the vertices are interpreted as point masses, and the vertex weights as mass values. The direction of elongation then corresponds to the principal axis of the distribution of point masses that has the smallest corresponding principal moment of inertia, i.e. the principal axis about which the masses are most closely concentrated. For vertex coordinates located in

$\mathbb{R}^2$, the inertia matrix for the corresponding distribution of masses is

$$\mathbf{I} = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{yx} & I_{yy} \end{bmatrix}$$

where, denoting the x-coordinate of a vertex v by v_x and the y-coordinate by v_y,

$$I_{xx} = \sum_{v \in V} v_y^2$$
$$I_{yy} = \sum_{v \in V} v_x^2$$
$$I_{xy} = I_{yx} = -\sum_{v \in V} v_x v_y$$

The principal axes turn out to be defined by the eigenvectors[2] , and the principal moments of inertia the eigenvalues, of $\mathbf{I}$ [14]. Let the eigenvectors be denoted by $\mathbf{I}_1$ and $\mathbf{I}_2$, and the eigenvalues by α_1 and α_2, where $\alpha_1 < \alpha_2$. Having obtained the direction of elongation[3], defined by $\mathbf{I}_1$, the mass distribution is cut in two by projecting the point masses onto the axis passing through $\mathbf{I}_1$, finding the median of the projected point masses, and letting all points on one side of the median be in one subset and all points on the other side be in the other subset. This procedure can be repeated on one or both of the resulting subsets to obtain partitions of the original vertex set into any number of subsets.

The solutions generated with this heuristic tend to be "banded", which depending on the application may or may not be a good thing [17]. In addition, while the method is fast, the quality of the partitions tend to be poor in general, as it does not take the connectivity of the graph into account [17]. In particular, an example of a pathological case [14] is a graph that is "+"-shaped, where the vertices along the horizontal are widely dispersed along the horizontal axis and densely connected and those along the vertical are narrowly dispersed around the median of the vertices on the horizontal

[2] An *eigenvector* $\mathbf{u}$ of a matrix $\mathbf{T}$ is defined as a vector for which the linear transformation defined by $\mathbf{T}$ amounts to a scaling of the vector by some constant λ, i.e. $\mathbf{Tu} = \lambda\mathbf{u}$. The λ associated with an eigenvector $\mathbf{u}$ is its corresponding *eigenvalue*.

[3] Note that since we want the axis about which the point masses are most tightly clustered, this axis can also be obtained by finding the least squares linear fit for the point mass distribution.

and densely connected. Here the inertial heuristic would result in a partition with a large associated cut.

Spectral Bisection

Spectral methods use the "spectrum" of a graph to generate a partition. Here a graph's spectrum refers to the set of eigenvalues of the graph's Laplacian matrix $\mathbf{L}$, defined as the difference between the graph's degree matrix $\mathbf{D}$ and its adjacency matrix $\mathbf{A}$, i.e.

$$\mathbf{L} = \mathbf{D} - \mathbf{A}$$

where, for some numbering of the graph's vertices from 1 to $n = |V|$, the elements of $\mathbf{D}$ and $\mathbf{A}$ are defined as

$$D_{ij} = \begin{cases} d_i = \text{ degree}^4 \text{ of vertex } i & , \text{if } i = j \\ 0 & , \text{otherwise} \end{cases}$$

and

$$A_{ij} = \begin{cases} 1 & , \text{if an edge exists between vertices } i \text{ and } j \\ 0 & , \text{otherwise,} \end{cases}$$

respectively.

In Chaco the spectrum of $\mathbf{L}$ can either be used to recursively bisect a graph, or to divide it into four (quadrisection) or eight (octasection) pieces at once. Here however we will only be concerned with bisection. To demonstrate how the spectrum of $\mathbf{L}$ can be used to partition a graph, we now describe the formulation presented in [16] of spectral bisection for a connected graph where all vertex and edge weights are equal to 1. A detailed description of the formulation used in Chaco for weighted spectral partitioning can be found in [18].

Suppose the vertices of a connected graph $G = (V, E)$ are numbered from 1 to $n = |V|$. We can then assign, for each vertex $i \in V$, a variable $x_i = \pm 1$ such that $\sum_{i \in V} x_i = 0$. Such an assignment corresponds to a partition of V into two sets (all

[4] The *degree* of a vertex is the number of edges touching that vertex.

vertices assigned 1 in one set and -1 in the other) that, assuming the number of vertices is even, is perfectly balanced (for each vertex assigned 1, there is some other vertex assigned -1).

For a given assignment $\mathbf{x}$, we can then use the function $f(\mathbf{x}) = \frac{1}{4}\sum_{\{i,j\}\in E}(x_i - x_j)^2$ to count the number of edges crossing between subsets in the corresponding partition. This works because if vertices i and j, $\{i,j\} \in E$, are in the same set, they contribute $[(1)-(1)]^2 = [(-1)-(-1)]^2 = 0$ to the sum, while if they are in different sets they contribute $[(1)-(-1)]^2 = [(-1)-(1)]^2 = 4$ to the sum. Expanding $f(\mathbf{x})$, we have

$$\frac{1}{4}\sum_{\{i,j\}\in E}(x_i - x_j)^2 = \frac{1}{4}\left[\sum_{\{i,j\}\in E}(x_i^2 + x_j^2) - \sum_{\{i,j\}\in E} 2x_i x_j\right].$$

Recalling that the degree sum $\sum_{v\in V} d_i$ counts each edge exactly twice, we can rewrite the first term in brackets as

$$\sum_{\{i,j\}\in E}(x_i^2 + x_j^2) = \sum_{\{i,j\}\in E} 2 = 2|E| = \sum_{i\in V} d_i = \sum_{i\in V} x_i^2 d_i = \mathbf{x}^T\mathbf{D}\mathbf{x}.$$

The second term in brackets can be rewritten as

$$\sum_{\{i,j\}\in E} 2x_i x_j = \sum_{i\in V}\sum_{j\in V} x_i A_{ij} x_j = \sum_{i\in V} x_i \sum_{j\in V} A_{ij} x_j = \mathbf{x}^T\mathbf{A}\mathbf{x}.$$

Hence

$$\frac{1}{4}\sum_{\{i,j\}\in E}(x_i - x_j)^2 = \frac{1}{4}(\mathbf{x}^T\mathbf{D}\mathbf{x} - \mathbf{x}^T\mathbf{A}\mathbf{x}) = \frac{1}{4}\mathbf{x}^T(\mathbf{D} - \mathbf{A})\mathbf{x} = \frac{1}{4}\mathbf{x}^T\mathbf{L}\mathbf{x}.$$

Therefore the discrete optimizaton problem we wish to solve is

$$\begin{aligned} \text{Minimize} \quad & \frac{1}{4}\mathbf{x}^T\mathbf{L}\mathbf{x} \\ \text{such that} \quad & x_i = \pm 1 \qquad \forall\, i \in V, \\ & \textstyle\sum_{i\in V} x_i = 0. \end{aligned} \tag{3.1}$$

Since 3.1 is difficult to solve exactly (would require a large enumeration of feasible solutions and solving via branch and bound), and for practical purposes an approximate solution is adequate, the discreteness constraint can be relaxed, yielding the continuous approximation

$$\begin{aligned} \text{Minimize} \quad & \frac{1}{4}\mathbf{x}^T\mathbf{L}\mathbf{x} \\ \text{such that} \quad & \mathbf{x}^T\mathbf{x} = n \qquad x_i \in \mathbb{R}\ \forall\, i \in V, \\ & \textstyle\sum_{i\in V} x_i = 0. \end{aligned} \tag{3.2}$$

The solution of (3.2) relies on four properties of the Laplacian matrix **L**, which are given in Theorem 3.2.1 and proven in [18]:

Theorem 3.2.1. *Let $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n$ be the normalized* [5] *eigenvectors of $\boldsymbol{L}$, and $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ be the corresponding eigenvalues. Then $\boldsymbol{L}$ has the following properties.*

(a) $\boldsymbol{L}$ is symmetric and positive semidefinite.[6]

(b) The $\boldsymbol{u}_i$ are pairwise orthogonal. [7]

(c) $\boldsymbol{u}_1 = n^{-\frac{1}{2}}\boldsymbol{1}$, and $\lambda_1 = 0$.[8]

(d) If G is connected, then λ_1 is the only zero eigenvalue of $\boldsymbol{L}$.

Property *(b)* implies that the $\mathbf{u}_i$ span $\mathbb{R}^n$; hence we can write $\mathbf{x}$ in (3.2) as a linear combination of the $\mathbf{u}_i$'s, i.e. $\mathbf{x} = \sum_{i\in V} \alpha_i \mathbf{u}_i$, where the α_i's are real constants. The

[5] $||\mathbf{u}_i|| = \sqrt{\mathbf{u}_i^T\mathbf{u}_i} = 1 \Rightarrow \mathbf{u}_i^T\mathbf{u}_i = 1$ for all i.

[6] An $n \times n$ matrix $\mathbf{T}$ is *positive semidefinite* if, for any n-vector $\mathbf{x}$ with real-valued entries, $\mathbf{x}^T\mathbf{T}\mathbf{x} \geq 0$.

[7] $\mathbf{u}_i^T\mathbf{u}_j = 0$ for all $i \neq j$.

[8] $\mathbf{1}$ is the n-vector $(1, 1, \ldots, 1)^T$.

first constraint in (3.2) implies that, for any feasible $\mathbf{x}$, $\sum_{i\in V}\alpha_i^2 = n$:

$$\begin{aligned}
\mathbf{x}^T\mathbf{x} &= \left(\sum_{i\in V}\alpha_i\mathbf{u}_i\right)^T\left(\sum_{i\in V}\alpha_i\mathbf{u}_i\right)\\
&= (\alpha_1\mathbf{u}_1^T + \cdots + \alpha_n\mathbf{u}_n^T)(\alpha_1\mathbf{u}_1 + \cdots + \alpha_n\mathbf{u}_n)\\
&= (\alpha_1^2\mathbf{u}_1^T\mathbf{u}_1 + \cdots + \alpha_1\alpha_n\mathbf{u}_1^T\mathbf{u}_n + \cdots + \alpha_n\alpha_1\mathbf{u}_n^T\mathbf{u}_1 + \cdots + \alpha_n^2\mathbf{u}_n^T\mathbf{u}_n)\\
&= \sum_{i\in V}\alpha_i^2 = n,
\end{aligned}$$

since $\mathbf{u}_i^T\mathbf{u}_j = 1$ if $i = j$ (normalized) and 0 otherwise (pairwise orthogonal). Substituting $\mathbf{x} = \sum_{i\in V}\alpha_i\mathbf{u}_i$ into the objective function in (3.2) gives

$$\begin{aligned}
\frac{1}{4}\mathbf{x}^T\mathbf{L}\mathbf{x} &= \frac{1}{4}\left(\sum_{i\in V}\alpha_i\mathbf{u}_i\right)^T\mathbf{L}\left(\sum_{i\in V}\alpha_i\mathbf{u}_i\right)\\
&= \frac{1}{4}\left(\sum_{i\in V}\alpha_i\mathbf{u}_i\right)^T\left(\sum_{i\in V}\alpha_i\mathbf{L}\mathbf{u}_i\right)\\
&= \frac{1}{4}\left(\sum_{i\in V}\alpha_i\mathbf{u}_i\right)^T\left(\sum_{i\in V}\alpha_i\lambda_i\mathbf{u}_i\right)\\
&= \frac{1}{4}(\alpha_1\mathbf{u}_1^T + \cdots + \alpha_n\mathbf{u}_n^T)(\alpha_1\lambda_1\mathbf{u}_1 + \cdots + \alpha_n\lambda_n\mathbf{u}_n)\\
&= \frac{1}{4}(\alpha_1^2\lambda_1\mathbf{u}_1^T\mathbf{u}_1 + \cdots + \alpha_n\alpha_1\lambda_1\mathbf{u}_n^T\mathbf{u}_1 + \cdots + \alpha_1\alpha_n\lambda_n\mathbf{u}_1^T\mathbf{u}_n + \cdots + \alpha_n^2\lambda_n\mathbf{u}_n^T\mathbf{u}_n)\\
&= \frac{1}{4}(\alpha_1^2\lambda_1 + \cdots + \alpha_n^2\lambda_n)\\
&= \frac{1}{4}(\alpha_2^2\lambda_2 + \cdots + \alpha_n^2\lambda_n),
\end{aligned}$$

since, by Properties *(c)* and *(d)*, $\lambda_1 = 0$ and $\lambda_i > 0\ \forall i \geq 2$. But $\lambda_2 \leq \cdots \leq \lambda_n$ means that

$$\frac{1}{4}\mathbf{x}^T\mathbf{L}\mathbf{x} = \frac{1}{4}(\alpha_2^2\lambda_2 + \cdots + \alpha_n^2\lambda_n) \geq \frac{1}{4}(\alpha_2^2 + \cdots + \alpha_n^2)\lambda_2 = \frac{1}{4}n\lambda_2.$$

This lower bound on the objective funcion in (3.2) can be achieved by setting $\mathbf{x} =$

$\mathbf{x}^* = \sqrt{n}\mathbf{u}_2$:

$$\begin{aligned}\frac{1}{4}\mathbf{x}^{*T}\mathbf{L}\mathbf{x}^* &= \frac{1}{4}(\sqrt{n}\mathbf{u}_2)^T\mathbf{L}(\sqrt{n}\mathbf{u}_2)\\ &= \frac{1}{4}n\mathbf{u}_2^T\lambda_2\mathbf{u}_2\\ &= \frac{1}{4}n\lambda_2\mathbf{u}_2^T\mathbf{u}_2 = \frac{1}{4}n\lambda_2\end{aligned}$$

This solution is also feasible, since

$$\mathbf{x}^{*T}\mathbf{x}^* = (\sqrt{n}\mathbf{u}_2)^T(\sqrt{n}\mathbf{u}_2) = n\mathbf{u}_2^T\mathbf{u}_2 = n,$$

and

$$\sum_{i\in V} x_i^* = \mathbf{x}^{*T}\mathbf{1} = (\sqrt{n}\mathbf{u}_2)^T\mathbf{1} = \mathbf{u}_2^T\sqrt{n}\mathbf{1} = \mathbf{u}_2^T\frac{n}{\sqrt{n}}\mathbf{1} = n\mathbf{u}_2^T\mathbf{u}_1 = 0,$$

where the satisfaction of the latter constraint follows from Properties *(c)* and *(b)*. Hence $\mathbf{x}^*$ is a solution to (3.2). A bisection of the corresponding graph can be found by calculating the median of the values in $\mathbf{x}^*$ and assigning the vertices corresponding to all x_i^* greater than the median to one set, and the remaining vertices to the other set. Since multiplying all x_i^* by -1 does not change the median, $\mathbf{x} = -\sqrt{n}\mathbf{u}_2$ corresponds to the same partition. Finally, if $\lambda_2 \neq \lambda_3$, $\mathbf{x}^*$ is a unique solution.[9]

Multilevel Kernighan-Lin

A major difficulty in finding an optimal graph partition is that the number of candidate partitions grows very quickly as the number of vertices increases. For example, suppose we want to partition the graph $G = (V, E)$ into two pieces. Then the number of possible partitions, without any restrictions on the sizes of each piece, can be derived by first considering the power set $\mathcal{P}(V)$ [10] of V. Each possible subset of V corresponds to a partition of G into two pieces (for a given subset A, the correspond-

[9] Recall that we have assumed here that the graph being bisected is connected. As noted in [16], during recursive spectral bisection disconnected subgraphs do arise in practice. This is addressed in Chaco by adding a minimal number "phantom edges" to any disconnected subgraphs that arise, partitioning the resulting connected subgraph, and then removing the added edges.

