On Ranking the Relative Importance of Nodes in Physical Distribution Networks

by

Christian Filion

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Master of Applied Science
in
Management Sciences

Waterloo, Ontario, Canada, 2011

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Physical distribution networks are integral parts of modern supply chains. When faced with a question of which node in a network is more important, cost immediately jumps to mind. However, in a world of uncertainty, there are other significant factors which should be considered when trying to answer such a question. The integrity of a network, as well as its robustness are factors that we consider, in making a judgement of importance.

We develop algorithms to measure several properties of a class of networks. To accelerate the optimization of multiple related linear programs, we develop a modification of the revised simplex method, which exploits several key aspects to gain efficiency. We combine these algorithms and methods, to give rankings of the relative importance of nodes in networks.

In order to better understand the usefulness of our method, we analyse the effect parameter changes have on the relative importance of nodes. We present a large, realistic network, whose nodes we rank in importance. We then vary the network's parameters and observe the impact of each change.

Acknowledgements

I would like to thank Professor James Bookbinder, for his guidance and wisdom throughout the writing of this thesis. A special thanks goes to Tiffany Matuk for her advice.

Dedication

This is dedicated to Sheena and Max. To Sheena for all the love and support one can ask for. To Max for his friendship and unwavering belief in me.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Distribution networks exist in all manner of companies, from large multinational retail conglomerates, to local pizza delivery franchises. The purpose of each network is simple: bring goods from one location to another. In an ideal world, all distribution networks would be designed according to the best models and practices available at the time. There exist very robust network design models that can help answer questions such as: "How many distribution centres should you have?" and "How should we schedule replenishment of inventory?". With the aid of such models, it is possible to build and manage even the largest networks in an efficient and cost-effective manner. Unfortunately, we do not live in the ideal world, and the realities of expanding and contracting distribution networks can lead to sub-optimal network configurations.

Were we to have an ideal network, however, a logical question to ask would be: "Which of our facilities is important to us?". A naive answer would be those which generate the most profits. However, considering only profits is not wise. When building our networks, we took account not only of costs but locations, sizes, schedules, among a variety of factors.

Reducing our definition of ”importance” to cost thus ignores those factors, and insights we used while designing our networks.

Pertinent answers to the preceding question would permit network managers to make more informed decisions with regards to network integrity, reliability, as well as longer term strategic decisions. An example of such a long term strategic decision could be to choose which of two possible facilities to close in a supply chain. Given an accurate measure of importance, the choice could be made to close one and incur a much higher cost of transportation for an interval of time, knowing that after that period, a possible expansion of the network would be better served by temporarily keeping the less profitable facility open. Such a decision to incur short term losses in light of a long term gain could not be made with an importance measure that considered only the cost of removing a node. Our goal, then, should be to find a more insightful answer to the question of importance.

We present, as an example, an imaginary pizza restaurant that focuses specifically on delivering food to customers. Our hypothetical company would have several locations, spread throughout its operating area, whether it be a whole city or simply a suburb. We can equate a single ordered pizza to one unit of product, for simplicity. The general procedure, then, would be for the customer to place the order with a central ordering service, which would route the order to the appropriate location. That location would then create the order and hand it to one of its delivery drivers, who would see the order handed to a customer.

The questions would then be ”How important are each of our locations?” and ”How important are each of the districts that we serve?". This thesis aims to give a method to answer those questions, for any distribution network, while considering aspects and subtleties of the networks studied. The following section gives an overview of research in network reliability and node importance, as well as other related considerations. A formal problem statement, as well as the mathematical solution methodology, will then be presented. We will conclude with applications, results and areas for further research.

Chapter 2

Literature Review

There does exist research into the importance of nodes in networks. However, most of the papers focus on information networks, or traffic networks. We present an overview of this branch of research, which is closely related to network reliability, for which we also give an overview. In this thesis, we study the importance of nodes in *physical distribution* networks. These are distinct from those studied in the literature. The methods already presented are thus not directly applicable to our subject matter, although they inform our decisions. We conclude our overview of existing research with some additional considerations.

2.1 Node Importance

Node importance is an active field of research, especially where it pertains to "complex" networks. A network is said to be "complex" when its topological features are non-trivial (i.e. they do not resemble lattices or random graphs). The physical distribution networks we study are not complex, since their structure is simple (although not trivial). The main reason these networks are currently studied in such great detail is that most computer and social networks are complex networks. Dafermos (1982) gives formulations and solution methods for a general class of complex networks.

Newman (2003) gives a very thorough review on the state of research into complex networks. Most of the research is specific to complex networks, and as such is not entirely relevant to our topic, due to our "simple" networks. However, when discussing properties of networks, Newman mentions the concept of "betweeness centrality". This is a measure of centrality for a node in a complex network. In essence, "betweeness centrality" measures, for a specified node A, the number of geodesic paths between all pairs of nodes which pass through A. In our case, the simplified network structure we study lends itself to an interpretation of the centrality of a node which counts the number of paths contributed to the total. While Newman's complex networks can see large changes in average path lengths when certain nodes are removed, we simply see a reduction in number of paths present. Regardless of the differences, the concept is transferable. This, along with other concepts, suggests that the number of paths is an important aspect of a network.

Any discussion of importance with regards to computer networks mentions, first and foremost, Brin and Page (1998), which famously gives the PageRank algorithm used by Google. Page et al. (1999) further clarify the algorithm which lies as the foundation of all searches made on the Google search engine. The main idea behind PageRank is that the nodes which are important are those who receive links from other important nodes: A link from an important node is thus more meaningful than a link from a less important one. The PageRank algorithm assigns a rank to all the nodes based on the importance of its neighbours. Interestingly, if node A has only one incoming link, but this link is from a very important node, then node A is given a very high value of importance, based solely on that single link. This measure of importance is appropriate for *information* networks where the relevance of information is likely to be gauged by the rate at which other information refers to it. However, distribution networks do not have the reciprocal properties of complex networks, and thus PageRank is not applicable.

Lianxiong et al. (2009) give an adapted PageRank algorithm for measuring the importance of nodes in traffic networks. They propose using connectivity and traffic flows inherent in road networks to measure the relative importance of intersections. Similarly to the base PageRank, the bi-directionality of flow on edges is a key component of the importance measure. That bi-directionality means that such an adaptation of PageRank will not suit our type of network. (An additional example of PageRank adaptations can be seen in the work of Dwyer (2007), where the author studies the effect of word-of-mouth in marketing.)

Importance measures unrelated to PageRank have been studied in works such as White and Smyth (2003), Hawick and James (2007), as well as Le and Hewei (2010). The authors use many graph theoretic methods to compute importance measures for nodes in complex networks. Of most interest are Hawick and James' use of mean degree and Dijkstra's all pairs measures, which compute, respectively, the average degree of each node and the average length of the shortest path for all pairs of nodes in the network. These two measures strongly suggest that connectivity is significant for transportation networks, which leads us to incorporate a similar measure in our own definition of importance.

Further evidence of the significance of connectivity in networks is given by Kobayashi et al. (2009), who apply path counting algorithms to measure the importance of nodes and edges. In counting the number of paths that are no longer valid when a node is removed, Kobayashi et al. are measuring the "robustness" of a network, based on each individual node's contribution to the whole. Although those authors are limited with respect to counting paths in undirected networks, we can enumerate with much greater ease the paths of a directed network. Section 3.2 deals with counting such paths.

In addition to connectivity, the concept of the damage done to a network is one which we consider important. Li and Li (2004) study the *integrity* of graphs, which measures the difficulty of separating a graph and the extent of the damage done to the graph once separated. This concept, integrity, informs our decision to measure the disruptions created when removing a node from a physical distribution network. We call such a disruption a "rerouting", since the flow of goods through the removed node must be routed to new nodes to satisfy demands.

2.2 Network Reliability

Kelleher (1991), gives a broad overview of communication network reliability, as well as a very detailed primer on graph theory concepts used in many models and analytical methods. It is important to note that Kelleher focuses on the graph properties of information networks, with the major concern that the whole network remain connected when any nodes or edges are removed. In supply chains, connectivity is essential to maintaining service to customers. This suggests a high level of importance for nodes which lead to disconnected graphs when removed from the network.

The research of Antikainen et al. (2009) gives an analysis of network reliability for power distribution networks, and proposes a method for minimizing the negative impacts of failures. This research focuses on the costs incurred when a failure is present in the network infrastructure, be it a node or arc. The proposal to intentionally fragment the network into “islands” presents an insight into the usefulness of redundancy in networks. Further to this idea, Carvalho and Ferreira (2004) present a model for power distribution that considers the cost of *reliability*, in addition to the regular infrastructure costs, when planning a distribution network. The inclusion in their model of un-reliability as a separate penalty function reinforces our notion that there are important aspects of networks which cannot be measured directly by cost, but which can be included in our decision making by approximating them with a cost. Whereas Carvalho and Ferreira choose to incorporate the approximate cost of reliability into their optimization model, we present methods with which to calculate the factors that contribute to reliability, and use the result to give a ranking of importance.