[10] The *power set* $\mathcal{P}(S)$ of the set S is the set of all possible subsets of S. For example, for $S = \{a, b, c\}$, $\mathcal{P}(S) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$.

ing pieces of G are the vertices in A and those in $V \setminus A$). However, if we were to simply count the number of possible subsets of V we would be double counting, since the partitions defined by A and $V \setminus A$ and $V \setminus A$ and $V \setminus (V \setminus A) = A$ are the same. Also, we do not want to count the trivial partition (one piece is the original graph, the other has nothing), which in $\mathcal{P}(V)$ corresponds to $\emptyset$ and V. Hence the number of possible partitions of G into two pieces is[11]

$$\frac{|\mathcal{P}(V)| - 2}{2} = \frac{2^{|V|} - 2}{2} = 2^{|V|-1} - 1.$$

This of course grows very fast with $|V|$; for example, a graph with 100 vertices has $2^{99} - 1 = 6.34 \times 10^{29}$ possible partitions into two pieces.

The multilevel Kernighan-Lin heuristic [19] attempts to reduce the difficulty in finding a good graph partition by first finding a good partition of a coarse approximation of the original graph, which is often a much easier problem. The graph is then de-coarsened in stages, and refined along the way.

In particular, a coarse graph is generated by finding a maximal matching[12] in the original graph, merging each pair of matched vertices into a single vertex, and repeating on the resulting graph until it is sufficiently small. Each vertex in a coarsened version of the original graph is also assigned a weight equal to the sum of the weights of the finer vertices it contains, and the adjacency structure of the original graph is preserved by making each coarse vertex adjacent to all the neighbors of its constituent finer vertices. Where two finer vertices share a neighbor, the two edges are merged into one with a weight equal to the sum of the two finer edges' weights.

Once the original graph has been suffciently coarsened, any graph partitioning heuristic that can handle vertex and edge weights can be invoked. The implementation in Chaco uses either spectral bisection, quadrisection, and octasection described

[11] $|\mathcal{P}(S)| = 2^{|S|}$ because each element of S is either in a given subset or not; hence each subset corresponds to exactly one binary sequence of length $|S|$, where each digit corresponds to a unique element of S and 1 indicates subset membership. The number of possible subsets is therefore equal to the number of binary sequences of length $|S|$, i.e. $2^{|S|}$.

[12] A *matching* in a graph $G = (V, E)$ is a subset of E whose endpoints are all distinct. A matching is *maximal* if adding any edges would cause the set to no longer be a matching.

in the previous section [19] for this step.

The coarse, partitioned graph is then uncoarsened by proceeding along the initial coarsening process in reverse. At each step, a generalized version of the Kernighan-Lin algorithm [19], first proposed in [23] and implemented in linear time in [12] is invoked if desired to improve the current partition. The use of this algorithm is appropriate here since it is essentially a local, greedy optimization heuristic [17] whose utility depends on the quality of the initial partition it is given. Given an initial partition, it works by moving one vertex at a time between sets based on the "gain", i.e. the improvement to the partition, associated with the move. In particular, let w_{vw} be the weight of edge $(v, w) \in E$, and c_{pq} be the (symmetric) cost of having an edge cross between pieces p and q. Then the gain $g_q(v)$ associated with moving a vertex $v \in V$ currently in piece p to piece q is defined as

$$g_q(v) = \sum_{(v,w)\in E} \begin{cases} w_{vw}c_{pq} & \text{, if vertex } w \text{ is in piece } q, \\ -w_{vw}c_{pq} & \text{, if vertex } w \text{ is in piece } p, \\ w_{vw}(c_{pm} - c_{qm}) & \text{, if vertex } w \text{ is in neither piece } p \text{ nor } q. \end{cases}$$

Details on how the generalized Kernighan-Lin algorithm selects moves can be found in [19], which also contains a comparison of how the multilevel Kernighan-Lin heuristic performs compared to the inertial and spectral methods described in the previous section. While the inertial method was the fastest but produced the poorest partitions, and spectral was much slower but produced high quality partitions, multilevel Kernighan-Lin produced partitions similar in quality to the spectral method in times closer to the inertial method.

Chaco can also apply the Kernighan-Lin algorithm to any given initial partition, and in particular can use it to improve partitions generated by the inertial and spectral methods. In this thesis both the inertial and spectral methods alone, as well as coupled with (local) Kernighan-Lin, will be considered.

3.3 Toy Subnetwork Allocations

Each of the three methods described in Section 3.2.3 were used to allocate the toy subnetwork's lanes into two hypothetical bids. As noted in Section 3.1.2, the quality of a given allocation depends both on the imbalance of subset sizes (measured by the total number of loads per week bid out at once) as well as the total weight of neighborhood network edges whose endpoints are in different bids. The importance of these two performance measures can of course be weighed differently; hence depending on the decision maker, the ultimate attractiveness of a given allocation generated by the graph partitioning approach may vary. In practice, this approach could be taken as an initial step in the allocation process, with the generated allocations forming the basis for improvements based on other constraints.

3.3.1 Measure 1: Reward High Volume

The top left hand corner of Figure 3-3 shows the edge weights, corresponding to inter-neighborhood synergy values, of the toy subnetwork's neighborhood network obtained using Measure 1, which rewards high volumes. The figure also shows the allocations of the toy subnetwork's lanes found using the multilevel Kernighan-Lin, inertial, and spectral methods, with the latter two applied both with and without the local Kernighan-Lin algorithm. In particular, lanes drawn with a solid line were allocated Bid 1, and those drawn with a dashed line were allocated to Bid 2. The performance of each allocation is summarized in Table 3.2, where both the bid sizes ("B1 Size" & "B2 Size" columns) and synergy loss (S. Loss column) are in loads per week. Here the "B1 Lanes" column indicates the number of lanes assigned to Bid 1, and similarly for "B2 Lanes". The multilevel-KL and spectral (without local KL) methods produced the best allocations according to bid size difference and synergy loss. If however one is willing to allow for a larger discrepancy between the bid sizes, one can have the synergy loss reduced from 36 to 34 using the allocations produced with the inertial with KL and spectral with KL methods.

Method	B1 Lanes	B2 Lanes	B1 Size	B2 Size	Size Δ	S. Loss
Multilevel-KL	**7**	**12**	**37**	**43**	**6**	**36**
Inertial	6	13	36	44	8	46
Inertial w/ KL	11	8	50	30	20	34
Spectral	**7**	**12**	**37**	**43**	**6**	**36**
Spectral w/ KL	11	8	50	30	20	34

Table 3.2: Measure 1 (Reward High Volume): Bid Statistics

Method	B1 Lanes	B2 Lanes	B1 Size	B2 Size	Size Δ	S. Loss
Multilevel-KL	**10**	**9**	**44**	**36**	**8**	**3**
Inertial	6	13	36	44	8	5
Inertial w/ KL	**10**	**9**	**44**	**36**	**8**	**3**
Spectral	9	11	35	45	10	4
Spectral w/ KL	13	6	49	31	18	3

Table 3.3: Measure 2 (Reward Volume Balance): Bid Statistics

3.3.2 Measure 2: Reward Volume Balance

The neighborhood network edge weights and allocations using Measure 2 are shown in Figure 3-4, while the performance of each allocation is summarized in Table 3.3 above. Here the multilevel-KL and inertial with KL methods produced identical allocations which dominate the others with respect to both bid size difference and synergy loss.

3.3.3 Measure 3: Reward Both High Volume & Balance

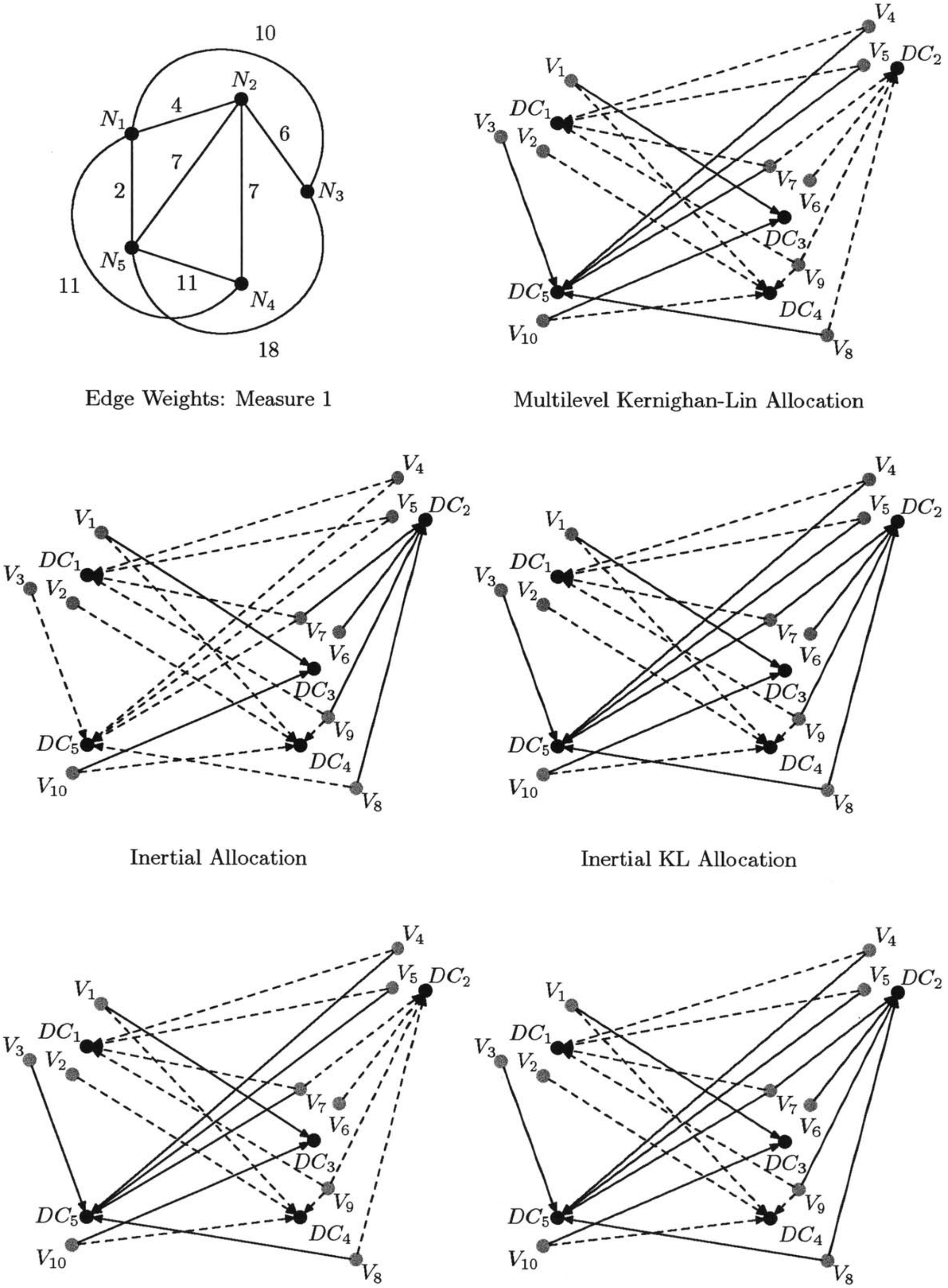
Figure 3-5 shows the edge weights of the neighborhood network using Measure 3, as well as the allocations based on this measure. Table 3.4 gives the associated allocation statistics. Here if minimizing bid size difference is paramount then the spectral method produced the best allocation, while if minimizing synergy loss is more important the allocation generated by the inertial with KL and spectral with KL methods is best. The allocation generated using the multilevel-KL method is a constitutes an intermediate solution.

Method	B1 Lanes	B2 Lanes	B1 Size	B2 Size	Size Δ	S. Loss
Multilevel-KL	**11**	**8**	**45**	**35**	**10**	**47**
Inertial	6	13	36	44	8	56
Inertial w/ KL	11	8	50	30	20	46
Spectral	**7**	**12**	**37**	**43**	**6**	**48**
Spectral w/ KL	11	8	50	30	20	46

Table 3.4: Measure 3 (Reward High Volume & Balance): Bid Statistics

3.4 Summary

This chapter presented an approach to the allocation problem that is based on minimizing the loss in inter-lane synergies resulting from allocating lanes in the network to different bids. We began by translating the inbound toy subnetwork into a neighborhood network, and defining several measures of synergy over this network (Section 3.1). This network was then allocated to two different bids using different approaches to solving the graph partitioning problem. While the best allocation was clear under synergy measure 2, for measures 1 and 3 the allocation with the smallest bid size difference did not also have the smallest synergy loss, and vice versa. Hence to be able to apply this approach in practice the relative importance of minimizing bid size difference versus minimizing synergy loss needs to be known. In Chapter 5, we will revisit the allocations presented in this chapter in comparing the graph partitioning approach to the bin-packing approach, which is described in Chapter 4.



Edge Weights: Measure 1

Multilevel Kernighan-Lin Allocation

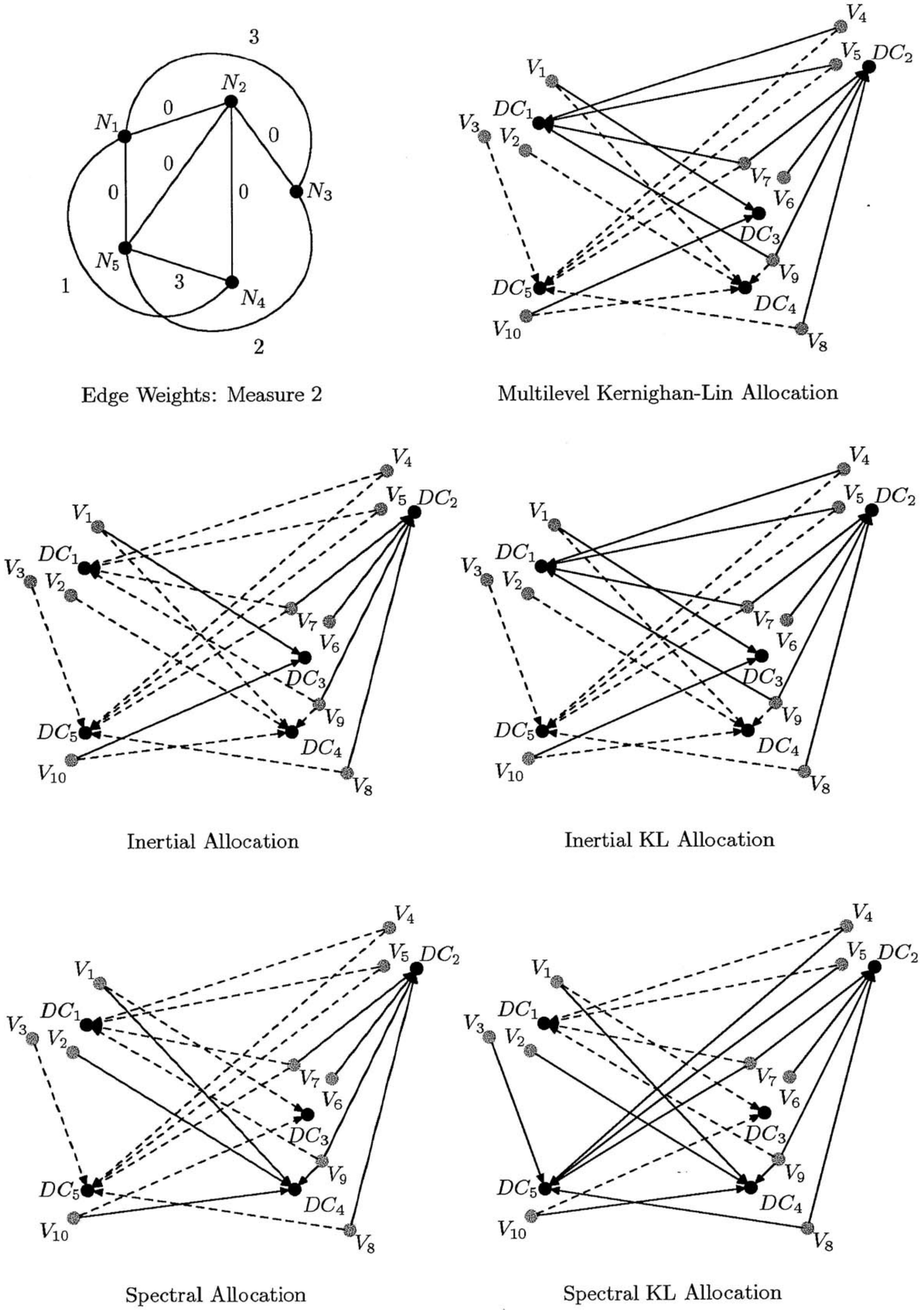
Inertial Allocation

Inertial KL Allocation

Spectral Allocation

Spectral KL Allocation

Figure 3-3: Allocations: Synergy Measure 1 (Reward High Volume)



Edge Weights: Measure 2

Multilevel Kernighan-Lin Allocation

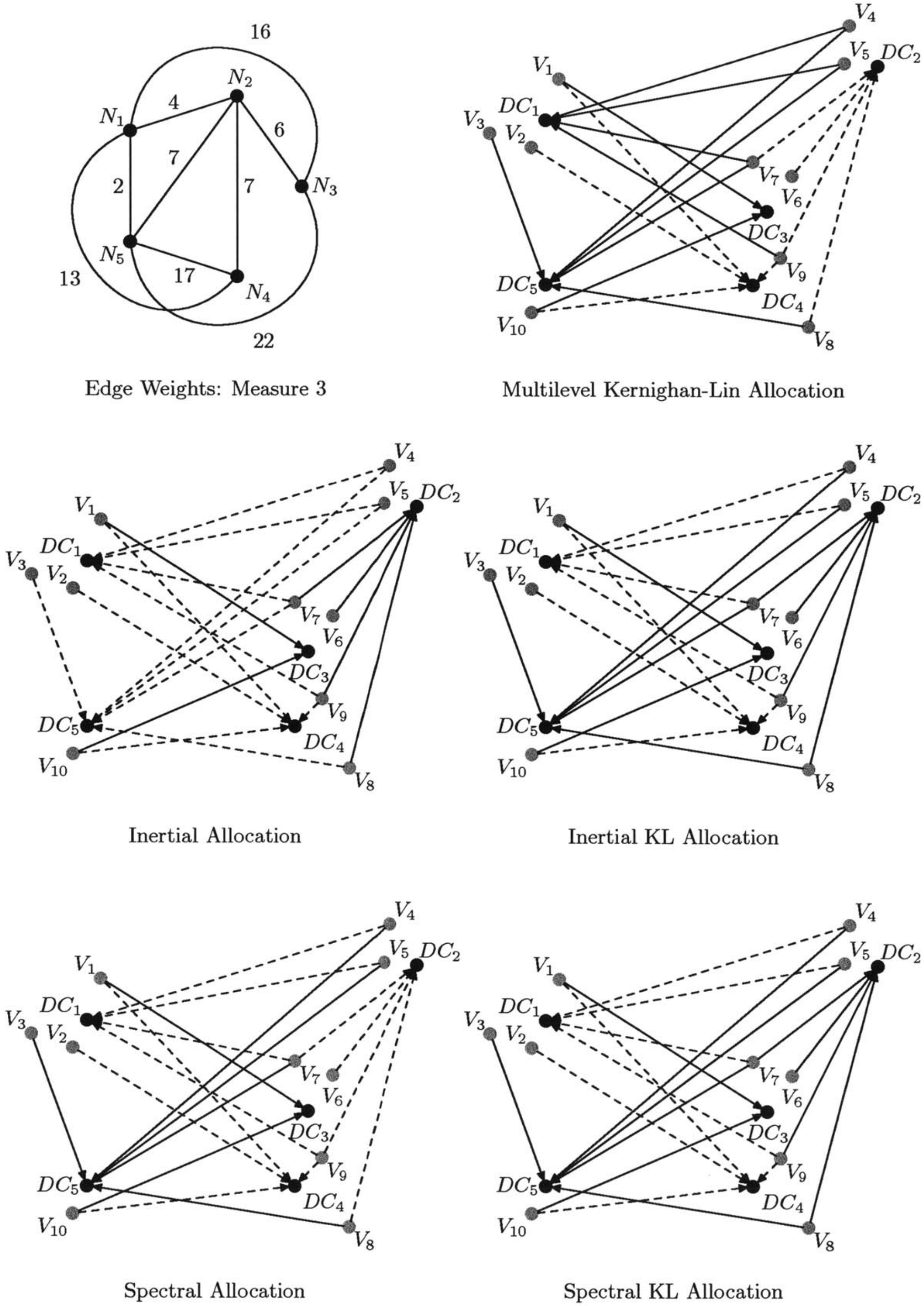
Inertial Allocation

Inertial KL Allocation

Spectral Allocation

Spectral KL Allocation

Figure 3-4: Allocations: Synergy Measure 2 (Reward Volume Balance)



Edge Weights: Measure 3

Multilevel Kernighan-Lin Allocation

Inertial Allocation

Inertial KL Allocation

Spectral Allocation

Spectral KL Allocation

Figure 3-5: Allocations: Synergy Measure 3 (Reward High Volume & Balance)

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 4

Implicit Synergy with Business Constraints (Bin-Packing) Approach

This chapter explains how the problem of finding a good allocation of lanes to a given number of bids, given a set of business constraints, can be formulated as an integer program [20]. In addition, the problem of automating the rating of lanes according to desirability is addressed in Section 4.2, and results for the toy subnetwork are presented in Section 4.3.

4.1 Modeling the Allocation Problem

Recall that the criteria given in Section 1.2.2 for a good allocation of lanes are:

(1) **Bid Value Balance** The total value of the lanes in each bid should be roughly the same.

(2) **Location Balance** Depending on the location type, either

   (a) evenly distribute the location's volume between bids (applies to fleet domiciles, the inbound side of all distribution centers, and some ZIP clusters), or

(b) keep all volume inbound to/outbound from the location in the same bid, and evenly allocate locations of that type between bids (applies to center points, import facilities, and some ZIP clusters).

(3) **Lane Quality Balance** Each bid should contain roughly the same number of desirable and undesirable lanes.

(4) **Region-to-Region Balance** Each pair of regions should have roughly the same volume, for both directions, assigned to each bid.

Section 4.1.1 explains the objective function that the model attempts to minimize, while Section 4.1.2 explains the constraints. The complete formulation is given in Appendix A, along with a tables with the definitions of the sets of things used in the model (Table A.2), the indices used (Table A.3), the decision variables (Table A.4), the data variables (Table A.5, and the objective function coefficients (Table A.6).

4.1.1 Objective Function

For each type of balance, we assign a weight indicating the desirability of achieving that type of balance. Each of these weights can be interpreted as the cost of having one unit of imbalance (e.g. a difference of $ 1 between the values of the two bids) between the bids we're allocating lanes to. The objective is then to minimize the total cost that arises from the different kinds of imbalances (bid value, location volumes, etc.). Letting b_i refer to bid i, contained in the set of all bids B among which we want to allocate the lanes in the network, and where R_m refers to region m, the weights for the different kinds of balance are

(1) $W_{(b_i,b_j)}$, for bid value balance (amount of imbalance denoted by $\sigma_{(b_i,b_j)}$),

(2) $K_{(b_i,b_j),\omega}$, for location volume balance (amount of imbalance denoted by $\nu_{(b_i,b_j),\omega}$),

(3) $Q_{(b_i,b_j)}$, for the balance of the number of locations between bids for those locations where we want to keep all inbound/outbound volumes together (amount of imbalance denoted by $\rho_{(b_i,b_j)}$),

(4) $WD_{(b_i,b_j)}$, for lane quality balance (amount of imbalance denoted by $\delta_{(b_i,b_j),d}$), and

(5) $WR_{(b_i,b_j),(R_m,R_n)}$, for region-to-region balance (amount of imbalance denoted by $\chi_{(b_i,b_j),(R_m,R_n)}$.

Besides the set of all bids B, other sets used in the model include the set of all lanes $l \in L$, the set of all locations $\omega \in \Omega$, the set of all possible lane ratings $d \in \Delta$, and the set of all regions $R_m \in R$. The model's objective function is shown in (4.1).

$$\begin{aligned}\text{Minimize} \quad & \textstyle\sum_{(b_i,b_j):i<j}(W_{(b_i,b_j)} \cdot \sigma_{(b_i,b_j)}) + \sum_{\omega\in\Omega}\sum_{(b_i,b_j):i<j}(K_{(b_i,b_j),\omega} \cdot \nu_{(b_i,b_j),\omega}) + \\ & \textstyle\sum_{(b_i,b_j):i<j}(Q_{(b_i,b_j)} \cdot \rho_{(b_i,b_j)}) + \sum_{d\in\Delta}\sum_{(b_i,b_j):i<j}(WD_{(b_i,b_j)} \cdot \delta_{(b_i,b_j),d}) + \\ & \textstyle\sum_{(R_m,R_n)\in R\times R}\sum_{(b_i,b_j):i<j}(WR_{(b_i,b_j),(R_m,R_n)} \cdot \chi_{(b_i,b_j),(R_m,R_n)})\end{aligned} \tag{4.1}$$

The decision variable for this problem is $x_{l,b}$, which is 1 if lane l in the set of all lanes L is assigned to bid b and 0 otherwise.

4.1.2 Constraints

To ensure that each lane is assigned to exactly one bid, we add the constraint

$$\sum_{b\in B} x_{l,b} = 1, \quad \forall\, l \in L. \tag{4.2}$$

Bid Value Balance

Letting p_l be the value of lane l, the total value assigned to each bid is

$$\sum_{l\in L} p_l x_{l,b} = S_b, \quad \forall\, b \in B. \tag{4.3}$$

We then define $\sigma_{(b_i,b_j)}$ as the absolute difference between the total values assigned to bids b_i and b_j, i.e.

$$|S_{b_i} - S_{b_j}| \le \sigma_{(b_i,b_j)} \quad \forall\, b_i, b_j \in B. \tag{4.4}$$

Location Balance

In this formulation, location balancing is done on the number of lanes assigned. One can balance on value instead by multiplying each $x_{l,b}$ in (4.5) by p_l and each $y_{\mu,b}$ in (4.8) by $\sum_{\ell \in L_\mu} p_\ell$.

Balance at Locations We first consider the constraints for the balanced distribution of volume at each location ω where we wish to do so. Letting L_ω be the set of lanes incident to location ω, the number of lanes assigned to each bid b at each location ω is

$$\sum_{l \in L_\omega} x_{l,b} = N_{b,\omega} \quad \forall\, b \in B,\ \omega \in \Omega. \tag{4.5}$$

We then define $\nu_{(b_i,b_j),\omega}$ as the absolute difference between the number of lanes at location ω assigned to bids b_i and b_j, i.e.

$$|N_{b_i,\omega} - N_{b_j,\omega}| \leq \nu_{(b_i,b_j),\omega} \quad \forall\, b_i, b_j \in B,\ \omega \in \Omega. \tag{4.6}$$

Balance of Locations For each location μ in the set M of locations where we want all incident lanes to be assigned to the same bid, the variable $y_{\mu,b}$ is defined to be 1 if location μ is assigned to bid b and 0 otherwise. To ensure that each such location is assigned to exactly one bid, we add the constraint

$$\sum_{\mu \in \mathrm{M}} y_{\mu,b} = 1 \quad \forall\, b \in B. \tag{4.7}$$

The total number of such locations assigned to bid b is then

$$\sum_{\mu \in \mathrm{M}} y_{\mu,b} = T_b \quad \forall\, b \in B. \tag{4.8}$$

Letting L_μ be the set of all lanes incident to location μ, to ensure that each lane incident to each such location is assigned to the same bid we add the constraint

$$x_{\ell,b} = y_{\mu,b} \quad \forall\, \ell \in L_\mu,\ b \in B,\ \mu \in \mathrm{M}. \tag{4.9}$$

The absolute difference between the number of such locations assigned to bids b_i and b_j is defined to be $\rho_{(b_i,b_j)}$, i.e.

$$|T_{b_i} - T_{b_j}| \leq \rho_{(b_i,b_j)} \quad \forall\, b_i, b_j \in B. \tag{4.10}$$

Lane Quality Balance

Here the quality of a lane is designated by -1 if the lane is undesirable, 0 if neutral, and 1 if desirable. Letting $\Delta = \{-1, 0, 1\}$, and L_d be the set of all lanes of quality d, the total number of lanes of quality d assigned to each bid b is

$$\sum_{\ell \in L_d} x_{\ell,b} = D_{d,b} \quad \forall\, d \in \Delta,\ b \in B. \tag{4.11}$$

The absolute difference between the number of lanes of quality d assigned to bids b_i and b_j is defined by $\delta_{(b_i,b_j),d}$, i.e.

$$|D_{d,b_i} - D_{d,b_j}| \leq \delta_{(b_i,b_j),d} \quad \forall\, d \in \Delta,\ b_i, b_j \in B. \tag{4.12}$$

Region-to-Region Balance

Letting v_l be lane l's volume in loads per week, and $L_{(R_m,R_n)}$ the set of lanes between regions R_m and R_n, the total volume between regions R_m and R_n assigned to bid b is

$$\sum_{\ell \in L_{(R_m,R_n)}} v_\ell x_{\ell,b} = Z_{(R_m,R_n),b} \quad \forall\, R_m, R_n \in R,\ b \in B. \tag{4.13}$$

The absolute difference between the volume between regions R_m and R_n assigned to bids b_i and b_j is defined by $\chi_{(b_i,b_j),(R_m,R_n)}$, i.e.

$$|Z_{(R_m,R_n),b_i} - Z_{(R_m,R_n),b_j}| \leq \chi_{(b_i,b_j),(R_m,R_n)} \quad \forall\, R_m, R_n \in R,\ b_i, b_j \in B. \tag{4.14}$$

4.2 Modeling Lane Desirability

This section considers how the lane desirability ratings that come into play in the Lane Quality Balance constraints can be obtained in practice. For small networks, one may have an expert review and rate the desirability of each lane by hand. However, in applications to large networks, which is the primary focus of this thesis, this is likely to be infeasible. Here we would like to automate the lane rating process by somehow translating the expert's thought processes in coming up with a lane rating into one or more classification rules that can be implemented on a computer. In general, such a classification rule would, for each lane to be rated, receive as input a certain number of the lane's characteristics captured in numeric form and output the lane's rating, which of course should agree with what the expert's rating would have been.

In general, while we expect that the expert will employ a set of internal rules for desirability that are applied to each lane, we also expect her to possess a great deal of business knowledge, such as future freight availability in certain locations, that will play into her final decision. It may be difficult to translate such knowledge, which may be very lane and/or time specific, into a set of general rules we can use to automate the rating process. In addition, different experts will likely employ slightly different rules and, even if the same rules are used, weigh the importance of certain rules differently based on past experience. Hence here the problem of how to translate expert knowledge into machine knowledge is not trivial in general.

Because of this difficulty in obtaining a literal representation of the human expert on a computer, we instead consider an indirect approach. While we expect particular experts to vary somewhat in their lane ratings, we also expect that there is some sort of underlying regularity to these ratings, and that if we choose the proper set of lane characteristics, these underlying patterns can be extracted. This pattern can then be used on its own as an acceptable approximation to an expert's opinion, or may constitute an initial step in a large-scale classification scheme in which the computer performs the initial classification and an expert subsequently identifies important lanes and adjusts the computer's ratings if necessary.

This approach of using data to learn about the underlying process that generated the data is the primary concern of the related fields of statistics and machine learning. When applied to large datasets, this approach is also aptly termed data mining. Section 4.2.1 describes the data that we attempted to "mine" for patterns that can be turned into classification rules. Section 4.2.2 describes a heuristic classification method obtained from trial and error, while Section 4.2.3 desribes three statistical/machine learning methods that can be used to find classification rules from expert-classified data. Finally, Section 4.2.4 considers the performance of these four methods compared to the human expert.