Qiang and Nagurney (2008) give a unified measure of performance and importance in networks, with an eye on their vulnerability. The networks studied are closely related to our physical distribution networks in that there are distinct origin-destination pairs. However, Qiang and Ngurney study networks where the cost of an arc is dependent on the flow along that arc. The assumption is that each origin and destination node is an individual entity whose goal is to maximize its own utility. In contrast, our networks are assumed to be owned by a single decision maker, who can make sub-optimal incremental decisions if they lead to a globally optimal solution. Nagurney (2006) gives a method for converting supply chain networks into transportation networks, which are themselves complex networks, but our physical distribution networks are not convertible in such ways.

2.3 Additional Considerations

Min and Zhou (2002) give a review of supply chain modelling. They present the development of supply chain models that integrate all echelons of a supply chain, as well as the exchanges of information that take place in networks. It is important to note that our distribution network model resides in what Min and Zhou call "Outbound Logistics," since removing the inbound portion of the supply chain eliminates many stakeholders, and greatly simplifies decision making. We assume that our models deal only with goods that are already owned by the company, and thus we can ignore the requirements and constraints of third party suppliers.

Hawick (2007) gives important insight into algorithms and computational tools available when handling and evaluating graph properties. Of particular interest is Hawick's algorithm for computing the Dijkstra all-pairs distance. Insights gained from this code are used in our algorithms from Chapter 3.

The consensus, then, is that the importance of a node is not strictly tied to its cost, and that other important factors are reliability and integrity of a network. We can measure reliability in a network by counting paths. For integrity, measuring the disruptions created by removing nodes is a good indicator. Chapter 3 gives algorithms to calculate these factors, as well a method to rank the importance of nodes in a given network.

Chapter 3

Methodology

This chapter will deal with all of the models, algorithms and methods used to establish our importance measure. We start by defining the basic model. The next sections deal with specific algorithms to count paths, optimize sub-networks and count re-routings.

3.1 Directed Distribution Network Model

Suppose that a company has a set of customers, a set of inventory handling facilities (be they warehouses, cross-docks or otherwise), and a set of source nodes. We arrange these sets into a directed distribution network, such that goods travel from source nodes to facilities and then to customers. It is the case that some networks will have multiple intermediate layers of facilities, meaning that goods from the source node will travel through more than one facility before arriving at the customer. To avoid needlessly-complicated exceptions, we enforce a few restrictions on the networks.

- A source node may not be the end point of any arc.

- There can be no directed cycles in the network.
- The net demand by an intermediate facility is zero.

The lack of cycles is why we call the networks directed, since the flow of goods is directed from source nodes to customers. An arc between two nodes indicates that there exists a shipping link between them; shipping costs are therefore modelled as arc costs. We include arc capacities to reflect the fact that there is a limit on the amount of flow that can be shipped between any two nodes. Similarly, we assume that a maximum amount of flow can pass through any given facility. Note that in our model we do not include fixed costs for any facility being open, since we assume that the networks we are studying are currently in use, and therefore all facilities are in use. We consider this cost later when discussing importance measures. The following are the list of sets, parameters and decision variables in our model.

Sets:

- N : the set of nodes
- E : the set of arcs

Parameters:

- d_i : supply or demand at node i. This takes a negative value if the node is a supply node, a positive value if it is a demand node, and 0 if it is an intermediate node.
- w_i : capacity of node i
- f_i : cost of node i

- m_{ij} : capacity of the arc linking node i to j
- c_{ij} : cost of one unit of flow going through the arc ij

Decision Variables:

- x_{ij} : quantity of flow along arc ij

This gives us the following linear programming model, LP1.

LP1:

$$\text{min}: \quad z = \sum_{pairs\ i,j} c_{ij}x_{ij} \tag{3.1}$$

$$\text{Subject to:} \tag{3.2}$$

$$\sum_k x_{jk} - \sum_i x_{ij} + d_j \leq 0 \qquad \forall j \tag{3.3}$$

$$\sum_i x_{ij} - w_j \leq 0 \qquad \forall j \tag{3.4}$$

$$x_{ij} - m_{ij} \leq 0 \qquad \forall i, j \tag{3.5}$$

$$x_{ij} \geq 0 \qquad \forall i, j \tag{3.6}$$

A few assumptions are made that need to be explained in order to clarify this model. For (3.1), we assume that there is no cost to process an item at the customer node, thus we only count outgoing flow when calculating facility handling cost. In (3.3), we have negative supply and positive outgoing flow. Nodes thus never send more flow than their supply. We also enforce incoming flow to be greater than the demand plus the outgoing flow, so that constraints on demand, as well as flow conservation, are met. (3.3) is important for intermediate nodes, and may be binding in that case. (3.3) is non-binding for a pure supply or demand node, whose capacity is assumed to be sufficient to handle the respective supply or demand.

Note that this is not a proper transhipment problem. We can easily formulate the problem as a transhipment problem by expanding each existing node into two nodes connected by an arc. By assigning the old node's capacity to the arc, we create a proper transhipment problem. For our purposes, we continue to use the LP1 formulation, however we acknowledge the benefits of using a proper transhipment formulation. These benefits are discussed further in Section 5.2.

Figure 3.1 is a simple directed distribution network.

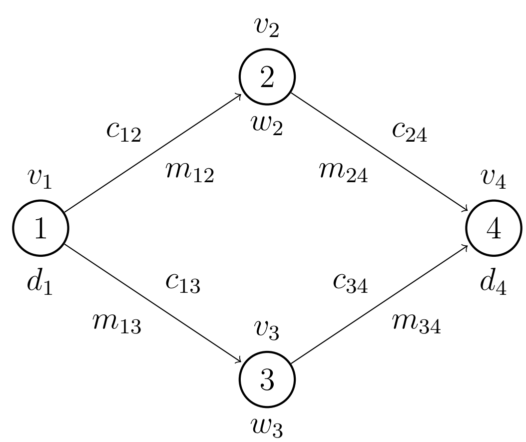

Figure 3.1: A simple network

Above each arc is its cost, and below its capacities. Below each supply and demand node, the respective supply or demand is indicated. Only intermediate nodes are labelled with their capacities below. (This is done to avoid labelling supply and demand nodes with capacities and intermediate nodes with supply variables, since such data would give no additional information.) From now on, any figures of directed distribution networks will use this labelling style, unless explicitly stated.

3.2 Path Counting

In this section, we give an algorithm for counting all the paths in a network, as well as for counting the number of paths each node contributes to the total. To facilitate path counting, we introduce the concept of "tiers of nodes". Since in our networks, the flow of commodities is directed from source nodes to customers, we can construct tiers of nodes, with the following properties:

1. We define τ to be the set of all tiers, and T_i to be the tiers in τ. Note that the sets T_i are subsets of N.
2. If n_i is a supply node then $n_i \in T_0$.
3. For arbitrary tiers a and b and $\forall n_i \in T_a$ and $\forall n_j \in T_b$ if $a \leq b$ then $\nexists e_{ji} \in E$.
4. For arbitrary tiers a and b and $\forall n_i \in T_a$ and $\forall n_j \in T_b$ if $\exists e_{ij} \in E$ then $a < b$.
5. If $n_i \in T_a$ and $n_j \in T_b$ are customer nodes, then $a = b$.

Before we present Algorithm 1 that will sort the nodes of a network into the appropriate tiers, a few definitions and clarifications must be made.

Definition We say a node n_i feeds a node n_j if $\exists e_{ij} \in E$.

We will be using the adjacency matrix M^* for our network. To obtain this, we simply take the matrix of arc capacities $M = [m_{ij}]$ and define all non-zero values to be 1. Thus if we have $M = \begin{bmatrix} 12 & 7 \\ 4 & 0 \end{bmatrix}$ our adjacency matrix would be $M^* = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$.

Algorithm 1 Enumeration of tiers in a directed distribution network

---

Given a directed distribution network $D = (N, E)$ and its adjacency matrix M^*, enumerate the subsets $T \in \tau$.

1. $\forall n_i \in N$ initialize its tier variable $t_i = 1$. Start with the current tier equal to 1.
2. If there are no nodes in the current tier, end the algorithm.
3. For each node in the current tier, visit all the nodes that feed it and set their tier to the current tier $+1$.
4. Increment the current tier by 1 and go to step 2.