4.2.1 Data Description

The dataset from which we attempted to generate a good classification rule was obtained by first randomly selecting 267 lanes[1] from a large shipper's inbound network and giving these lanes to an expert at the shipper to rate. Each lane was ultimately assigned a rating of either undesirable, neutral, or desirable. We then considered the available statistics for each lane (e.g. distance, volume, etc.) and came up with a set of lane statistics that we felt would help indicate why a given lane was rated the way it was. This set of statistics includes

(1) **Distance** Increased distance was observed in the data to contribute to desirability in some cases.

(2) **Average Weekly Volume** Increased volume can mean increased operational predictability, and hence increased desirability.

(3) **Lane Revenue** This is the total amount charged to the carrier for a lane, i.e.

$$\text{Revenue} = \text{MIN}(\text{Raw Lane Rate} \times \text{Distance}, \text{Min. Charge}) \times \text{Avg. Weekly Vol.},$$

where, for our dataset, Raw Lane Rate is in dollars per mile per load, Distance

[1] Only lanes that are TL, dry van, and which have a raw lane rate greater than zero were considered.

is miles, Min. Charge is in dollars per load, and Avg. Weekly Vol. is in loads per week. We expect that an increase in revenue would make a lane more desirable. One of the two following variables can be used.

(4) **Driver Idle Hours** Because of limitations on driver working hours, serving a given lane entails a certain amount of idling time for the driver. We expect that lanes with less associated idle time will appear more desirable.

To estimate the total time that the driver for a given lane is idle, we first assumed that drivers drive in 11-hour shifts, and that on average a truck travels at 60 miles per hour. Given these assumptions, the amount of idling time for a given lane is simply the remainder of the total number of hours required to serve the lane divided by 11, i.e.

$$\text{Drive Idle Hours} = \text{MOD}[\text{Distance}/(60 \text{ mph}), 11 \text{ hours}],$$

where MOD(a, b), or a modulo b, indicates the remainder of a/b.

(5) **Geographic Impact Value (GIV)** As was noted in Section 2.2, the particular origin and destination of a lane plays an important role in determining the carrier's final rate for the lane. This is because to the carrier there is a certain amount of value associated with having a truck in a certain location. If a location is a consistent source of freight, the value of having a truck at that location is high; conversely, at locations where it is difficult to secure outbound freight the value to the carrier may even be negative.

The values, or "geographic impacts", perceived by a carrier for each relevant 3-digit ZIP code were obtained from [6]. These values are the origin and destination 3-digit ZIP code coefficients obtained by regressing lane cost per load on distance and origin & destination, i.e. the estimated impact on a lane's cost per load due to the lane's origin & destination alone.

All of the aforementioned variables are real-valued. Of course, any of these variables can be discretized, e.g. by defining the variable to be 0 if the associated real-

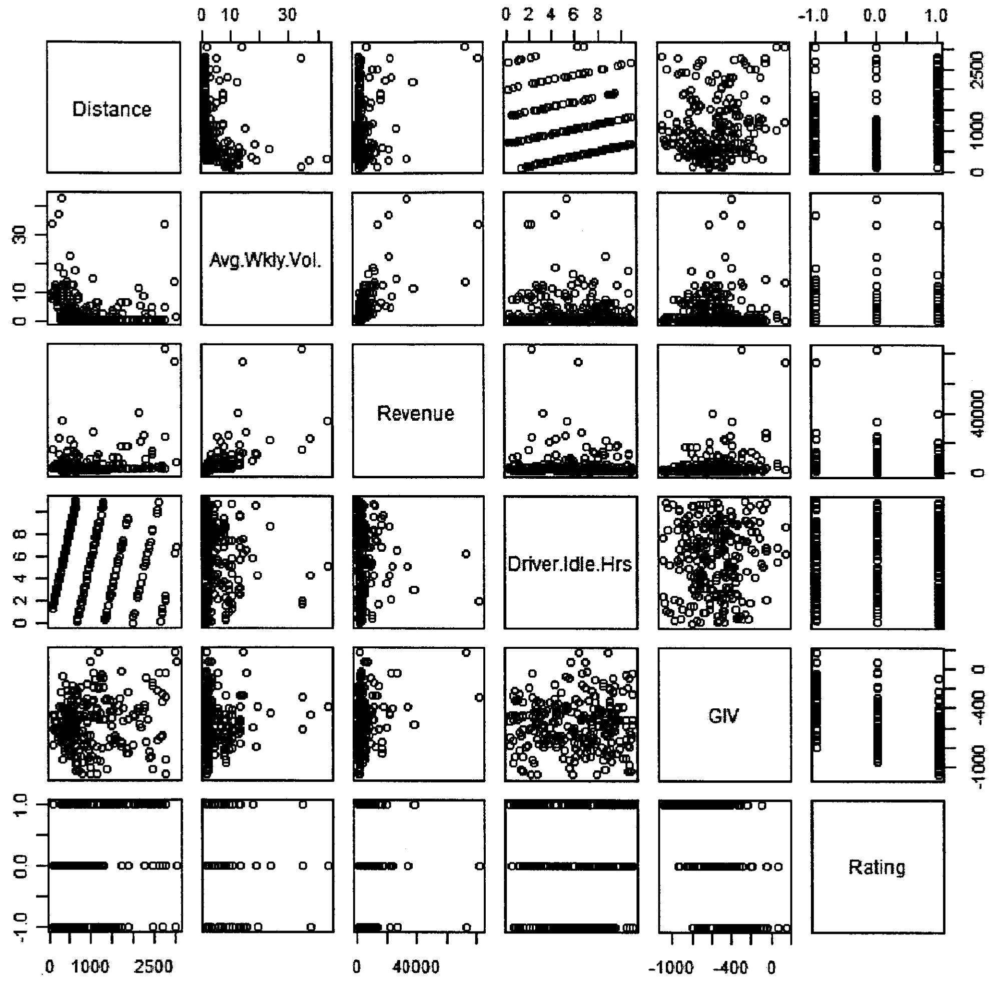


Figure 4-1: Expert-Classified Dataset: Bivariate Correlations

valued variable is within a certain range, 1 if within another range, etc. This was attempted with the Drive Idle Hours variable, but no clear improvements in classification performance were observed with the arbitrarily chosen cutoff of 5.5 hours. Future work may involve attempts to find good discretizations of this and other variables.

Figure 4-1 above shows the correlations between all pairs of features and features, as well as between features and the response. Besides Distance, Average Weekly Volume, and Revenue (the latter of which is derived from the former two), and between Distance and Driver Idle Hours (which is derived from Distance), no other pairwise correlations among the features and response are apparent.

4.2.2 Heuristic Method

Prior to investigating methods to extract patterns from the data, several heuristic classification rules were tested and improved upon via trial and error. This section outlines the approach taken in coming up with the best-performing heuristic method (Algorithm 4.2.1).

Initial Classification Based on Extremity of GIV Value

One way to classify the lanes is to first treat the geographic impact value of each lane, defined as the origin GIV plus the destination GIV, as an indicator of the degree of that lane's desirability with, in our case, smaller (negative) GIV's corresponding to greater desirability. Then, assuming that in general the proprtions of lanes that are undesirable, neutral, and desirable are fixed to $P(U)$, $P(N)$, and $P(D)$, respectively, a given lane can be classified according to how extreme that lane's GIV is compared to all possible lane GIV's. In particular, if the lane's GIV falls within the top $[P(U) \times 100]^{\text{th}}$ (e.g. if the proportion of undesirable lanes is 25%, the top 25th) percentile, the lane is classified as undesirable. If the GIV falls below the bottom $[P(D) \times 100]^{\text{th}}$ percentile, the lane is classified as desirable. Otherwise, the lane is classified as neutral. The two assumed distributions of lane desirability that were tested are an even split into thirds, and 25% undesirable and neutral and 50% desirable. The former was an *a priori* assumption, while the second was based on the observed proprtions in the dataset. Of course, other proportions are possible, and one can even find a set of proportions that minimize the misclassification error on the dataset. One concern, however, is how well this optimized set of proportions can generalize on new data.

Improving Classification Performance Based on Distance

To improve on the classification obtained from lane GIV's, each lane's initial class was modified based on the lane's distance. First, it was observed that an upper distance cutoff, above which any lane that was not already classified as desirable was promoted by one level (e.g. undesirable made neutral), improved classification performance. In addition, since intra-state/short haul lanes are often undesirable in practice, both an intra-state indicator (1 if the lane is intra-state, 0 otherwise) and lower distance cutoffs were tested. The final values of the upper and lower cutoffs were obtained by trial and error.

Final Heuristic Method

The best performing heuristic method is given as Algorithm 4.2.1. The method takes as an input a set of n lanes $L = \{\ell_t\}_{t=1}^{n}$ to be rated, where each lane has a GIV, Distance, and empty Class attribute (denoted by GIV_t, Distance_t, and Class_t), and returns the same set of lanes with updated classes.

```
Algorithm 4.2.1: HeuristicClassify($L$)

for $t \leftarrow 1$ to $n$
  do
    if $\text{GIV}_t < -589.14$
      then $\text{Class}_t \leftarrow 1$
      else if $\text{GIV}_t > -380.28$
      then $\text{Class}_t \leftarrow -1$
      else $\text{Class}_t \leftarrow 0$
    if $\text{Distance}_t > 1700$
      then
        if $\text{Class}_t < 1$
          then $\text{Class}_t \leftarrow \text{Class}_t + 1$
      else if $\text{Distance}_t < 250$
      then
        if $\text{Class}_t > -1$
          then $\text{Class}_t \leftarrow \text{Class}_t - 1$
return ($L$)
```

Note that what we have essentially done here is generate a classification tree by trial and error. Methods such as the ID3 and C4.5 algorithms can be used to generate classification trees. However, we were unable to generate a classification tree that outperformed the three methods described later in this chapter. In fact, this heuristic method turned out to be very competitive in terms of classification performance (see Table 4.1 in Section 4.2.4 for a summary of the performance of all tested methods) compared to the performance of the machine learning models presented in Section 4.2.3.

4.2.3 Learning a Classification Rule from the Data

This section considers three methods that can be used to find patterns in data and translate these patterns into a classification rule. In particular, the following methods are all particular approaches to the problem of *supervised learning*, which is concerned with finding an association between features (e.g. distance, revenue, etc.) and some response (e.g. the desirability of a lane) that agrees well with a given set of correct features and responses (e.g. the dataset with rated lanes obtained from the expert). There are many approaches to supervised learning (see any data mining or machine learning text); the three methods presented here appeared to work best out of those tested on the given dataset.

Multinomial Logistic Regression

Given a K-class classification problem (each object belongs to one of K classes) multinomial logistic regression estimates, based on the given already-classified data, a set of $K-1$ functions. Each of these functions f_k takes as inputs the features of an object we want to classify and returns the probability that the object belongs to class k. In other words, given an input vector of N features $\mathbf{x} = [x_n]_{n=1}^{N}$, each function returns the probability, conditional on this vector of features, that the object the features correspond to belongs to class c_k, i.e.

$$f_k = \Pr(c_k \mid \mathbf{x}), \quad k = 1, \dots, K-1,$$

Only $K-1$ functions are needed because, since any object we consider must belong to one of the K classes,

$$\sum_{k=1}^{K} \Pr(c_k \mid \mathbf{x}) = 1,$$

which means that once we know $K-1$ of the probabilities, we know the remaining probability is $1 - \sum_{k=1}^{K-1} \Pr(c_k \mid \mathbf{x})$. The class with the highest probability given $\mathbf{x}$ is then taken to be the class of the corresponding object.

To find these functions, multinomial logistic regression first assumes that they are of the form

$$f_k = \Pr(c_k \mid \mathbf{x}) = \frac{\exp\left(w_{k,0} + \sum_{n=1}^{N} w_{k,n} x_n\right)}{1 + \sum_{i=1}^{K-1} \exp\left(w_{i,0} + \sum_{n=1}^{N} w_{i,n} x_n\right)}, \quad k = 1, \ldots, K-1,$$

where $\exp(a) = e^a$, and then attempts to find good values of the parameters $w_{k,0}$, $\ldots$, $w_{k,N}$ for each class c_k based on the data. This is usually done using *maximum likelihood estimation*, which selects the set of parameters that makes observing the given dataset most likely given the assumed form of the conditional probability distribution. More precisely, given a set of M objects, each with an associated N-vector of given features, without being given the correct classes we don't know with certainty which class a given vector of features corresponds to. Hence we can model the class of each feature vector as a random variable C which, given the vector of features, has a (discrete) probability distribution, i.e. $f_{C|\mathbf{x}}(c_k) = \Pr(C = c_k \mid \mathbf{x})$. Since we're assuming a parametric form for this distribution, this is written more precisely as $\Pr(C = c_k \mid \mathbf{x}, W)$, where $W = \{w_p\}_{p=1}^{P}$ is the set of P parameters. We can then view each of the correct classes we're given for our set of objects as a realization (i.e. a "draw") from the corresponding conditional probability distribution.

The question that remains then is how to choose the values of the parameters in W. The approach taken by maximum likelihood estimation is based on the assumption that, since each object has a true class (given in the data), each probability distribution should "peak" at this class. Hence if we were to go through each of the feature vectors for our M given objects and draw from its conditional distribution we should, given the correct distribution, get the true class with very high probability. If we assume that knowing the result of a draw from one conditional distribution does not affect any of the other conditional distributions (which is reasonable since each object's true class should not change if we know the true value of some other object), the joint probability, or *likelihood* L, of getting the given dataset by making M conditional draws is

$$L(\mathbf{X}, \mathbf{r}, W) = \prod_{m=1}^{M} \Pr(C = r_m \mid \mathbf{x}_m, W), \tag{4.15}$$

where $\mathbf{X} = [\mathbf{x}_m]_{m=1}^{M}$ is the $M \times N$ matrix of given features and $\mathbf{r} = [r_m]_{m=1}^{M}$ is the vector of correct classes for each vector of features[2]. To make each actual class as likely to be drawn as possible, given the assumed parameterization of the feature-conditional probability distributions, we can find the set of values of the parameters in W that maximize equation (4.15). Since taking the logarithm of a function does not change the location(s) of its extreme point(s) (the maximum/minimum of a function will require the maximum/minimum power of any base to get that value), we can maximize the *log-likelihood* ℓ to work with a sum instead of a product[3]:

$$\ell(\mathbf{X}, \mathbf{r}, W) = \log L(\mathbf{X}, \mathbf{r}) = \sum_{m=1}^{M} \log \Pr(C = r_m \mid \mathbf{x}_m, W) \tag{4.16}$$

The log-likelihood is usually maximized iteratively using *gradient ascent*, since there is no closed-form (i.e. a set of equations were we can plug in $\mathbf{X}$ and $\mathbf{r}$ and get the optimal values of the parameters in W) set of solutions for the best values of the parameters in W [30]. In general, given a vector-valued function $f(\mathbf{x})$, where $\mathbf{x} = [x_i]_{i=1}^{n}$, gradient ascent works by first choosing some starting values $\mathbf{x} = \mathbf{a}_0$ for the variables we want to maximize our function with respect to, and taking those values as an initial guess for the optimum values. Then, assuming that the function is differentiable around

[2] If two events A and B are independent, then $\Pr(A \mid B) = \Pr(A)$ and $\Pr(B \mid A) = \Pr(B)$. Hence $\Pr(A \cap B) = \Pr(A \mid B)\Pr(B) = \Pr(A)\Pr(B \mid A) = \Pr(A)\Pr(B)$.