Post processing: For each node n_i, add its index i to the tier set T_j such that $j =$ current tier $-1 - t_i$, unless it has no nodes that feed it and it is set in a lower tier, then set it to 0 (this is to avoid the case where a supply node could skip a tier and be classed lower).
Output: $\tau = T_0 ... T_k$

Notice that this assigns nodes to the highest tier possible. There are situations where you may have a node that could be in a lower tier, but the algorithm will assign it the highest tier it can. As an example, Figure 3.2 is a simple network with two supply nodes and two demand nodes. Node 2 could be in either tier 1 or 2, but Algorithm 1 will default to assigning it to tier 2.

Having enumerated the subsets τ, we can now set up our path counting algorithm.

The process by which Algorithm 1 arranges the nodes in tiers ensures that the set T_k contains every node which does not feed others. Therefore, this algorithm looks at all the parent nodes for a particular node, and sums their path contributions. This effectively counts all paths from the origin to that current node. Thus, with the guarantee that every

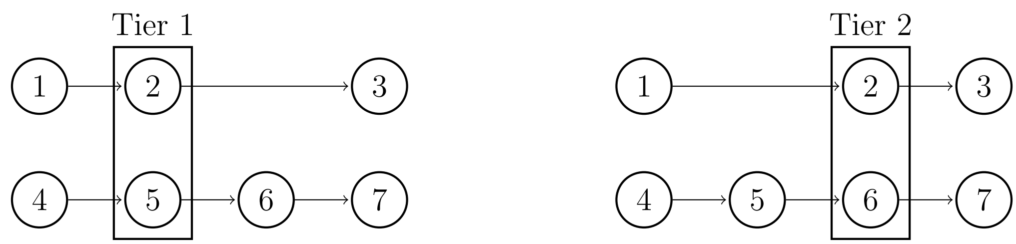

Figure 3.2: Showing how Algorithm 1 chooses tiers when multiple values are possible

Algorithm 2 Counting the number of paths in a directed network

Given a directed distribution network D = (N, E) and its adjacency matrix M^*, as well as subsets $\tau = T_0 ... T_k$, count the number of paths in D.

1. $\forall i \in T_0$, set the path variable $p_i = 1$.
2. Set current tier $ct = 1$.
3. $\forall i \in T_{ct}$, let $p_i = \sum_j p_j$ where nodes n_j feed node n_i.
4. Increment $ct + 1$.
5. If $ct = k$, let $P = \sum_{i \in T_k} p_i$ and end the algorithm. Otherwise, return to step 3.

Output P = total number of paths in the network.

customer node is in set T_k, the algorithm counts all paths in the network.

Since we want to count the total number of paths when we remove a node from the network, we can modify algorithm 2 to also count the paths in all sub-networks D_i where $D_i = (N \setminus n_i, E)$.

With these two algorithms, we are able to count the paths in any given directed distribution network, along with each of its sub-networks in which an intermediate node is removed. To clearly illustrate how the algorithms work, we present a simple network on which we use the algorithms to determine the tiers of nodes, as well as the number of

Algorithm 3 Counting the number of paths in all sub-networks of a given network

Using Algorithm 2 and appending the following steps:

6. $\forall i \in [T_1 ... T_{k-1}]$
   (a) Let T_j be the tier containing n_i and assign the current tier $ct = j + 1$
   (b) Assign $p_j^i = p_j \ \forall j$ such that if $j \in T_a$ then $a < ct$. p_j^i is the path variable for n_j when n_i is removed from the network.
   (c) $\forall l \in T_{ct}$, let $p_l^i = \sum_j p_j^i$ where nodes n_j feed node n_l.
   (d) Increment $ct + 1$.
   (e) If $ct = k$, let $P_i = \sum_{l \in T_k} p_l^i$, otherwise return to (c).

Output P, $[P_i, ...] \ \forall i \in [T_1 ... T_{k-1}]$

paths. Figure 3.3 is the layout of our network. The network has the following adjacency matrix:

$$M^* = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

We now begin Algorithm 1 to identify the tiers in the network. Figure 3.4 shows how the algorithm labels each node with a tier value, then searches through the nodes to find parent nodes, and increments the tier value of these nodes. In this figure, the label above each node is its current tier value.

With the tiers enumerated, we can now count the paths. In Figure 3.5, the labels above the nodes are the current path variables.

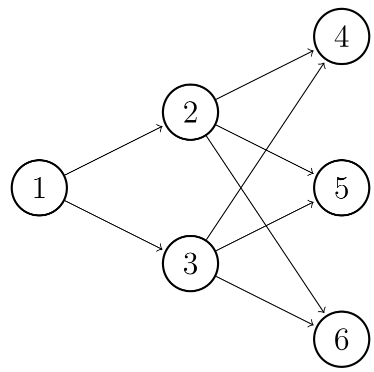

Figure 3.3: A five node network

As we can see from Figure 3.5, the final output of Algorithm 3 is $P = 6, P_2 = 3, P_3 = 3$.