[3] Recall that $\log AB = \log A + \log B$. This is because, taking the base of the logarithm to be b, we have $AB = b^{\log AB}$. But since $A = b^{\log A}$ and $B = b^{\log B}$, $b^{\log AB} = (b^{\log A})(b^{\log B}) = b^{\log A + \log B}$.

our initial guess, we calculate the gradient

$$\nabla f(\mathbf{a}_0) = \left[\frac{\partial f(\mathbf{a}_0)}{\partial x_1}, \ldots, \frac{\partial f(\mathbf{a}_0)}{\partial x_n}\right]$$

at that point, which is a vector pointing in the direction in which $f(\mathbf{x})$ is increasing the fastest at $\mathbf{a}_0$, whose magnitude corresponds to the steepness of ascent. We then update our initial guess by moving it a certain number of steps η in the direction of the gradient, i.e.

$$\mathbf{a}_1 \leftarrow \mathbf{a}_0 + \eta \cdot \nabla f(\mathbf{a}_0).$$

We then set $\mathbf{a}_1$ as our new guess for the optimum, and repeat the process above until the difference between guess i and guess $i+1$ is small enough (i.e. $|\mathbf{a}_i - \mathbf{a}_{i+1}| < \epsilon$, for some small $\epsilon > 0$, which implies that the gradient at $\mathbf{a}_i$ is basically flat), and take $\mathbf{a}_{i+1}$ as the value of $\mathbf{x}$ that maximizes $f(\mathbf{x})$.

To estimate the values of the parameters in W that maximize (4.16), then, we initialize gradient ascent with some initial guess $\mathbf{w}_0 = [w_{0,p}]_{p=1}^{P}$ for the optimal parameter values, set some step size η, and update the initial guess according to the gradient of $\ell(\mathbf{X}, \mathbf{r}, W) = f(\mathbf{w})^4$ at $\mathbf{w}_0$:

$$\mathbf{w}_1 \leftarrow \mathbf{w}_0 + \eta \cdot \nabla f(\mathbf{w}_0) = \mathbf{w}_0 + \eta \cdot \left[\frac{\partial f(\mathbf{w}_0)}{\partial w_1}, \ldots, \frac{\partial f(\mathbf{w}_0)}{\partial w_P}\right].$$

A variant of gradient ascent called conjugate gradient ascent, which tends to converge more quickly, is used when computational efficiency is important [30].

Neural Network Classification

Neural network classification is an approach to estimating the function that maps feature vectors to one of the K possible classes that is inspired by certain aspects of how biological neural networks, e.g. brains, function. The most basic functional unit in a biological neural network is the *neuron*. Figure 4-2 shows the neuron's general structure. Each neuron can be viewed as a simple processing unit, and is connected

[4] The data $\mathbf{X}$ and $\mathbf{r}$ is taken as given, so the log-likelihood is a function only of the parameter values given by the vector $\mathbf{w}$.

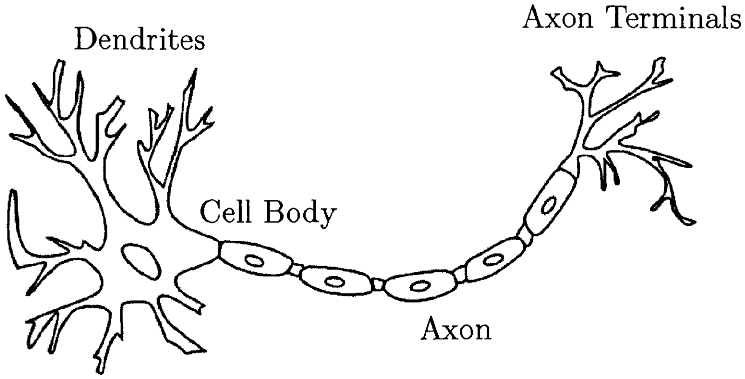


Figure 4-2: Biological Basis for the McCulloch & Pitts Neuron

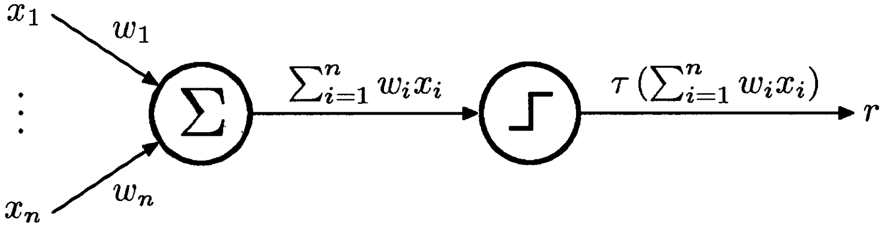


Figure 4-3: McCulloch & Pitts Neuron (Perceptron)

to other neurons via connections called *synapses.* The input to each neuron comes in the form of electrochemical signals from other neurons, which are conducted by the dendrites to the body of the cell. If the input signals are strong enough, i.e. beyond some threshold, the neuron itself generates a signal that is transmitted down along its axon and out to other neurons connected to it via its axon terminals. Computation in the brain, which contains on the order of 100 billion neurons and 100 trillion synapses [26], occurs in parallel over these simple processing units, in constrast to most modern personal computers, for which processing is centralized over at most only a few processors.

The mathematical model of the neuron which constitutes the basic building block of neural network classification is the McCulloch and Pitts neuron [27], or perceptron, shown in Figure 4-3 above. The input to the perceptron is an N-vector $\mathbf{x} = [x_n]_{n=1}^{N}$. The perceptron takes this input and first calculates the dot product of $\mathbf{x}$ with a stored weight vector $\mathbf{w} = [w_n]_{n=1}^{N}$, $\mathbf{w} \cdot \mathbf{x} = \sum_{n=1}^{N} w_n x_n$. The dot product is then fed to a threshold function τ, which is used to decide how the perceptron fires, i.e. what the response r will be. One common threshold function simply returns 1 if the dot product is greater than some threshold θ and 0 otherwise, i.e.

$$r \leftarrow \tau(\mathbf{w} \cdot \mathbf{x}) = \begin{cases} 1 & \text{, if } \mathbf{w} \cdot \mathbf{x} > \theta \\ 0 & \text{, otherwise.} \end{cases} \tag{4.17}$$

Another common threshold function is the sigmoid, which transitions smoothly from 0 to 1:

$$r \leftarrow \tau(\mathbf{w} \cdot \mathbf{x}) = \frac{1}{1 + \exp[-(\mathbf{w} \cdot \mathbf{x} - \theta)]}. \tag{4.18}$$

The value of the threshold θ can be represented as an input (or "bias") weight by creating an additional "zeroth" input x_0 whose value is always 1[5], and whose weight is denoted by w_0, i.e. $\theta = w_0 x_0 = w_0$. If this is done, the threshold function (4.17) simply checks the sign of the dot product of the input vector (with the additional element x_0) with the weight vector (with the additional element w_0). For the threshold function is (4.18), we simply let $\theta = w_0$.

A single perceptron can perform binary classification. In particular, given a set of weight values and a threshold value, if we use (4.17) the perceptron simply checks whether the input vector $\mathbf{x}$ is on one side of the hyperplane $w_1 x_1 + \cdots + w_N x_N = \theta$ or the other. If $\mathbf{x}$ is on one side, it is assigned to one class, and if it is on the other it is assigned to a second class. If we use (4.18), the output of the perceptron can be interpreted as the probability that $\mathbf{x}$ belongs to the class corresponding to an output of 1. Of course, in order for the single perceptron to be able to perfectly discriminate between the two classes, the classes must be linearly separable, i.e. if each object is represented in N-space, we can draw a hyperplane such that all objects of one class lie on one side, and those of the other class lie on the other.

Given M feature vectors $\mathbf{X} = [\mathbf{x}_m]_{m=1}^{M}$ and a vector of corresponding classes $\mathbf{t} = [t_m]_{m=1}^{M}$, $t_m \in \{0, 1\} \; \forall \, m$, the perceptron can "learn" the orientation of the hyperplane that best separates the two classes using Algorithm 4.2.2, where the weights $\mathbf{w}$ are randomly initialized to small values, η is the "learning rate" indicating how much we

[5] The sign of the bias input x_0 doesn't really matter, as long as its absolute value is 1. The point of expressing the threshold in this way is to have the perceptron learn the proper value of the threshold, given by the weight w_0, rather than setting it ourselves.

want to update the weights by when they're updated, and T is the desired number of iterations [26].

```
Algorithm 4.2.2: TRAINPERCEPTRON($\mathbf{X}, \mathbf{t}, \mathbf{w}, \eta, T$)

for $t \leftarrow 1$ to $T$
  do for $m \leftarrow 1$ to $M$
       do $r_m \leftarrow \tau(\mathbf{w} \cdot \mathbf{x}_m)$
          for $n \leftarrow 0$ to $N$
            do $w_n \leftarrow w_n + \eta \cdot (t_m - r_m) \cdot x_n$
```

The reasoning behind the update term $\eta \cdot (t_m - r_m) \cdot x_n$ is as follows. Suppose the object corresponding to the m^{th} feature vector belongs to the second class, i.e. $t_m = 1$. If the neuron's response r_m, given the m^{th} feature vector, is also 1, no weight update is necessary. If however $r_m = 0$, we need to change the weights in a way that will bring the neuron closer to giving the correct response of 1. Since the neuron's response is determined by the magnitude of the dot product $\mathbf{w} \cdot \mathbf{x} = \sum_{n=1}^{N} w_n x_n$, if we want to make the neuron go from a response of 0 to 1 we need to increase the value of the dot product. We can do this by updating each w_n so that $w_n x_n$ increases. The amount by which we change each w_n is determined both by the magnitude of the learning rate η as well as by the magnitude of x_n, and the direction of change (positive or negative) is determined by the sign of x_n. The reason we multiply by x_n is to move the dot product in the correct direction[6]. If $x_n > 0$, then we need to increase the value of w_n. On the other hand, if $x_n < 0$ we need to make w_n smaller in order to make $w_n x_n$ greater (less negative).

If $K > 2$ classes are involved, we can first encode the class of each object as a binary vector of length K, where the k^{th} element is 1 if the object belongs to class k

[6] Alternatively, we can replace x_n in the weight update term with $\text{sign}(x_n)$.

and 0 otherwise. Hence the vector of targets $\mathbf{t}$ above becomes an $M \times K$ 0-1 matrix $\mathbf{T} = [\mathbf{t}_m]_{m=1}^{M}$, where $\mathbf{t}_m = [t_k]_{k=1}^{K}$ encodes the class of the m^{th} feature vector. We can then assign a single perceptron to each of the K target positions, and train each using Algorithm 4.2.2. What we're essentially doing here is training each perceptron to recognize its assigned class, i.e. output 1 if the input vector belongs to its assigned class and 0 if it belongs to any other class.

Another approach to learning more than two classes is to start with a certain number of perceptrons, all of which share the same input vector $\mathbf{x}$, and connect the outputs of these perceptrons to an additional layer of perceptrons. The outputs of this second layer of perceptrons can then either be taken as the final output of the resultant *multilayer perceptron*, or connected to still more layers. However, in theory only two layers of perceptrons are required , since it can be shown [26] that a multilayer perceptron with only two layers can potentially approximate any function to an arbitrary accuracy. Training a multilayer perceptron, however, is more involved. The gradient descent backpropagation algorithm (see, for example, [26]) is a popular way to train multilayer perceptrons.

Ensemble Classification

As was noted at the start of this section, there are many different approaches to the problem of learning patterns from data besides multinomial logistic regression and neural nets. Each of these methods has different strengths and weaknesses, and often a method that works well on a particular dataset will perform poorly on another (and vice versa). In other words, different types of methods are better at extracting certain kinds patterns than others.

Given a dataset where the underlying pattern (e.g. the distribution of objects of different classes in the feature space) is very complex, no single learning method may be capable of finding that pattern on its own. If however we can assemble a group of classifiers that can each learn something different about the data, and properly combine what these classifiers learn into a composite classifier, it may be possible to find this complex pattern. This approach is called *ensemble classification*, and is analagous to doctors having a battery of medical tests done before reaching a final diagnosis and having a panel of experts over a single one. The power of any individual test or expert may be limited, and their results may vary from one to another, but taken together can give stronger and more consistent results.

The ensemble approach evaluated on the lane dataset is called bagging, which is short for bootstrap aggregating[7]. It involves first taking a number (e.g. 50-100) of bootstrap samples (samples with replacement) from the original dataset. Then a classification method with poor performance on its own is trained on each bootstrap sample. To classify a given feature vector, the vector is first classified by each trained weak classifier, i.e. each classifer "votes" on what it thinks the correct class is. The final class given by our ensemble is then taken to be the one with the most votes. The ensemble's vastly greater performance over the individual performance of any of its constituent classifers (see, for instance, [26] pp. 161) comes from both the bootstrap samples, which help ensure that each weak classifier sees a slightly different part of the data with greater emphasis (from duplicates) on certain aspects of the data, and the aggregation of results. Regarding the latter, suppose that each classifier has a success rate p that is greater than 50% (i.e. in a two-class problem, each classifier performs better than chance). Under majority voting, in order for the ensemble to output the correct class for a given feature vector, we need more than half of them to vote for the correct class. Letting L be the number of weak learners, the probability that $\lfloor L/2 \rfloor + 1$[8] or more of the weak classifers are correct is

$$\sum_{k=\lfloor L/2 \rfloor+1}^{L} \binom{L}{k} p^k (1-p)^{L-k},$$

which approaches 1 as L approaches infinity when $p > 0.5$ [26]. In other words, if for a two-class problem each classifier performs better than chance, as the number of weak learners increases the probability that the bagged ensemble classifier will give

[7] Another popular method called boosting, performed much worse than bagging on the lane dataset.

[8] Here $\lfloor a \rfloor$ is equal to the largest integer that is less than or equal to a, i.e. a with its decimal part removed. For example, $\lfloor 1.5 \rfloor = 1$.

the correct class approaches complete certainty. This however does not say anything about exactly how many weak learners will be needed for a given problem for the ensemble to perform perfectly.

Two bagged ensemble learners were constructed using the lane dataset. The first consists of 100 decision tree stumps (i.e. decision tress consisting only of a root node and terminal branches emanating from the root), while the second consists of 20 neural networks with 10 hidden layer nodes each and 20 decision tree stumps.

4.2.4 Classification Performance

Each of the three approaches mentioned above was used to fit one or more models, representing the patterns in the distribution of classes in the feature space that were extracted, based on that approach using the already-classified lane dataset. The important performance consideration for these models is whether they can be used with confidence to classify new lanes. This "generalization performance" was estimated for each model by first fitting the model using only a subset of the full lane dataset, called the *training set.* Then, we feed the feature vectors in the subset not used for training, called the *test set*, into the model, and compare the output class from the model for each feature vector to the vector's given correct class. The fraction of model outputs that are correct is then taken as an estimate of how well the model will do in general on unseen data. A total of 201 out of the 267 already-classified lanes were assigned to the training set, and the remaining 66 to the test set.