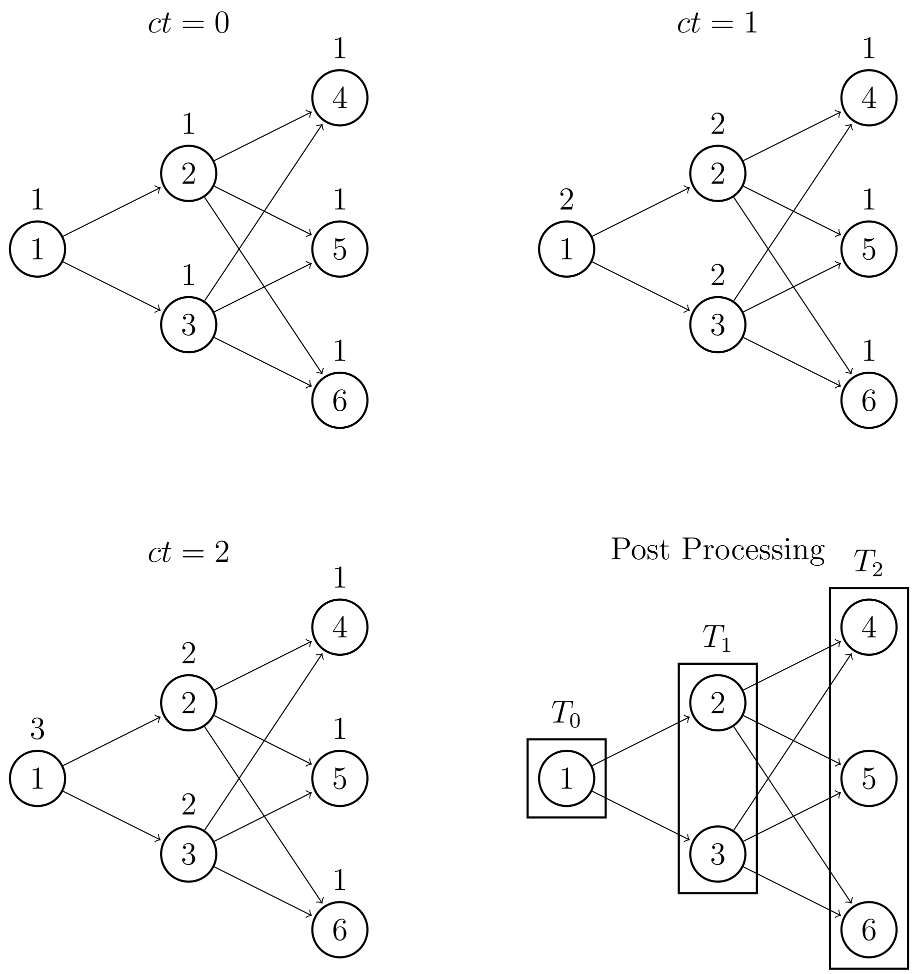

Figure 3.4: The tier enumeration algorithm

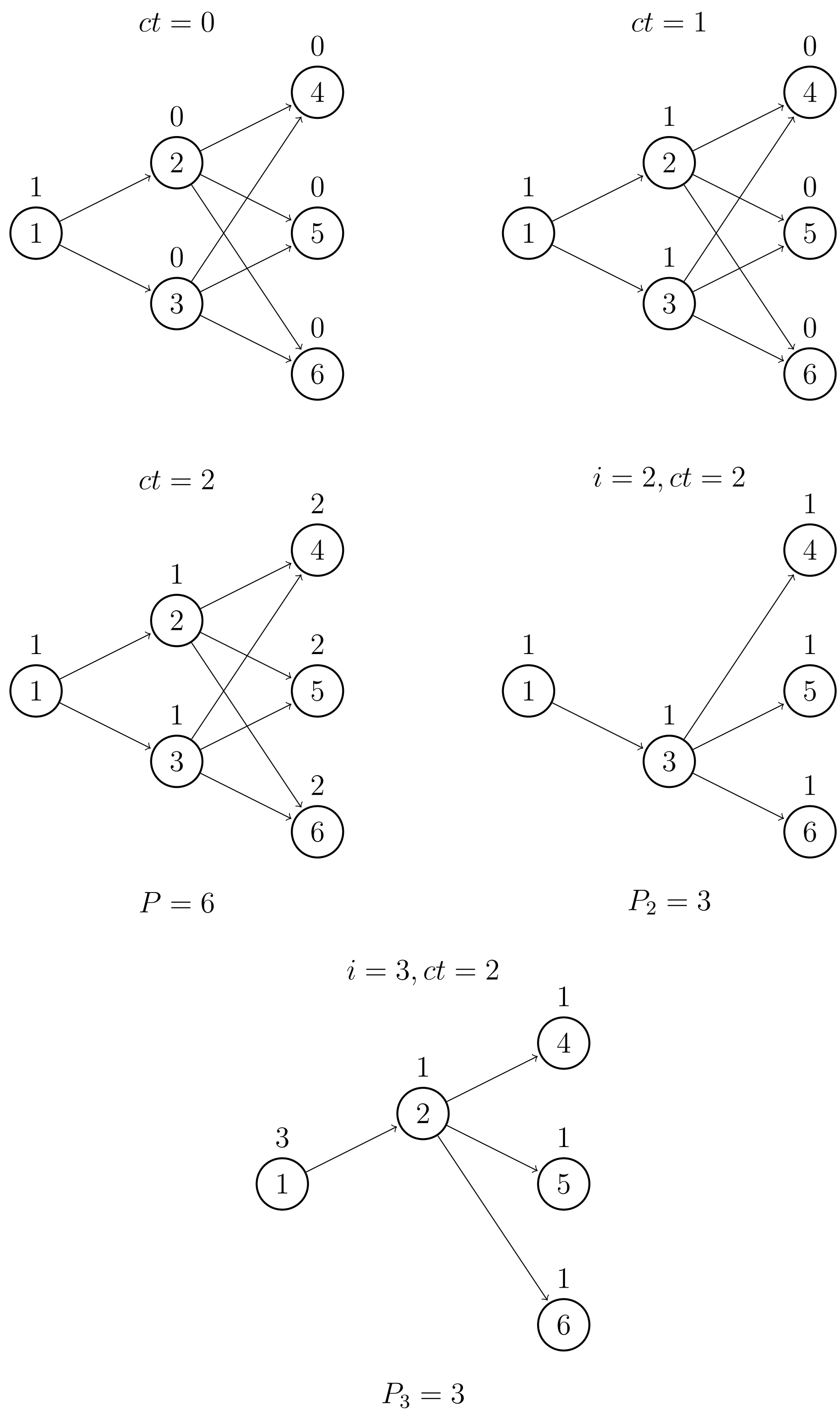

Figure 3.5: The path counting algorithm. Note, for example, that when $i = 2$ (node 2 is removed), the number of paths to node 4 changes from 2 to 1.

3.3 A Modification of the Revised Simplex Method for Optimizing Sub-Networks

We have stated that cost on its own is not an accurate representation of the importance of a node to the network, but without cost, any importance measure is inadequate. Here, we refer to the cost of a node as being the change in the objective value when we optimize the base network and the sub-network with that node removed. Specifically, we define the cost to the network of removing node i to be:

$$nc_i = z_i^* - z^* - f_i \tag{3.7}$$

Here, f_i is the cost of keeping node i open, z^* is the optimal solution when solving the base network $D = (N, E)$ and z_i^* is the optimal solution when solving the network $D_i = (N \setminus n_i, E)$. Clearly, to calculate all the nc_i's we need the optimal solutions z_i^* for each sub-network of D. We could simply run our preferred optimization algorithm on each of the networks D_i in succession, but it is easy to imagine cases where doing so would be very inefficient; for example, the distribution network of WalMart Stores, Inc. includes over 8000 stores and 200 intermediate facilities. Even though we benefit from not requiring an integer programming formulation, the time involved in calculating the optimal solutions of many networks is very large. In this section, we give a brief outline of some modifications made to the revised simplex method. Then we present some observations on which our modifications are based, and finally we give the complete algorithm.

3.3.1 Outline of Modifications

Throughout this section, we are dealing with arbitrary linear programs, rather than our specific directed distribution network model. As such, it is important that the notation be clearly laid out, in order to avoid confusion. Given an arbitrary linear program
LP2:

$$\begin{aligned} &\text{maximize } c^T x \\ &\text{subject to } Ax = b \\ &\qquad\qquad\quad x \geq 0 \end{aligned}$$

We denote B, a basis of A, in such a way that B contains the indices of the columns which form a basis A (We assume that A has full rank). Likewise, we denote N to be the non-basic columns of A. When we write A_B, we refer to the square matrix composed of the basic columns of A. Similarly A_N is the matrix of non-basic columns of A. We denote $\tilde{x}$ to be a feasible solution to LP2. Typically, each row of the matrix A represents an original constraint of the program including a slack variable, added when necessary, so that we have a program in standard equality form. Before outlining the modifications we make to the revised simplex method, we present a standard statement of the method:

Statement of the Revised Simplex Method

1. Start with a feasible basis B and a corresponding basic feasible solution x^*.

2. Solve $A_B^T y = c_b$ for y.

3. Find $k \in N$ such that $\bar{c_k} = c_k - A_k^T y > 0$. (If none exists, stop; x^* is optimal.)

4. Solve $A_B d = A_k$ for d.

5. If $d \leq 0$, stop; the problem is unbounded.

6. Compute $t = \min\limits_{\substack{i \in B \\ d_i > 0}} \dfrac{x_i^*}{d_i}$ and choose $r \in B$ such that $d_r > 0$ and $\dfrac{x_r^*}{d_r} = t$.

7. Replace x_i^* by $x_i^* - td_i$ for all $i \in B$ and replace x_k^* by t.

8. Replace B by $(B \cup \{k\}) \setminus \{r\}$ and go to step 2.

The main idea behind our modifications follows. We wish to identify that iteration during which a constraint first becomes involved in the basis and basic feasible solution, keep the information about that iteration's basis and basic feasible solution on hand, and come back to it later. When we say that "a constraint becomes involved in the basis", we mean that one of the variables which has a non-zero coefficient in the constraint enters the basis. With the information about the basis and basic feasible solution, we can restart the revised simplex method, beginning from a more advanced initial point, all the while setting the right hand side of the desired constraint to 0. It is likely, however, that this will impact the sequence of pivots in the simplex method. However, with the results of Section 3.3.2, it will become clear that we can effectively modify the RHS of particular constraints that are not yet active in the basis.

The second main idea is to run the revised simplex method once through on the base LP. We will keep track of when specific constraints become active, and the state of the algorithm at those points, and then restart the method from an advanced starting point for each constraint we wish altered. Let us begin by presenting the observations necessary for this method to work.

3.3.2 Observations on the Revised Simplex Method

The first observation is related to the eligibility of slack variables to leave the basis.

Property 1. *A slack variable s_j cannot leave the basis unless at least one of the variables with a non-zero coefficient in constraint j is in the basis, or one such variable is entering the basis in the current iteration of the revised simplex method.*

Proof. Set s_j to be the slack variable for the j'th row of matrix A, the index of s_j to be the u'th element of B, and I to be the set of indices for which the coefficient in the j'th row of A is non-zero.

Assume that $B \cap I = \emptyset$, and $k \notin I$. Then when solving $A_B d = A_k$ for d in step 4 of the revised simplex method, we have that $A_{k,j} = 0$ and the j'th row of A_B is all zero except for one entry in the u'th column which is $[A_B]_{u,j} = 1$. Thus, $d_j = 0$.

Now in step 6 of the revised simplex method, we choose the leaving variable by finding the minimum over all basic variables i for which the associated $d_i > 0$, since d_j is not positive, s_j is not the entering variable.

Notice that there are only four cases for k and I:

1. Either $k \in I$ and $I \cap B = \emptyset$
2. Or $k \in I$ and $I \cap B \neq \emptyset$
3. Or $k \notin I$ and $I \cap B = \emptyset$
4. Or $k \notin I$ and $I \cap B \neq \emptyset$.