A total of four models were estimated – one multinomial logistic regression model, one 2-layer neural network, and two ensemble classifiers. The two layers in the neural network consist of a hidden layer in which each neuron has a sigmoid activation function, and an output layer in which each neuron simply outputs the sum of its inputs. The number of hidden layer nodes used – 84 – was decided on first by finding, for a number of hidden nodes from 1 to 100, the average performance of 10 neural networks with that number of hidden nodes on a subset of the training set called the *validation set*[9] The average performance is used because, due to the initial

[9] To get a better estimate of generalization performance, the data in the test set should not be

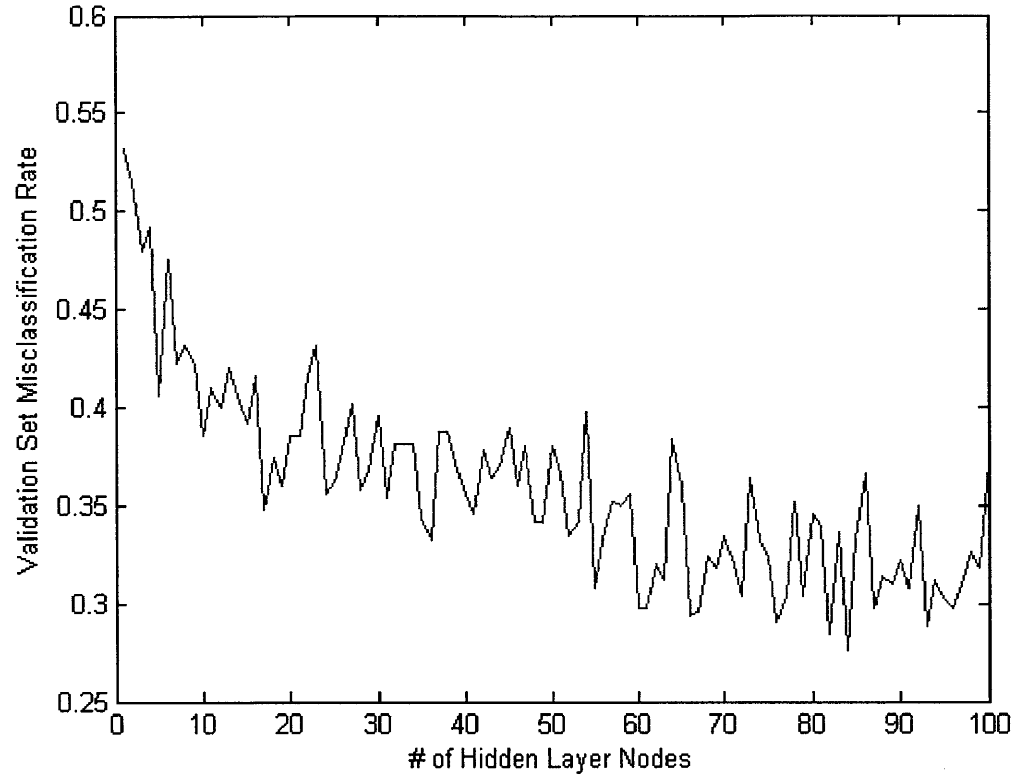


Figure 4-4: Average Neural Net Performance as a Function of Hidden Layer Size

randomization of weights, the performance of neural networks trained at different times will vary even if each shares the same training set. Figure 4-4 above shows how the 2-layer neural network's average performance changed as the number of hidden layer nodes was increased. The final number of hidden nodes to use was then taken to be the smallest number of hidden nodes that gave the least average misclassification rate.

The first ensemble classifier is a bagged ensemble consisting only of decision tree stumps. The number of decision stumps to use was obtained in the same way as the number of hidden nodes for the 2-layer neural network; namely by setting aside a validation set from the training set and evaluating the performance of ensembles trained with different numbers of decision stumps. Figure 4-5 shows the behavior of

used at all in either model training or selection. This is why a validation set, a subset of the training set, is held out and used to evaluate the performance of the neural networks with different hidden layer sizes.

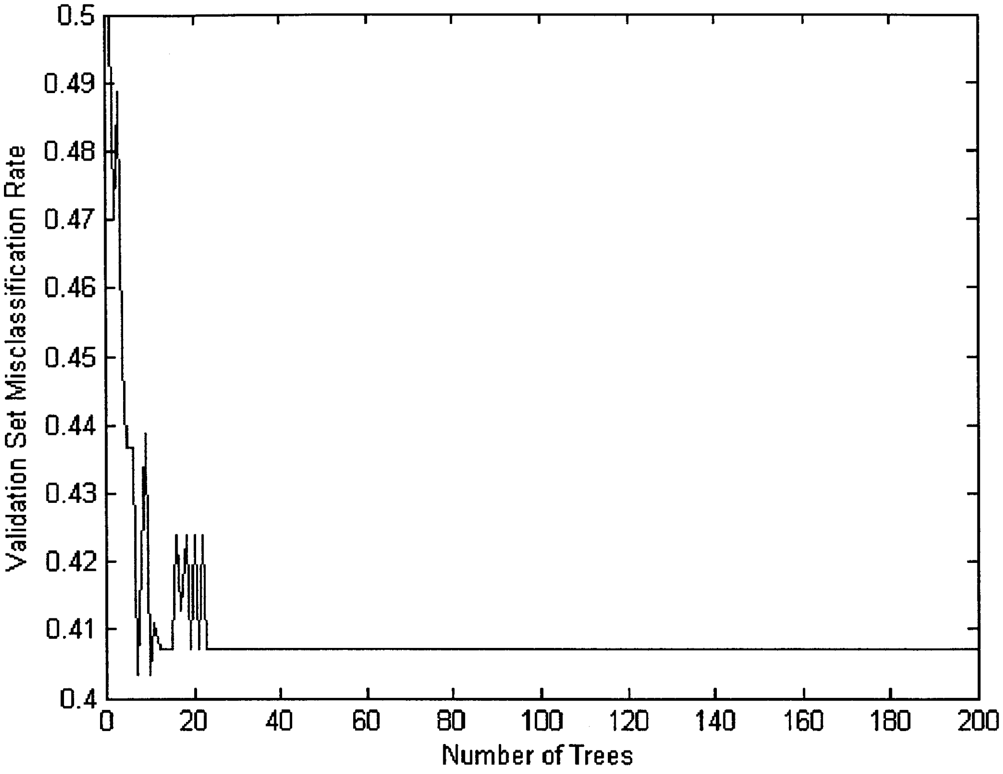


Figure 4-5: Bagged Ensemble Performance as a Function of the Number of Trees (Decision Stumps)

the misclassification rate as the number of stumps is increased. The final number of decision stumps – 40 – was chosen semi-arbitrarily given this behavior[10].

The second bagged ensemble consists of 20 2-layer neural networks, each with 8 hidden layer nodes[11], and 20 decision stumps. This combination seemed to give decent performance compared to the others; systematically checking different numbers of neural nets and decision stumps, as well as the use of other types of weak learners, is of course possible.

Table 4.1 summarizes the classification performance of the four models on the test set. Of these, the heuristic method and bagged ensemble using 40 decision

[10] After around 20 decision stumps, the misclassification rate levels off, but it seems better to choose a number greater than the point where it levels off, since the further away we are the further we are from the area where the ensemble's performance varies. A decision stump count of 40 was chosen because it gave fairly consistent performance.

[11] Arbitrarily chosen - all that was desired here was a weak nonlinear classifier for the present problem.

Model	% Correct	% Over-Rated	% Under-Rated
Heuristic Method	**74.24**	**16.67**	**9.09**
Multinomial Log. Reg.	65.15	19.70	15.15
2-Layer Neural Net	53.03	25.76	21.21
Bagging - 40 DS	**74.24**	**12.12**	**13.64**
Bagging - 20 NN, 20 DS	65.52	18.97	15.52

Table 4.1: Lane Classification Methods: Test Set Performance

stumps both performed similarly well, with the former having a greater bias towards over-rating than the latter towards under-rating. It is important to keep in mind, however, that the correct classification rates given in Table 4.1 are only estimates of how well the model will perform on data it was not trained on, based on a relatively small number of datapoints (recall the test set here contains 66 lanes). Also, these estimates pertain to the case where future objects to be classified are drawn from the same population as that of the objects used to build the models. In other words, the models are only valid for lanes with the same relationship between features and desirability as those used to build the models, i.e. the models' validity depends on the validity of the labels given in the training and test sets. The models also depend on the selection of features used to build them; in particular, it is possible that the errors in the models above are simply because one or more unidentified features, which are relevant to lane desirability, were left out. For the feature to be useful, however, it of course must be measurable.

The "validity" of the given labels, of course, may not be something that can be absolutely determined. As noted at the beginning of this chapter, experts may vary in how they rate lanes; based on their past experience and business knowledge, different factors may come into play, and the same factors may be weighed differently in arriving at the final rating. Here we assumed that the data were labeled correctly; determining whether the initial labelings themselves are correct is outside the scope of this thesis.

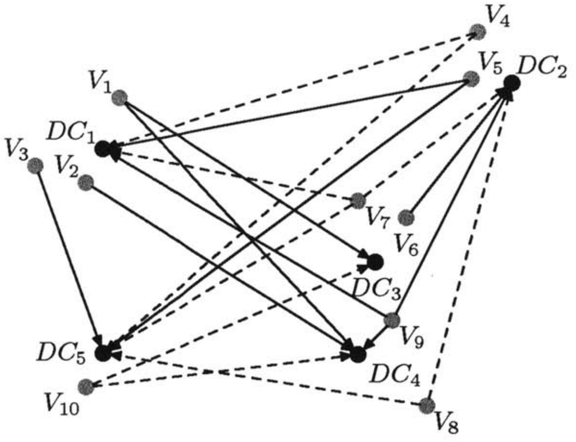


Figure 4-6: Allocation: Bin-Packing Scenarios 1 & 2

4.3 Toy Subnetwork Allocations

While it would of course be ideal to simultaneously balance bid value, location volumes, lane quality, and region-to-region volumes, in practice this may not be feasible. In particular, in order to get a solution it is likely that one will need to trade off balancing one attribute against balancing another. Here we will consider two scenarios; in the first, we focus on balancing DC volumes between bids, while in the second the focus is on balancing bid value.

4.3.1 Scenario 1: DC Volume Balance

In this scenario, the objective function coefficient corresponding to bid value balance, $W_{(b_i,b_j)}$, was set to 1, while the value of the coefficient corresponding to location balance, $K_{(b_i,b_j),\omega}$, was set to 15. The region-to-region balance coefficient $WR_{(b_i,b_j),(R_m,R_n)}$ was set to 15, and all other coefficients set to zero, in both scenarios. The resulting allocation is shown in Figure 4-6 above. Table 4.2 summarizes each bid. Both bids are fairly balanced in terms of number of lanes, value, and the number of desirable, neutral, and undesirable lanes between bids.

Table 4.3 gives the volumes assigned to each bid for each DC. The relatively large discrepancy for DC_3 is due to the fact that DC_3 only has two inbound lanes, one with a volume of 7 loads per week and the other with 16.

Bid	# Lanes	Tot. Value	# Desirable	# Neutral	# Undesirable
1	10	$ 21,379.40	6	1	3
2	9	$ 21,321.50	6	1	2

Table 4.2: Scenarios 1 & 2: Bid Statistics

DC	Bid 1 Volume	Bid 2 Volume
DC_1	2	6
DC_2	6	7
DC_3	7	16
DC_4	14	8
DC_5	7	7

Table 4.3: Scenarios 1 & 2: DC Volume Splits

Since each location in the toy subnetwork corresponds to a single region, and each lane's origin/destination pair is distinct, the region-to-region constraint does not affect the solution because each value of $\chi_{(b_i,b_j),(R_m,R_n)}$ will be the same regardless of how the lanes are allocated. In particular, the left hand side of each of the contraints of the form (4.14) will be the same regardless of the lane allocation because, in the toy subnetwork, "region-to-region" flows are all-or-nothing.

4.3.2 Scenario 2: Bid Value Balance

The only difference between this scenario and the one above is that the values of the bid value objective function coefficient and the origin balance coefficient are swapped. The resulting allocation, which is exactly the same as that obtained in Scenario 1, is also depicted in Figure 4-6, and Table 4.2 again summarizes each bid.

4.4 Summary

This chapter presented a linear (integer) programming approach to the allocation problem. Here the minimization of synergy loss was not accounted for explicitly, as it was in Chapter 3. Rather, the focus was on satisfying a set of constraints regarding the characteristics of each resultant bid, in particular the even distribution of lanes with respect to attributes such as value, desirability, and origin/destination. However, some of these constraints may be seen as, or can potentially help in, encouraging the preservation of certain synergies in the TL network. As was mentioned in Section 1.2.2, evenly distributing the volume inbound and outbound to a location between bids can have the effect of preserving the structure of the network (i.e. we get a scaled version of the location's flows in each bid), and hence any synergies at that location. Keeping region-to-region flows evenly distributed between bids, on the other hand, helps preserve flow relationships, in terms of both the existence of flow as well as the proportions of flows between regions, at the regional level. Hence one would expect region-level synergies arising from regional flows to be evenly distributed between bids. In fact, using a generalization of the synergy measures proposed in Chapter 3, the toy subnetwork allocation presented in this chapter is fairly competitive with the allocations obtained using the graph partitioning approach in terms of the total amount of synergy lost from separating the lanes into two bids (see Section 5.1 for the generalization, and Table 5.1 for the total synergy loss, for each synergy measure, associated with the bin-packing allocation).

The question of how to automate the process of rating lanes according to their desirability was also considered by exploring several methods related to estimating (either through trial and error or via machine learning methods) the human expert's thought processes in rating a lane. The performance of neither the heuristic method nor the machine learning methods considered, however, indicates that the relationship between the lane features and lane ratings hidden in the set of already-rated lanes has been found. While many other methods, derived from either trial and error or machine learning, potentially exist for this problem which were not explored for this thesis, for future work it seems equally if not more important to consider in more depth what sorts of features, and what subsets of these or other features, will make the hidden patterns more apparent. Of course, this problem of automating lane ratings will be solved if an agreed-upon checklist or flow chart defining how lanes are to be rated is developed.

THIS PAGE INTENTIONALLY LEFT BLANK

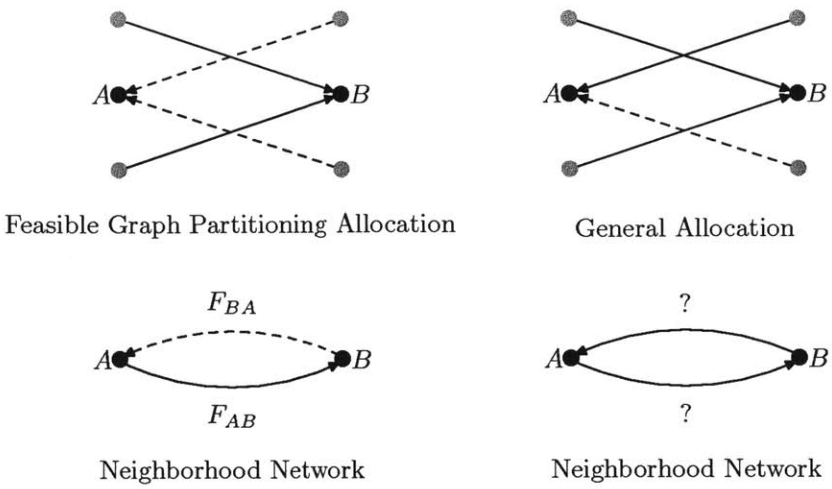
Chapter 5

Comparison of the Two Approaches

The two approaches presented in Chapters 3 and 4 to the allocation problem are compared in this chapter. In particular, we will first look at the performance of the bin-packing allocation in terms of the synergy measures presented in Section 3.1.1. Then, we will see how the best graph partitioning allocations fare in terms of the business contraints used in the bin-packing approach.

5.1 Synergy Preservation in the Bin-Packing Approach

In order to apply the synergy measures presented in Chapter 3, they first must be generalized to the case where not all lanes inbound to the same DC (i.e. neighborhood) are assigned to the same bid. In particular, we know how to calculate the synergy loss, using one of the three synergy measures, associated with separating all of the lanes inbound to DC A from those inbound to DC B, shown on the left in Figure 5-1. Under measure 1 the loss (cut size in the neighborhood network) is simply $F_{AB} + F_{BA}$, under measure 2 it is $\text{MIN}(F_{AB}, F_{BA})$, and under measure 3 it is $2 \cdot (F_{AB} + F_{BA}) - |F_{AB} - F_{BA}|$. The question that we must address in applying these synergy measures to allocations



Feasible Graph Partitioning Allocation

General Allocation

Neighborhood Network

Neighborhood Network

Figure 5-1: How Do We Apply Synergy to General Allocations?

obtained via the bin-packing approach is how to treat the case on the right in Figure 5-1, which is impossible under the graph partitioning approach (all lanes are assigned to their destination DCs) but perfectly possible under the bin-packing approach.

The synergy measures are generalized as follows. Consider the inbound and outbound lanes to a given neighborhood C. A loss in synergy is incurred whenever some outbound lanes are assigned to a different bid than some inbound lanes, the idea being that such a separation means that one or more follow-on opportunities for inbound lanes are lost. For a pair of neighborhoods D and E, under syergy measure 1 a synergy loss of F_{DE} is incurred in the D to E direction if the lanes inbound to D are assigned to one bid and the lanes constituting the flow from neighborhood D to E are assigned to another, and similarly for the E to D direction. Hence if the graph partitioning method cuts the edge between neighborhoods D and E, i.e. assigns the inbound lanes to D to one bid and the inbound lanes to E to the other, the loss associated with this neighborhood pair will be $F_{DE} + F_{ED}$. In other words, the loss under measure 1 for a given pair $\{i, j\}$ of neighborhoods is constructed as follows. Using only the lanes assigned to Bid 1, we write out the neighborhood flows in both directions between the neighborhood pair, and similarly for Bid 2. These flows are denoted by $F_{mn,k}$, the flow between neighborhoods m and n assigned to bid k. Then, we define the variable $\phi_{n,k}$ as the fraction of lane volume inbound to neighborhood n assigned to bid k, where for two bids $\phi_{n,2} = 1 - \phi_{n,1}$. Under the graph partitioning approach, this is always either 0 or 1. The synergy loss under measure 1 for neighborhood pair $\{i, j\}$ is then

$$\text{Loss}_{ij}(M1) = \sum_{k=1}^{2} [F_{ji,k} \cdot (1 - \phi_{j,k}) + F_{ij,k} \cdot (1 - \phi_{i,k})].$$

The total loss associated with a given allocation is then found by summing $\text{Loss}_{ij}(M1)$ over all neighborhood pairs $\{i, j\}$ in the network, i.e.

$$\text{Loss}(M1) = \sum_{\forall\{i,j\}} \text{Loss}_{ij}(M1),$$

which gives the same result in the graph partitioning approach as summing the weights of all cut edges in the neighborhood network. Similarly, the following generalizations of measures 2 and 3 give the same result as summing the weights of cut neighborhood network edges:

$$\text{Loss}(M2) = \sum_{\forall\{i,j\}} \{\underset{k\in\{1,2\}}{\text{MIN}} [F_{ji,k} \cdot (1 - \phi_{j,k}) + F_{ij,k} \cdot (1 - \phi_{i,k})]\}$$

$$\text{Loss}(M3) = \sum_{\forall\{i,j\}} \left\{ 2 \cdot \text{Loss}_{ij}(M1) - \left| \sum_{k=1}^{2} (-1)^{2n-1} [F_{ji,k} \cdot (1 - \phi_{j,k}) + F_{ij,k} \cdot (1 - \phi_{i,k})] \right| \right\}$$

Note that these are only valid for at most two bids; the consideration of allocation to more than two bids at once is beyond the scope of this thesis.

Table 5.1 gives the total synergy losses for the bin-packing allocation given in Section 4.3 calculated using the three loss functions given above. Interestingly, while the bin-packing approach does not explicitly attempt to minimize the defined synergy measures in allocating the lanes, the total losses associated with each measure are not significantly worse than those of any of the graph partitioning allocations, and are better than the worst graph partitioning allocations under each measure.

Synergy Measure	Total Loss	Avg. Graph Part. Loss
$M1$	41.58	37.2
$M2$	4.11	3.6
$M3$	49.79	48.6

Table 5.1: Bin-Packing Allocation: Synergy Losses

Allocation	Bid	# Lanes	Tot. Value	# Des.	# Neu.	# Und.
$M1$ MKL/Spec.	1	7	$ 17,246.10	7	0	0
	2	12	$ 25,454.80	5	2	5
$M2$ MKL/In. KL	1	10	$ 22,088.40	4	1	5
	2	9	$ 20,612.50	8	1	0
$M3$ MKL	1	11	$ 28,593.85	9	1	1
	2	8	$ 14,107.05	3	1	4
$M3$ Spectral	1	7	$ 17,246.10	7	0	0
	2	12	$ 25,454.80	5	2	5

Table 5.2: Best Graph Partitioning Allocations: Bid Statistics

5.2 Business Constraint Satisfaction in the Graph Partitioning Approach

We now consider how the best graph partitioning allocations based on each synergy measure fared in terms of the business constraints used in the bin-packing approach. The best allocation under measure 1 is taken to be the one generated using the multilevel-KL & spectral methods. The best under measure 2 is taken to be the allocation generated by multilevel-KL & inertial with KL. Finally, for measure 3 we will evaluate the allocations generated using the multilevel-KL and spectral methods.

The bid statistics for each of the above allocations are given in Table 5.2 above. Not surprisingly, all of the allocations are more unbalanced over # of lanes, value, and distribution of lanes by desirability. Also, since each DC's inbound lanes are constrained to be assigned to the same bid, there is no balancing happening whatsoever with respect to location volumes under these methods.

5.3 Lane Pairs Frequently Allocated Together

Another way to compare the two approaches, as well as allocations generated using the same approach, is to look at the frequency with which pairs of lanes are allocated to the same bid, and to see whether any common characteristics exist between lanes that are commonly, or never, allocated to the same bid. Based on the frequency with which each possible pair of lanes is allocated to the same bid, i.e. the number of distinct allocations under which each pair was allocated to the same bid, we can check how many of the possible pairs were in the same bid in none of the allocations, one of the allocations, etc. Five distinct allocations are considered here – the graph partitioning methods yielded a total of 4 distinct allocations, while the bin-packing approach yielded 1. The toy subnetwork includes a total of 19 lanes, so the number of lane pairs we consider here is $\binom{19}{2} = 171$. The plot in the top left corner of Figure 5-2 shows the number of pairs, out of all possible pairs, allocated in the same bid for each of the possible number of allocations. The majority of pairs were allocated to the same bid in one to four of the distinct allocations. At the extremes, fourteen pairs were never allocated to the same bid, while eleven pairs were allocated together in all 5 allocations. The characteristics observed for lane pairs allocated together a certain number of times include, for each lane pair, the average distance, value, and volume. The average values of these characteristics, for each number of times the pairs were allocated together, are also shown in Figure 5-2 in the top right (avg. pair average distance), bottom left (avg. pair average value), and bottom right (avg. pair average volume) plots. For distance and value, there is a weak upward trend – the lane pairs more frequently allocated together tend to have greater average distances and value than those less frequently allocated together. For volume, on the other hand, the trend is downward – the pairs most frequently allocated together have a lower average volume than those less frequently allocated together.

Figure 5-3 provides the same information as Figure 5-2, except only the distinct allocations obtained using the graph partitioning approach are considered. Here the majority of lane pairs were allocated together in either 1 or 3 of the 4 graph parti-

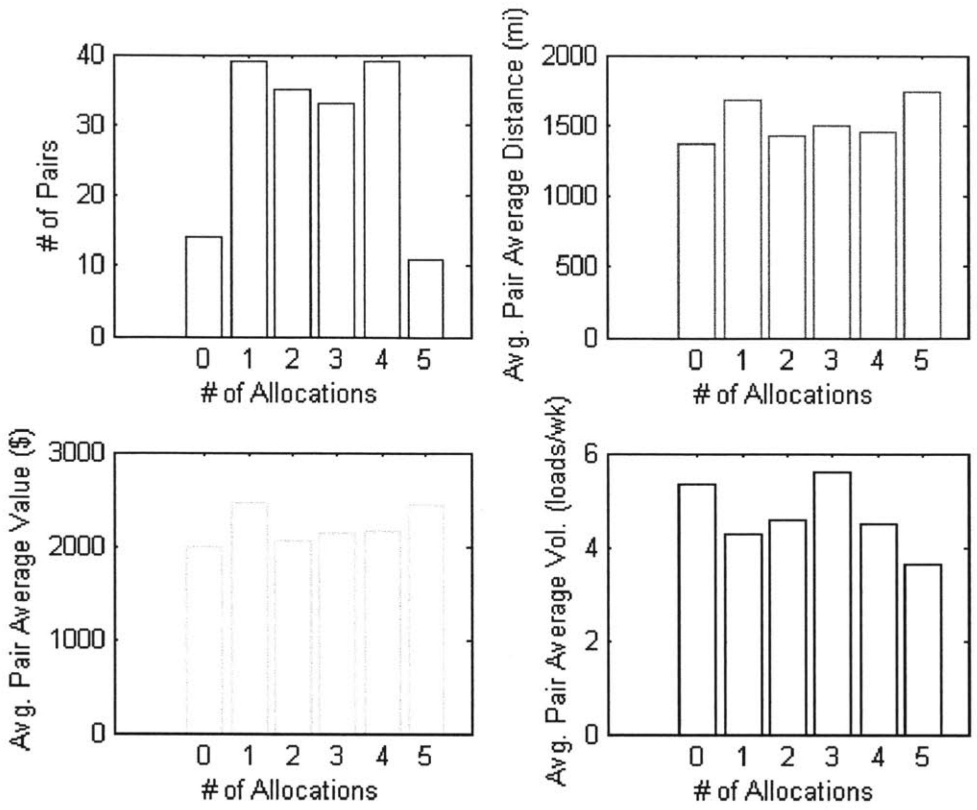


Figure 5-2: Pair Attributes By How Often Pairs are Allocated Together

tioning allocations (90 total out of 171). Among the graph partitioning allocations, the relationship between frequency of allocation to the same bid and average pair distance is weaker. For value, there is still a discernible upward trend, albeit a weak one. The downward trend for volume seems to persist here as well, despite a spike at 3 allocations.

Hence while there does seem to be a tendency for longer-haul, more valuable, and lower volume lane pairs in the toy subnetwork to be allocated together more frequently by the two approaches considered in Chapters 3 and 4, the trend is not a particularly strong one. This inconclusiveness may be due to the small number of distinct allocations obtained using the two approaches; on the other hand, the consideration of other characteristics may yield stronger relationships.

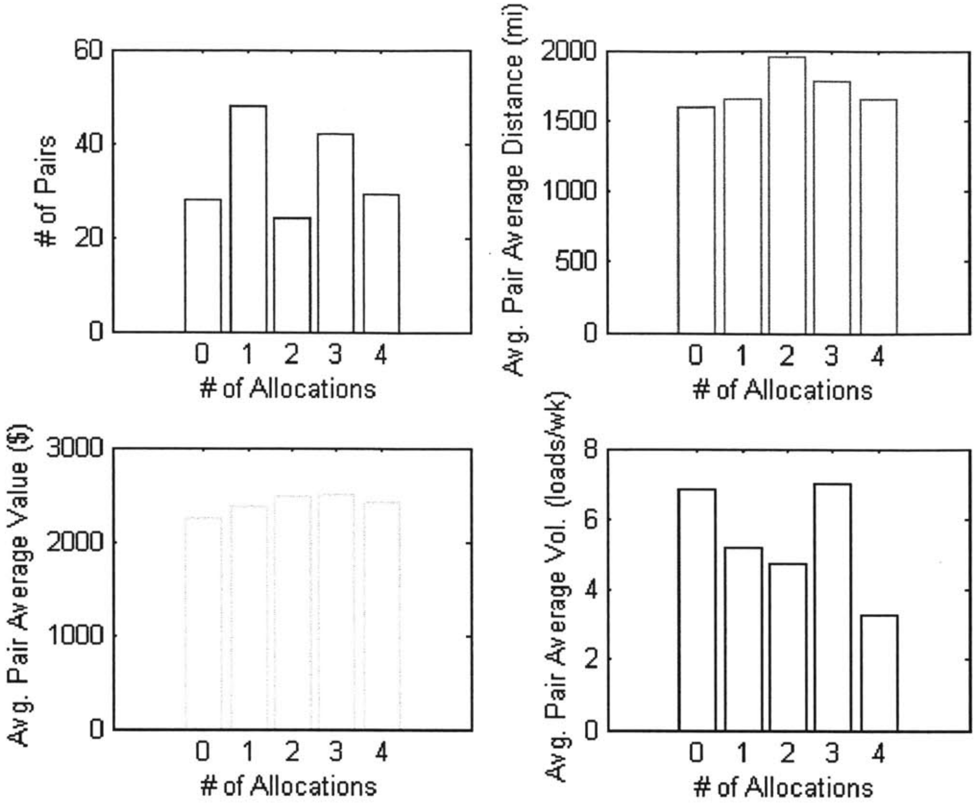


Figure 5-3: Pair Attributes By How Often Pairs are Allocated Together (Graph Partitioning Allocations Only)

5.4 Summary

This chapter presented several ways to compare the allocations produced by the two approaches proposed in this thesis. First, we considered how to apply the synergy measures developed for allocations generated using graph partitioning to allocations where the lanes inbound to a given DC do not all have to be allocated to the same bid. Based on the proposed generalizations of the three synergy measures, the synergy losses associated with each measure were calculated for the bin-packing allocation. Despite not considering these synergy measures explicitly, the bin-packing allocation did not perform much worse than the graph partitioning allocations with respect to synergy loss.

The graph partitioning allocations, however, produced much more unbalanced allocations, with respect to value balance, lane desirability balance, etc. , than the bin-packing allocation. In other words, the bin-packing approach produced an allocation that is comparable to those produced by the graph partitioning approach which, although not the best in terms of "synergy", represents a more even distribution of the toy subnetwork lanes over two bids with respect to characteristics which are likely important in practice for a large shipper.

Finally, an analysis of the frequencies with which pairs of lanes in the toy subnetwork were allocated together by the two approaches was attempted. While some trends were observed, with lane pairs more frequently allocated together tending to have greater average distance, greater average value, and lower average volume, none of these were especially strong and, considering the toy subnetwork's small size, should not be seen as general trends without caution.

Chapter 6

Conclusion

This chapter begins by providing a recap of the previous chapters. Then, based on the findings in these chapters, we provide a recommendation for large shippers faced with the problem of bidding out the lanes in its freight network in two or more pieces. Finally, we conclude by identifying some directions for further research.

6.1 Thesis Recap

Since the problem this thesis is concerned with – how a shipper should bid out its truckload (TL) lanes over time – rests on how bidding out lanes separately may affect carriers', and hence the shipper's, costs, we began in Chapter 2 by considering how truckload (TL) carriers operate. In particular, the point-to-point nature of TL operations, i.e. loads are shipped from their points of origin directly to their destinations, implies that it is up to the carrier to make trucks available where shipments are requested. This in turn means that the desirability of a load to a carrier depends not only on the revenue generated from the load itself, but on where the load is situated in time and space with respect to the carrier's existing service network. For example, a load for which the shipper pays a high linehaul rate may actually be seen as undesirable by a carrier if it does not have any trucks available within a reasonable distance for the load's departure time, or if it is known that it will be difficult to find additional loads for the carrier's trucks at the load's destination. On the other hand, the carrier may be willing to reduce its minimum linehaul rate for a lane that "complements" its existing network by providing follow-on opportunities for trucks moving across its service network. The availability of these follow-on opportunities mean increased chances for cost savings due to empty mile reduction for the carrier and in turn, since TL is predominantly a cost-plus business, reduced rates for the shipper.

In bidding out groups of the shipper's lanes that are complementary in the sense of providing follow-on opportunities to the carrier, either among themselves or with the carrier's existing lanes, we want the carrier's bid on these lanes to reflect the savings associated with these opportunities, i.e. the lowest rate possible for these lanes. A combinatorial auction setting, in which carriers can make their bids on certain lanes conditional on winning other lanes, can encourage carriers to bid their "true valuations" of groups of lanes, which take into account cost savings from empty mile reduction. Hence in such a setting we want to keep lanes which are highly complementary, which we term as having a large amount of *synergy*, in the same bid so that participating carriers can form conditional bids with these lanes, and in turn give the shipper a chance at achieving a lower cost allocation of lanes to carriers than may have been possible if carriers could only bid on lanes individually. In Chapter 3, we set out to define quantities which can serve as proxies for synergy, which we called *synergy measures.* Three such measures were proposed for a modified version of an inbound freight network called a *neighborhood network*, and graph partitioning methods were used to allocate the nodes of the toy subnetwork's neighborhood network to two bids. Each of these allocations were then translated into allocations of the individual toy subnetwork lanes to the two bids. These lane allocations were then evaluated based on their associated synergy loss under each of the three synergy measures.