In Cases 1, 2 and 4, at least one of the conditions necessary for the slack variable to leave the basis is true. We have also shown that in Case 3, s_j would not be the leaving variable, thus we have shown the result. $\square$

The next result uses the same argument in its proof, and relates to the value of slack variables in the basic feasible solution.

Property 2. *For a slack variable s_j, its associated objective value x_j^* will be equal to the RHS of the constraint j at least until one of the variables with non-zero coefficient in constraint j enters the basis for the first time.*

Proof. In the proof of Property 1, we showed that there are four cases for k and I. Notice that in a given iteration of the revised simplex method, cases two and four can only occur when case 1 applied to a previous iteration. Therefore, since we are only concerned about whether or not x_j^* has changed before the first time case one occurs, we can simply show that x_j^* will not change during an iteration where case three is true.

Now, also from the proof of Property 1, we have shown that $d_j = 0$ when case three is true. Recall that in Step 7 of the revised simplex method, x_i^* is replaced by $x_i^* - td_i$. We know that $td_j = 0$, therefore x_j^* will remain unchanged during any iteration where case 3 is true. This is all that is necessary to prove the result. □

Taken together, Properties 1 and 2 define the earliest point at which a slack variable is eligible to leave the basis, and guarantee that the objective value of a slack variable will not change until that point. The next property shows that making any change to the RHS of a slack variable's constraint will not have an effect on the revised simplex method before the point at which that variable is eligible to enter the basis.

Property 3. *The RHS of a constraint can be changed, without having an effect on the sequence of pivots in the revised simplex method, until any variable with a non-zero coefficient in that constraint enters the basis.*

Proof. In order for a change in the RHS of a constraint to affect the pivot order of the revised simplex method, either a different entering variable must be selected in an iteration, or a different leaving variable must be selected. Let j be the constraint whose RHS is being changed. For the sake of argument, assume that the first two variables to enter the basis have zero coefficients in the j row; if this is not the case, the proof is trivial.

Let x_{E1} be the variable that would enter during the first iteration if the RHS is not changed. Now, entering-variable selection is done during step three of the revised simplex method. Note that $\bar{c}_k = c_k - A_k^T y > 0$ is not dependent on the vector x^* or the RHS b. Thus, x_{E1} would still be selected as an entering variable.

Let x_{L1} be the variable that would leave during the first iteration if the RHS is not changed. Leaving-variable selection is done during step six of the revised simplex method. Observe that since the first entering variable was the same, the vector d found in step four will remain the same. Thus, regardless of the change to x_j^*, the slack variable s_j will not leave the basis since we know from Proposition 1 that it cannot. Then, we have that x_{L1} is still the leaving variable, since all other entries of x^* are the same, and the ratio $\dfrac{x_{L1}^*}{d_{L1}}$ will be the minimum such ratio.

Next, assume that in the first $(g-1)$ iterations, the entering and leaving variables are unchanged when the RHS of constraint j is changed, that none of the basic variables have non-zero coefficients in the j^{th} constraint, and that the next entering variable is not one of these variables either. Let x_{Eg} be the variable that enters in iteration g if the RHS is unchanged. Then solving for y in the second step of the revised simplex method, we obtain the same value regardless of the change to b_j, since the basis B is unchanged. Thus, in Step 3, we select the same entering variable x_{Eg}. Let x_{Lg} be the variable that leaves the basis in iteration g when the RHS is not changed. Then, solving for d in Step 4 will result in the same value, regardless of the change to the RHS.

Thus, the only way in which x_{Lg} will not be selected as the leaving variable is if the ratio $\frac{x^*_{Lg}}{d_{Lg}}$ is no longer the minimum value of all such ratios. To show that this is untrue, we must show that the vector x^* does not change if the RHS is modified. We know that s_j will not enter the basis, since it is the slack variable for constraint j. We also know that the vector x^* is updated in Step 7 of the revised simplex method by subtracting $x_i - td_i$. From Property 2, x^*_j will not change from its original value. Recall that at each previous iteration, the d and t values found were identical, regardless of the change to the RHS. This is because d depends only on the entering variable and t on the leaving variable of that specific iteration. Thus, x^* will remain unchanged despite the change to the RHS. This implies that the same leaving variable will be selected in iteration g, regardless of the change to the right hand side of constraint j.

Consequently, by induction, we have shown the result. $\square$

This last property is the key to our modification, because it implies that if we change one of the b_j values in the RHS before a certain point, we can be assured that the basis B and the basic feasible solution x^* are valid for the modified linear program (Note that the x^*_j is changed because of the change to b_j, but this is the only change in x^*).

3.3.3 Modification of the Revised Simplex Method Algorithm

Having shown the properties in the previous section, we can now present our modifications to the revised simplex method. Those modifications do not affect the normal steps of the method, thus we present them as sub-steps.

1a For each constraint j that we wish to modify, create an index set I_j such that $i \in I_j$ if

$A_{i,j} \neq 0$. Indices of slack variables are not permitted for such values of i.

3a Once the index of the entering variable k is selected, verify for each j whether $k \in I_j$. If it is, then save the basis B and current feasible solution x^* as the pair $S_j = \{B, x^*\}$. Additionally, set $I_j = \emptyset$.

3b Once the optimal solution has been found for the unmodified linear program, use the regular revised simplex method to solve, for each j, the modified linear program where $b_j = b'_j$, using the basis B from S_j and the basic feasible vector x^* with $x^*_j = b'_j$ as initialization data.

Notice that in step 3b, we are assuming that the modified LP is feasible. This is not guaranteed to be the case, thus we need to solve an auxiliary problem via the two-phase method in order to obtain a feasibility certificate. We do not mention this step in our method, since we assume that this is a first step of any optimization algorithm. In that case, we have the guarantee that the information in S_j is a valid basis and basic feasible solution, contingent on the fact that the LP is feasible when b_j is modified. The output of this modified method is an optimal value and optimal vector pair $\{z, x^*\}_j$ for the base case, along with every modified constraint j.

Thus, if we want to calculate the network cost of a node i, which we defined as nc_i, we need to remove node i from our program. In our model LP1, we have a node capacity, which we call w_i. This parameter is involved as the right hand side of constraint 3.4. We now use a trick to remove the node from the network; by letting $w_i = 0$, we guarantee that no flow will be assigned to node i. This change has the same effect as removing the node. Thus, solving the LP when $w_i = 0$ is equivalent to solving the LP for the network $D_i = (N \setminus n_i, E)$. Therefore, to calculate network cost, we simply use our modified simplex method, and choose the constraints which model the capacity of intermediate nodes as those we wish to modify. The output of this method will be all the pairs $\{z, x^*\}_i$ we need. We can thus calculate $nc_i = z^* - z_i^*$ for each node i and we have our network cost.

3.4 Node Importance Algorithm

In the previous sections, we have described algorithms that permit us to calculate each of our importance measures. This section will lay out how all of those algorithms will fit together to form our node importance-algorithm. Remember that we are considering three measures: the number of paths contributed by a node, the network cost of this node, as well as the number of re-routings suffered when removing it. We have not given an explicit algorithm for the number of re-routings, but our modified revised simplex method gives us all the information we need to calculate it. Let us define the number of re-routings for a given node i as:

$$r_i = \sum_k |x_k^* - \{x_k^*\}_i| \tag{3.8}$$

Let us also define the triplet of importance measures $IM_i = \{P_i, nc_i, r_i\}$ and for ease of use, when the index $i = 0$ we define $IM_0 = \{P, z^*, |x^*|\}$. We now give the node importance algorithm.

In order to properly convert the output IM into a ranking of nodes, we calculate the change in values for each of our measures, and represent those in percentages. The number of reroutings, when taken compared to the amount of flow through the base network is represented as a percentage, even though it is not strictly a measure of the change

Algorithm 4 Node Importance Algorithm

Given a directed distribution network $D = (N, E)$.

1. Compute the adjacency matrix M^* of D.
2. Enumerate the tiers τ of network D by using Algorithm 1.
3. Run Algorithm 3, store the output in IM.
4. Run the Modified Revised Simplex Method on network D and consider the changes: $b_i = 0 \forall i \in [T_1 ... T_{k-1}]$ where k is the highest tier. If a change b_i results in an infeasible network, let the particular $\{z^*, x^*\} = \{-1, -1\}$.
5. Compute nc_i and $r_i \forall i \in [T_1 ... T_{k-1}]$, store these values in IM. If the values $\{z^*, x^*\} = \{-1, -1\}$ for a particular i, let $nc_i = -1$ and $r_i = -1$.
6. Assign $IM_0 = \{P, z^*, x^*\}$.
7. Output IM.

in total flow. Once we have the percentages for each node, we can rank the nodes by applying a weight to each measure's percentage and summing them together to obtain an "importance". The nodes are then ranked from highest to lowest importance. We do not compare nodes in different tiers, since these are intrinsically different.

Since we have determined that cost is the most important measure, we will weigh the path and rerouting percentages at one tenth of the weight of cost. This ensures that economical nodes are more important, and also that we are breaking ties between similarly costly nodes by using our importance measures. Chapter 4 gives examples of how the algorithms presented in the current chapter work, as well as the sensitivity of the measures to parameter changes.

Chapter 4

Application and Practical Examples

This chapter will present a simple example, showing how the node-importance algorithm works on networks, in order to clarify what kind of information is gained from the algorithm's output. We then present a few cases where the results are either not obvious, or some judgement is needed. We conclude with a large example that approximates a real distribution network.

4.1 Simple Applications

In order to clearly demonstrate the output of our node-importance algorithm, we construct a small network and apply our method. We discuss the steps taken and give interpretations for the resulting data. For our purpose, we create a small network based on some cities in eastern Canada. We select ten cities, and assign demands based on the population of those cities, arc costs based on the driving distance between cities, and intermediate node capacities are picked randomly, in such a way as to satisfy demand even if one of the nodes is removed. Note that for this simple example, we omit using a facility cost, and as such the network cost of a node is simply the difference in optimal solutions. Figure 4.1 gives the layout of the network, with relevant demands, node capacities and arc costs labelled below the respective arcs and nodes.

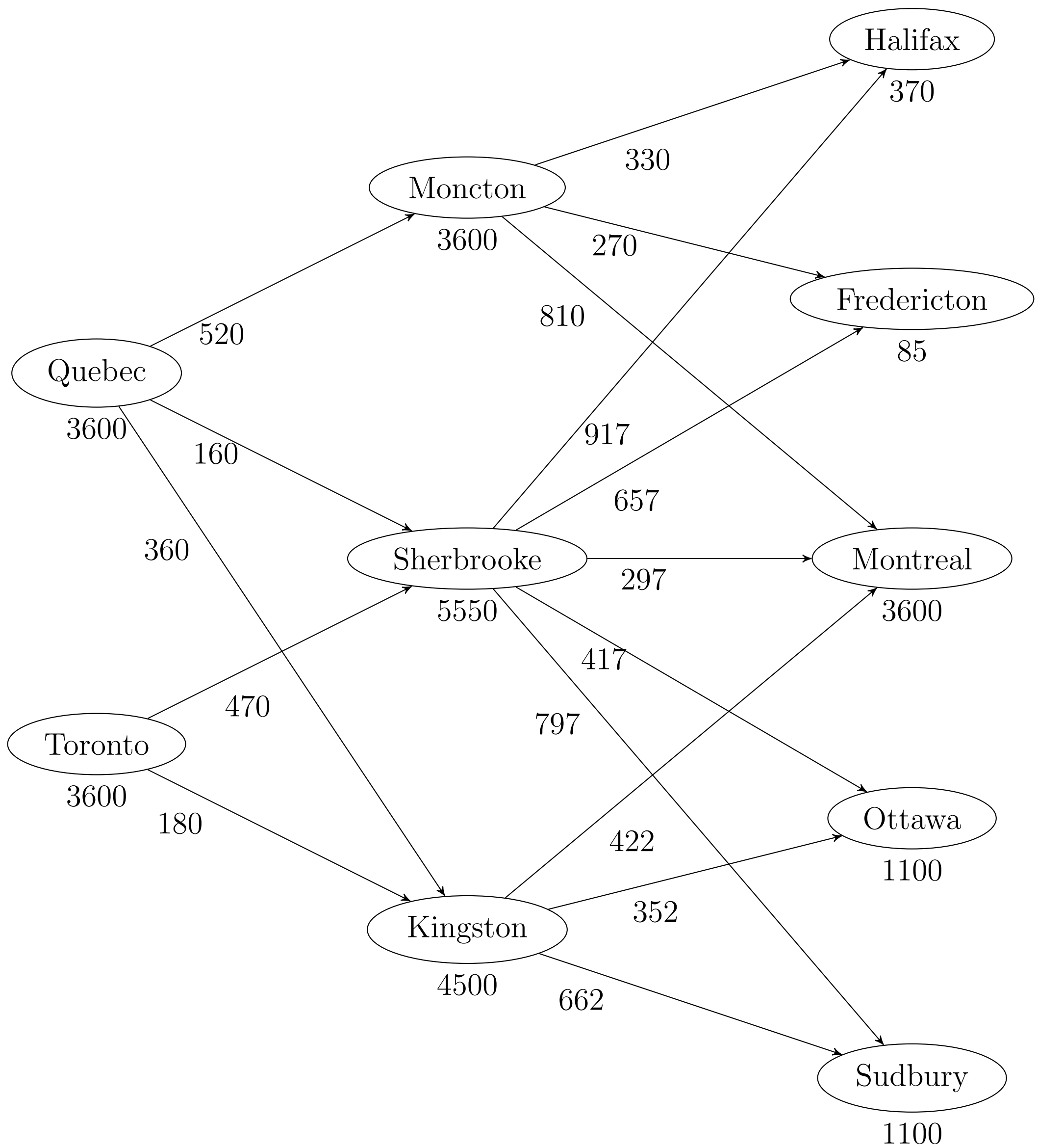

Figure 4.1: Layout of the simple Eastern Canada example

Node Removed	x^*	nc_i		Paths		Reroutings	
Base Case	2 811 061	0		19		10 626	
Sherbrooke	3 689 710	878 649	31%	9	52.6%	12 580	118%
Moncton	2 897 346	86 285	3%	16	15.7%	1 820	17%
Kingston	3 343 786	532 725	19%	13	31.5%	6 852	65%

Table 4.1: The output IM when Algorithm 4 is used on the simple East Canada network

In Table 4.1 the output IM is presented, with the relevant entries shown as numerical values, as well as percentage changes from the base case. The Base Case row gives the optimal solution, the number of paths as well as the amount of flow along all arcs for the base case. Before any weights are associated to the values, a ranking is obvious, since the Sherbrooke node has the largest changes for all three measures, and Moncton has the smallest changes for each measure. We thus get a ranking of:

1. Sherbrooke
2. Kingston
3. Moncton

The Eastern Canada example does not lend itself to further analysis, due to the simplicity of the network. Adjusting the parameters for any of the nodes or arcs does not alter the ranking unless an unrealistic, or infeasible, change is made. An unrealistic change would be to vary the cost of arcs leaving a facility to values which are orders of magnitude higher than their original values. An infeasible change is made when the network demands can no longer be met, after the removal of a node. In the Eastern Canada example, if we drop the capacity of the Kingston node by a very large amount, the linear program becomes infeasible when the Moncton node is removed. We would thus have to assign the status “Critical” to the Moncton node. This would consequently make it the most important node in the network, along with any other nodes with the “Critical” status.

What this example gives us, then, is a simple and straightforward application of the method, and the resulting data. For in-depth analysis, we require a larger, more realistic network.

4.2 Analysis of a Realistic Problem

In this section, we analyse a full size, realistic problem. We first construct the problem, then use the method on the basic network. Next we vary the parameters of the problem in order to analyse the impact such changes have on the rankings of nodes.

4.2.1 Generating a Large Realistic Problem

We generate a large problem by approximating the needs of an arbitrary retail company operating in the United States. We assume that our company sources some of its products from overseas and a portion locally. As such, we set three supply nodes to represent two marine ports, and one factory in the mainland United States. We assume that the cost of sourcing parts from each supply nodes is equal, for the sake of convenience. We select New Orleans and Newark as representative ports, based on the ranking of ports with highest international trade by volume. Kansas City is selected as the mainland factory, mainly for its central location. We next choose to separate our distribution network into regional distribution centres and local distribution centres. The selection of four regional distribution centres and eleven local distribution centres is somewhat arbitrary, spreading the centres to cover the geographic layout of the United States. We finally select customer nodes by picking the sixty most populous cities in the mainland United states, utilizing the population of each as a guide for the respective demands of the sixty nodes.

Parameters for the distribution centres are selected by referring to The Boyd Company Inc. (2010), a study of distribution warehousing costs. Boyd suggests the value of \$1.93 per mile, which we round to \$2 per mile, for transporting one 30 000 lb. truckload using a private carrier. We use this truckload value as our unit for supply and demand. Boyd also gives the costs of operating distribution centres, which we adjust for our needs by scaling them to the capacity we assign each of our warehouses. To determine the arc costs, we use the common road distance between the two nodes in question, collected from Google maps on February 27th, 2011. Quantities at source nodes, as well as the capacities of distribution centres, are selected semi-randomly in such a way that capacities reflect local demand fairly closely as well as a moderate level of surplus demand. We choose the set of arcs in light of the geographic closeness of customers and distribution centres, adding some long arcs in cases where customers are far from multiple centres. The complete data set for this problem is given in Appendix A. Figure 4.2 gives a map of the network, with arcs excluded. Each type of node is represented by a different icon.