Chapter 4 presented a different approach to the allocation problem. Rather than attempting to quantify synergies explicitly and minimize their loss, the focus here was instead on making the resulting allocation of lanes to bids as "balanced" as possible. Several forms of balance which may be desirable from the shipper's business perspective are presented, which include balancing the total value of lanes in each bid, volumes at the shipper's locations, the number of certain types of locations, the number of desirable/undesirable lanes, and region-to-region flows. To find allocations which are balanced in this manner, we proposed a linear (integer) programming/bin-packing model, and ran the model on the same toy subnetwork used in Chapter 3.

In Chapter 5, we then compared the two approaches by comparing the allocations that they generated. In particular, we found that, in terms of synergy as defined by our three measures, the allocation produced by the bin-packing approach in Chapter 4 was comparable to those obtained via graph partitioning. On the other hand, some of the graph partitioning allocations were much more unbalanced than the bin-packing allocation.

6.2 Recommendation

While minimizing the loss in "synergy" due to allocating the shipper's lanes to different bids is desirable from a theoretical point of view, especially if these bids will be combinatorial, the issue of how exactly one can best capture/measure inter-lane synergy is still an open question. In other words, it is not at all clear at this point that the measures of synergy proposed in Chapter 3 are actually useful in a practical sense with respect to giving allocations of lanes to bids that will help elicit lower rates from carriers. In particular, if the lanes will not be bid out in a combinatorial setting, one may not even have to consider synergy at all because, even if synergistic lanes are kept together, carriers will likely not bid based on the potential cost savings associated with these synergies because there is no guarantee that they will get all the lanes they need to actually realize these cost savings.

On the other hand, as mentioned in Section 4.4, while the bin-packing approach presented in Chapter 4 does not explicitly deal with synergy, some of the constraints can be seen as helping to preserve follow-on opportunities among the shipper's lanes being bid out. These constraints (location volume balance, region-to-region flow balance) help preserve the structure and flow proportions of the shipper's network across the separate bids to be run, and hence helps to keep synergies that may exist for the full network intact, albeit in a scaled down form, in each bid. In any event, at least based on the synergy measures presented in this thesis, the bin-packing allocation of the toy subnetwork's lanes to two bids entailed a sacrifice in synergy that was comparable to the graph partitioning allocations on average, while also providing more "balanced" allocations.

Hence for shippers faced with the need to bid out their freight network in pieces, we would at this point suggest focusing more on the sorts of constraints on bid characteristics that are beneficial to the shipper, rather than explicitly trying to deal with synergy. Inter-lane synergy has proved to be difficult to characterize in a succinct way, and its potential utility is only in encouraging, not guaranteeing, lower bids from carriers. This potential is also contingent on information about participating carriers' networks, i.e. it is not clear whether a bundle of the shipper's lanes, which seem to complement each other, will actually be of value to any bidding carriers. With a focus on business constraints, the shipper can at least be more sure of reaping the benefits of a "good" allocation, for example being able to adjust volumes at locations on a regular basis (a result of the location volume balance constraint). One important caveat to the bin-packing approach, however, is that it may not be possible to have perfect balance simultaneously over all types of balance considered. In addition, the behavior of the bin-packing model on larger (e.g. around 500 or more lanes) has not been tested.

6.3 Next Steps

There are a number of ways in which the work in this thesis can be extended. These include:

(1) **Lane Rating Automation** The performance of the lane desirability models in Chapter 4.2 indicate that, at least for the lane dataset we used, the relationship between lane attributes and desirability has not been found. Both further investigation into the nature of lane desirability from the expert at the shipper's standpoint, and a deeper exploration of statistical/machine learning methods,

will be needed before we can delegate the rating of lanes to the computer with any sort of confidence.

(2) **Explicit Synergy with Business Constraints** If an attractive measure of network synergy were found, it would be desirable to extend the bin-packing approach to take inter-lane synergies explicitly into account. Alternatively, or perhaps equivalently, one could attempt to create an algorithm that solves the graph partitioning problem with additional (business) constraints.

(3) **Up-scaling the Bin-Packing Model** Finally, in order to make the bin-packing model usable in practice, it of course needs to be able to actually handle the large number of lanes that large shippers will be bidding out. Modifications to the formulation may be necessary in order to make the model produce useful solutions.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix A

Full Bin-Packing Formulation

A.1 Model

Minimize $\sum_{(b_i,b_j):i<j}(W_{(b_i,b_j)} \cdot \sigma_{(b_i,b_j)}) + \sum_{\omega\in\Omega}\sum_{(b_i,b_j):i<j}(K_{(b_i,b_j),\omega} \cdot \nu_{(b_i,b_j),\omega}) +$
$\sum_{(b_i,b_j):i<j}(Q_{(b_i,b_j)} \cdot \rho_{(b_i,b_j)}) + \sum_{d\in\Delta}\sum_{(b_i,b_j):i<j}(WD_{(b_i,b_j)} \cdot \delta_{(b_i,b_j),d}) +$
$\sum_{(R_m,R_n)\in R\times R}\sum_{(b_i,b_j):i<j}(WR_{(b_i,b_j),(R_m,R_n)} \cdot \chi_{(b_i,b_j),(R_m,R_n)})$

subject to (1) $\sum_{b\in B} x_{l,b} = 1, \quad \forall\, l \in L$ (Lane Cover)

(2) $\sum_{l\in L} p_l x_{l,b} = S_b, \quad \forall\, b \in B$ (Bid Value Balance)

$S_{b_i} - S_{b_j} \leq \sigma_{(b_i,b_j)} \quad \forall\, b_i, b_j \in B$

$-(S_{b_i} - S_{b_j}) \leq \sigma_{(b_i,b_j)} \quad \forall\, b_i, b_j \in B$

(3) $\sum_{l\in L_\omega} x_{l,b} = N_{b,\omega} \quad \forall\, b \in B,\ \omega \in \Omega$ (Balance at Locations)

$N_{b_i,\omega} - N_{b_j,\omega} \leq \nu_{(b_i,b_j),\omega} \quad \forall\, b_i, b_j \in B,\ \omega \in \Omega$

$-(N_{b_i,\omega} - N_{b_j,\omega}) \leq \nu_{(b_i,b_j),\omega} \quad \forall\, b_i, b_j \in B,\ \omega \in \Omega$

(4) $\sum_{\mu\in \mathrm{M}} y_{\mu,b} = 1 \quad \forall\, b \in B$ (Balance of Locations)

$\sum_{\mu\in \mathrm{M}} y_{\mu,b} = T_b \quad \forall\, b \in B$

$x_{\ell,b} = y_{\mu,b} \quad \forall\, \ell \in L_\mu,\ b \in B,\ \mu \in \mathrm{M}$

$T_{b_i} - T_{b_j} \leq \rho_{(b_i,b_j)} \quad \forall\, b_i, b_j \in B$

$-(T_{b_i} - T_{b_j}) \leq \rho_{(b_i,b_j)} \quad \forall\, b_i, b_j \in B$

(5) $\sum_{\ell \in L_d} x_{\ell,b} = D_{d,b} \quad \forall\, d \in \Delta,\ b \in B$ (Lane Quality Balance)

$D_{d,b_i} - D_{d,b_j} \leq \delta_{(b_i,b_j),d} \quad \forall\, d \in \Delta,\ b_i, b_j \in B$

$-(D_{d,b_i} - D_{d,b_j}) \leq \delta_{(b_i,b_j),d} \quad \forall\, d \in \Delta,\ b_i, b_j \in B$

(6) $\sum_{\ell \in L_{(R_m,R_n)}} v_\ell x_{\ell,b} = Z_{(R_m,R_n),b}$ (Region-to-Region Balance)

$\forall\, R_m, R_n \in R,\ b \in B$

$Z_{(R_m,R_n),b_i} - Z_{(R_m,R_n),b_j}| \leq \chi_{(b_i,b_j),(R_m,R_n)} \quad \forall\, R_m, R_n \in R,\ b_i, b_j \in B$

$-(Z_{(R_m,R_n),b_i} - Z_{(R_m,R_n),b_j}) \leq \chi_{(b_i,b_j),(R_m,R_n)} \quad \forall\, R_m, R_n \in R,\ b_i, b_j \in B$

A.2 Index & Variable Definitions

Set	Includes All
B	Bids (Subsets of Lanes)
L	Lanes to be Allocated l
L_μ	Lanes Incident to Location μ
L_d	Lanes with Rating d
R	Regions
Δ	Distinct Lane Ratings
M	Locs. to Keep All Vol. in Same Bid
Ω	Locations to Split Vol. Btwn. Bids

Table A.2: Bin-Packing Formulation: Sets

Index	Indexes
b	Bids
b_i	Specific Bid i
d	Lane Ratings
l	Lanes
m	Regions
ℓ	Lanes with a Particular Characteristic
μ	Locations
ω	Locations where Vol. Split is Desired

Table A.3: Bin-Packing Formulation: Indices

Decision Variable	Definition
$x_{l,b}$	1 if Lane l is Allocated to Bid b, 0 otherwise
$y_{\mu,b}$	1 if Location μ is Allocated to Bid b, 0 otherwise

Table A.4: Bin-Packing Formulation: Decision Variables

Data Variable	Definition
p_l	Value of Lane l
v_ℓ	Volume on Lane ℓ

Table A.5: Bin-Packing Formulation: Data Variables

Coefficient	Cost per Unit of
$W_{(b_i,b_j)}$	Diff. Btwn. Bid b_i & b_j Tot. Values
$K_{(b_i,b_j),\omega}$	Diff. Btwn. # of Lanes in Bids b_i & b_j at Loc. ω
$Q_{(b_i,b_j)}$	Diff. Btwn. # of Locs. in Bids b_i & b_j
$WD_{(b_i,b_j)}$	Diff. Btwn. # of Lanes w/ Given Rating in Bids b_i & b_j
$WR(b_i,b_j),(R_m,R_n)$	Diff. Btwn. R_m to R_n Flow in Bids b_i & b_j

Table A.6: Bin-Packing Formulation: Objective Function Coefficients

THIS PAGE INTENTIONALLY LEFT BLANK

Bibliography

[1] Leonardo J. Basso and Sergio R. Jara-Diaz. Transport cost functions, network expansion and economies of scope. *Transportation Research Part E*, 39:271–288, 2003.

[2] Leonardo J. Basso and Sergio R. Jara-Diaz. Are returns to scale with variable network size adequate? *Transportation Science*, 40(3):259–268, 2006.

[3] William J. Baumol, John C. Panzar, and Robert D. Willig. *Contestable Markets and the Theory of Industry Structure.* Brace and Jovanovich, New York, 1988.

[4] Chris Burritt, Carol Wolf, and Matthew Boyle. Wal-mart asks suppliers to give up control of their deliveries. http://www.bloomberg.com/news/2010-05-21/wal-mart-asks-suppliers-to-give-up-control-of-their-deliveries-across-u-s-.html.

[5] Edward Capen, Robert Clapp, and William Campbell. Competitive bidding in high risk situations. *Journal of Petroleum Technology*, 23:641–653, 1971.

[6] Chris Caplice. 2008 model based benchmarking consortium truckload report. Chainalytics LLC Internal Report, 2008.

[7] Chris Caplice. Coping with economies of scope. *Journal of Commerce*, pages 58–59, June 2010.

[8] Chris Caplice and Yossi Sheffi. *Combinatorial Auctions for Truckload Transportation*, chapter 21, pages 539–571. Combinatorial Auctions. The MIT Press, Cambridge, Massachusetts, 2006.

[9] Christopher G. Caplice. *An optimization based bidding process: a new framework for shipper-carrier relationships.* PhD dissertation, Massachusetts Institute of Technology, Civil and Environmental Engineering, 1996.

[10] James Demmel. Cs267: Lectures 20 and 21, mar 21, 1996 and apr 2, 1996 – graph partitioning, part 1. http://www.cs.berkeley.edu/~demmel/cs267/lecture18/lecture18.html.

[11] Ozlem Ergun, Gultekin Kuyzu, and Martin Savelsbergh. Reducing truckload transportation costs through collaboration. 2007.

[12] C.M. Fiduccia and R.M. Mattheyses. A linear time heuristic for improving network partitions. *Proc. 19th IEEE Design Automation Conference*, pages 175–181, 1982.

[13] Michael R. Garey, David S. Johnson, and Larry Stockmeyer. Some simplified np-complete graph problems. *Theoretical Computer Science*, 1(2):525–530, September 1976.

[14] Keith D. Gremban, Gary L. Miller, and Shang-Hua Teng. Moments of inertia and graph separators. 1994.

[15] Brian Hayes. Computing science: The easiest hard problem. *American Scientist*, 90(3):113–117, 2002.

[16] Bruce Hendrickson and Robert Leland. Multidimensional spectral load balancing. *Proc. 6th SIAM Conf. Parallel Proc. Sci. Comput.*, 1993.

[17] Bruce Hendrickson and Robert Leland. *The Chaco User's Guide Version 2.0.* Sandia National Laboratories, Albuquerque, NM 87185-1110, July 1995.

[18] Bruce Hendrickson and Robert Leland. An improved spectral graph partitioning algorithm for mapping parallel computations. *SIAM Journal of Scientific Computing*, 16(2):452–469, 1995.

[19] Bruce Hendrickson and Robert Leland. A multilevel algorithm for partitioning graphs. *Proc. of Supercomputing*, 1995.

[20] Jefferson Huang, Chris Caplice, and Francisco Jauffred. Working paper. 2011.

[21] Sergio R. Jara-Diaz. *Transportation cost functions: a multiproduct approach.* PhD dissertation, Massachusetts Institute of Technology, Civil and Environmental Engineering, 1981.

[22] E. William Moore Jr., Janice M. Warmke, and Lonny R. Gorban. The indispensable role of management science in centralizing freight operations at reynolds metals company. *Interfaces*, 21(1):107–129, 1991.

[23] Brian Kernighan and Shen Lin. An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*, 29:291–307, 1970.

[24] Lane Kidd. Less is more: Chris sultemeier prepares walmart transportation to double freight volume. *Arkansas Trucking Report*, pages 24–33, April 2010.

[25] Kevin Kirkeby. Transportation: Commercial. Standard & Poor's Industry Surveys, January 2011.

[26] Stephen Marsland. *Machine Learning: An Algorithmic Perspective.* Chapman & Hall/CRC, Boca Raton, FL, first edition, 2009.

[27] Warren S. McCulloch and Walter H. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.

[28] Ralph C. Merkle and Martin E. Hellman. Hiding information and signatures in trapdoor knapsacks. *IEEE Transactions on Information Theory*, 24(5):525–530, September 1978.

[29] Stephan Mertens. *The Easiest Hard Problem: Number Partitioning*, chapter 5, pages 125–139. Computational complexity and statistical physics. Oxford University Press, New York, NY, 2006.

[30] Tom M. Mitchell. Generative and discriminative classifiers: Naive bayes and logistic regression. Draft Chapter of Second Edition of Machine Learning, T.M. Mitchell, McGraw Hill, http://www.cs.cmu.edu/~tom/NewChapters.html, January 2010.

[31] Michael J. Mulqueen. Creating transportation policy in a network that utilizes both contract carriers and an internally managed fleet. Master of engineering in logistics thesis, Massachusetts Institute of Technology, 2006.

[32] Tae H. Oum and W.G. Waters II. A survey of recent developments in transportation cost function research. *Logistics and Transportation Review*, pages 423–463, 1996.

[33] John C. Panzar. *Technological determinants of firm and industry structure*, pages 3–59. Handbook of Industrial Organization. North-Holland, New York, New York, 1989.

[34] Clinton L. Plummer. Bidder response to combinatorial auctions in truckload procurement. Master of engineering in logistics thesis, Massachusetts Institute of Technology, 2003.

[35] Warren B. Powell, Arun Marar, Jack Gelfand, and Steve Bowers. Implementing real-time optimization models: a case application from the motor carrier industry. *Operations Research*, 50(4):571–581, 2002.

[36] Walmart Transportation. Walmart transportation: Statistics. http://walmartprivatefleet.com/CompetitiveAdvantage/Stats.aspx.

[37] John Tsu and Mayank Agarwal. Use of transportation relays to improve private fleet management. Master of engineering in logistics thesis, Massachusetts Institute of Technology, 2009.