The network produced for this example is not optimized; it serves only as an example. We do not consider several possible refinements, such as would be obtained from location models. Nor do we try to eliminate redundant shipments, since these are outside the scope of this example. Our assumption is that in a real-world case, the network would have already been optimized before attempting to rank the nodes with our method. The aim of this particular network is to demonstrate and analyse the output of the ranking method. The fact that our network is not optimized is irrelevant to this goal, since we give a ranking

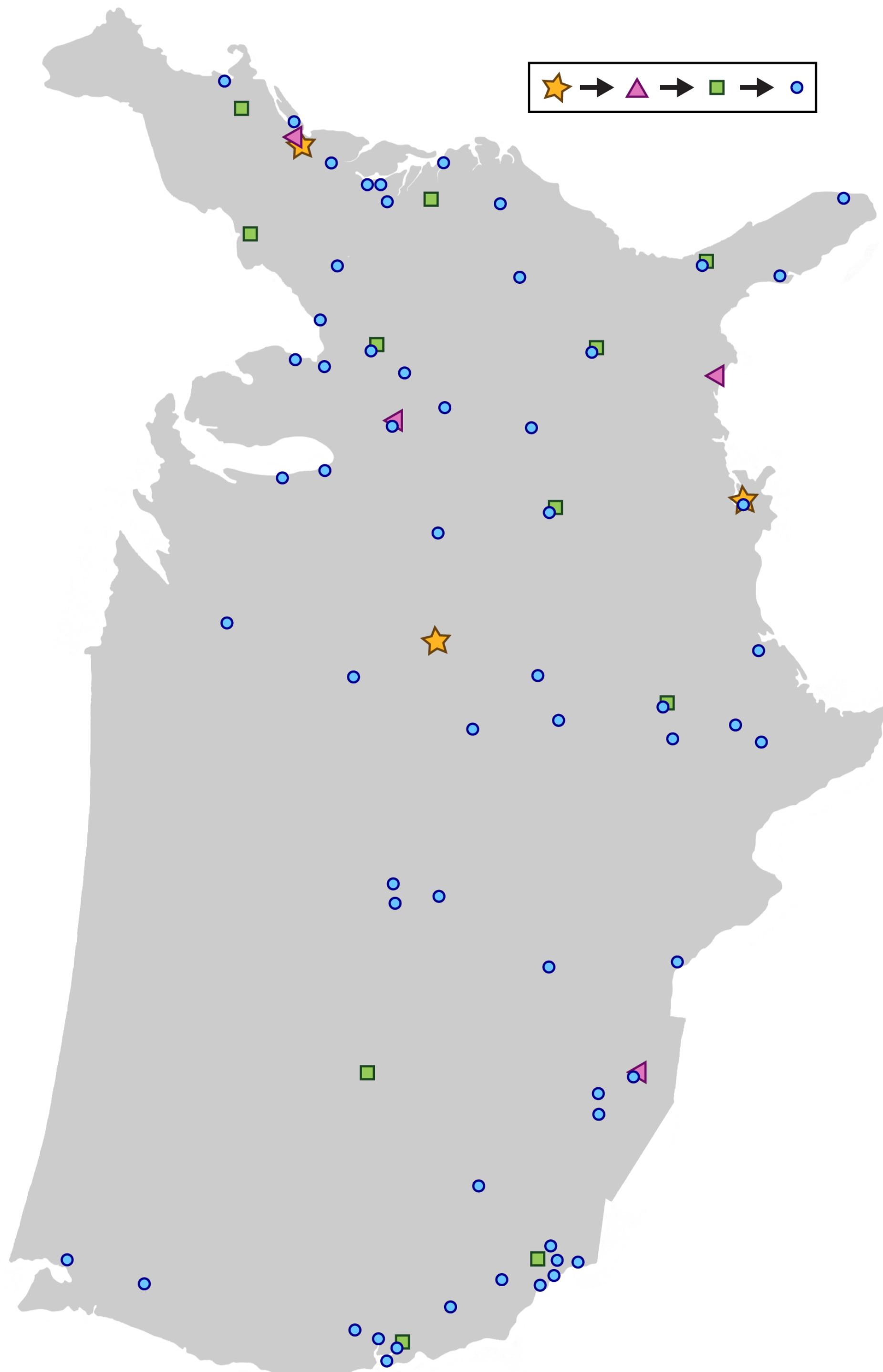

Figure 4.2: Map of the Large Realistic Example

of relative importances for our nodes. We are able to answer the question: Is node 1 more important than node 2? The fact that nodes 1 and 2 are a part of an unrefined network does not affect that question. This is to say, our method gives a ranking of importance, regardless of the state of the network.

4.2.2 Base Analysis

Table 4.2 gives the IM output of Algorithm 4 for the large network.

Node Removed	x^*	nc_i		Paths		Reroutings	
Base Case	5 803 801	0		1 176		15 438	
Indianapolis	6 337 184	414 546	7%	261	22%	2 950	19%
Pensecola	7 012 197	1 084 152	19%	360	31%	8 030	52%
Tucson	7 062 503	1 093 649	19%	294	25%	9 728	63%
Meadowlands	7 194 073	1 192 337	21%	261	22%	8 112	53%
Springfield	6 143 591	231 684	4%	60	5%	4 242	27%
Rochester	5 821 808	-76 951	-1%	72	6%	530	3%
Richmond	5 827 932	-2 163	0%	108	9%	1 148	7%
Atlanta	5 879 255	38 111	1%	132	11%	1 532	10%
Jacksonville	5 846 518	22 077	0%	90	8%	632	4%
Columbus	Infeasible						
Memphis	6 076 690	228 613	4%	84	7%	3 260	21%
Dallas	6 285 127	431 742	7%	96	8%	4 620	30%
San Bernardino	6 413 704	546 623	9%	96	8%	3 314	21%
San Jose	5 998 852	130 329	2%	102	9%	2 484	16%
Salt Lake City	5 814 316	-34 669	-1%	120	10%	590	4%

Table 4.2: The output IM when Algorithm 4 is used on the large example

This produces the following ranking:

Critical	
	Columbus
Regional Distribution Centres	
1	Meadowlands
2	Tucson
3	Pensacola
4	Indianapolis
Local Distribution Centres	
1	San Bernardino
2	Dallas
3	Springfield
4	Memphis
5	San Jose
6	Atlanta
7	Richmond
8	Jacksonville
9	Salt Lake City
10	Rochester

The rankings are separated into three categories, one for each tier of intermediate nodes as well as the critical category. In this base case, there is only one node in the critical category. However, were there more, all of the critical nodes together would be considered of equal importance. Regional and local distribution centres are seperated due to their differences in roles. We are ranking nodes relative to each other, therefore only those nodes that are "similar" should be ranked in the same way. A detailed analysis of a few cases is given for each of the three categories.

Critical Nodes

Memphis is the only critical node in our program. Inspecting the layout of the network as well as its incidence matrix, we notice that Columbus is centrally located and contributes the largest number of paths amongst the local distribution centres. Observe also that a few customers served by Columbus, namely Minneapolis and Chicago, do not have many incident arcs. Together, these factors contribute to the infeasibility of the network, once Columbus is removed. Excess capacity of nearby nodes is not large enough to meet demands of all client nodes served by Columbus, and thus that node is deemed critical.

Regional Distribution Centres

The data suggest a very close ranking os the top three regional distribution centres, with Indianapolis a very distant fourth. Inspecting the parameters, we can infer that Indianapolis is not important to the network. The total flow through Indianapolis, 600 units, is all then routed directly to the Columbus node. Interestingly, Columbus does not supply some of its closest neighbours, such as Cleveland and Cincinnati; Columbus rather supplies nodes, such as Milwauke and Chicago, which are furthest from other local distribution centres. This suggests that sending goods through Indianapolis is more expensive than other alternatives. Therefore, Indianapolis is not an effective supplier for most nodes, and thus handles very little flow. This contributes to the low network cost r_i as well as the small number of reroutings when compared to other regional distribution centres. Combine those points with the fact that the Indianapolis node contributes fewer paths to the network than other regional distribution centres, and we can say that this node is less important than each of Pensecola, Tucson, and Meadowlands.

Local Distribution Centres

At the local distribution centre level, the ranking of the first six nodes is evident without analysis. The decision between Jacksonville and Richmond is one that is not as obvious, and bears some thought. In terms of network cost both nodes are very close. However the path and rerouting values are quite different, placing Richmond ahead of Jacksonville in the ranking. Upon closer inspection, we observe that the number of reroutings for Jacksonville is equal to four times the total flow going through this node. That is the smallest amount of reroutings that can occur when a node is deleted: we count the difference in flow along arcs, so redirecting one unit of flow to pass through a different local distribution centre involves four changes.

The same cannot be said of Richmond, where the number of reroutings is greater than four times the total flow. This larger value is due to the fact that Newark, the supplier from which Richmond's flow originates, is at capacity. When Richmond is removed, its former customers are now being supplied by flows which do not originate in Newark. That means Newark now has available capacity. This excess is used to supply Detroit with less costly flows. In essence, when removing Richmond, the nearby nodes are not adequately able to handle the new demands from customers. This means there is a lack of excess capacity supporting Richmond. Richmond is thus more important than other nodes, e.g. Jacksonville, with similar network costs.

4.2.3 Impact of Parameter Changes on Rankings

This section contains the analysis of several versions of the base problem where parameters have been changed, in order to understand the impact each parameter has on the importance of a node.

Node Cost

Recall that we defined the network cost of a node i as $nc_i = z_i^* - z^* - v_i$. If we vary v_i for a node, the results of optimization are the same, since v_i is not included in our linear programming formulation. Thus, the only effect of changing the v_i for a node will be that its network cost will also change by that same value in the opposite direction. If we take the Jacksonville node and decrease its v_i by 40 000. Table 4.3 gives the new Jacksonville row for Table 4.2.

Node Removed	x^*	nc_i		Paths		Reroutings	
Jacksonville	5 846 518	62 077	1%	90	8%	632	4%

Table 4.3: The new Jacksonville row for Table 4.2 when the cost of Jacksonville is increased by 40 000.

The only effect this would have on the rankings is to move Jacksonville ahead of Richmond. Even though Jacksonville's network cost is now higher than Atlanta's, Atlanta still handles much more flow and contributes a much larger number of paths to the network. Jacksonville is thus less important than Atlanta.

Varying the cost, f_i, of a node can have a direct impact on the ranking of that node. The impact of such a change is, however, restricted to the node in question. Any change in ranking will occur due to node i moving up or down in the rankings. The relative ranking of all other nodes in reference to each other will be maintained (i.e. if the cost of node A is changed, the order of nodes B and C will remain the same in relation to each other).

Node Capacity

We identify several different types of changes to node capacity and observe each in turn, to understand the impact these have on the rankings of nodes.

We first consider a node whose capacity constraint is not currently binding. An increase in that capacity does not have any effect on the importance measures of the node in question. However it can have an impact on the importance measures of neighbouring nodes. Whenever we increase capacity for one node, we essentially increase the amount of excess capacity that other nodes have available to them. This leads to the possibility that by increasing the capacity of a "non-binding" node, its neighbouring nodes will lose importance due to the greater amount of excess capacity. Neighbouring nodes may have their importance decrease, and the other nodes in that tier may have a different relative ranking, but the importance of the altered node does not change.

To illustrate this point, we consider the case of increasing the capacity for Pensecola from 1 900 units to 2 300 units. Table 4.4 gives the resulting measures for the four regional distribution centres. Notice that the the values for Pensecola are unchanged but all the other nodes have had their measures decreased. Due to these changes, Pensecola now becomes more important than Tucson, yet still less important than Meadowlands.

The next change we consider is to lower the capacity of a non-binding node, to a level which is also non-binding. In essence, this change reduces the amount of excess capacity available to neighbouring nodes, which may lead to their importance increasing. The fact

Node Removed	x^*	nc_i		Paths		Reroutings	
Base Case	5 803 801	0		1 176		15 438	
Indianapolis	6 266 342	343 704	6%	261	22%	3 476	22%
Pensecola	7 012 197	1 084 152	19%	360	31%	8 030	52%
Tucson	6 886 863	918 009	16%	294	25%	8 636	56%
Meadowlands	7 194 073	1 192 337	21%	261	22%	7 896	51%

Table 4.4: The new importance measures for regional distribution centres, when Pensecola's capacity is increased to 2300 units.

that the new capacity constraint is non-binding means that the importance measures for the node in question do not change. The net effect of lowering the capacity of a non-binding node to a similarly non-binding level is that neighbouring nodes may become more important relative to the altered node, and that the relative ranking of all other nodes in that tier may change.

As an example of this type of change, we decrease the capacity of Meadowlands to 1 900 units, which yields the data in Table 4.5. Even though this does not lead to a change in ranking, from the base, we see a clear increase in values for Tucson. This increase puts Tucson very near Meadowlands in importance.

Node Removed	x^*	nc_i		Paths		Reroutings	
Base Case	5 803 801	0		1 176		15 438	
Indianapolis	6 337 184	414 546	7%	261	22%	2 950	19%
Pensecola	7 012 197	1 084 152	19%	360	31%	8 030	52%
Tucson	7 062 503	1 093 649	19%	294	25%	8 636	63%
Meadowlands	7 194 073	1 192 337	21%	261	22%	7 896	51%

Table 4.5: The new importance measures for regional distribution centres, when Meadowlands' capacity is decreased to 1900 units.

We next consider increasing the capacity of a node which is at capacity. The direct impact to the altered node is that it may handle more flow, which would mean a greater network cost and an increase in the number of reroutings, which would increase the node's importance. Since the altered node may be handling more flow, other neighbouring nodes may have had their flows reduced. In that case, the importance measures for those nodes would decrease. Some neighbouring nodes having their importance decreased can affect the relative ranking of all other nodes in that tier. Note that if the increased capacity is completely used by the network, there is no increase in excess capacity, thus there is no increase in importance to neighbouring nodes. However, in the case where the altered capacity is no longer binding, there may be a conflicting effect due to the increase in excess capacity and a decrease in flow handled by neighbouring nodes.

In Table 4.6, we present the data for local distribution centres when the capacity of the Columbus node is increased to 900. The change does not affect the ranking, however we can see that other nodes in the network have their importance measures reduced. Note that since the new capacity is no longer binding, there are a few values which are actually increased, namely the number of reroutings for Rochester. The effect is too small to impact the ranking, but it is noteworthy.

The last modification considered is to decrease the capacity of a node where the capacity constraint is already binding. Such a change has no effect on excess capacity; the altered node just handles less flow. That node thus has a lower network cost, and with a potentially smaller number of reroutings, its importance will decrease. Neighbouring nodes which handle the diverted flow have an increased network cost and a potentially increased number of reroutings. Decreasing a binding capacity lowers the importance of the altered node. Diminishing that capacity can increase the importance of neighbouring nodes, which can alter the relative ranking of all other nodes in the tier.

To illustrate this change, the capacity of the Dallas node was lowered to 550 units. The

Node Removed	x^*	nc_i		Paths		Reroutings	
Base Case	5 791 237	0		1 176		15 438	
Springfield	6 131 0601	219 153	4%	60	5%	4 236	27%
Rochester	5 797 117	-101 642	-2%	72	6%	866	6%
Richmond	5 822 001	-8 094	0%	108	9%	1 230	8%
Atlanta	5 822 369	-18 775	0%	132	11%	1 310	8%
Jacksonville	5 833 954	9 513	0%	90	8%	632	4%
Columbus	Infeasible						
Memphis	6 039 784	195 355	3%	84	7%	2 812	18%
Dallas	6 249 865	396 480	7%	96	8%	4 262	28%
San Bernardino	6 401 140	534 059	9%	96	8%	3 314	21%
San Jose	5 986 288	117 765	2%	102	9%	2 484	16%
Salt Lake City	5 801 752	-47 233	-1%	120	10%	590	4%

Table 4.6: The new importance measures for local distribution centres, when Columbus' capacity is increased to 900 units.

results in Table 4.7 show that Memphis joins Columbus as an infeasible node, and that Jacksonville passes Richmond in the ranking. Apart from those changes, most nodes see increases to their values. The Dallas node itself sees a drop in the number of reroutings, since it now handles less flow.

Arc Costs

Here we consider altering one or multiple arc costs for a node.

Lowering the cost of incoming or outgoing arcs for a particular node may make that node more attractive to the network, which could lead to increased flow through the node. This increase leads to a higher network cost and a greater number of reroutings, which enhances the importance of that node. Neighbouring nodes may now handle a reduced flow, which lowers their network cost and number of reroutings. Additionally for neighbouring nodes, the cost of excess capacity is diminished, thus their network cost may be further

Node Removed	x^*	nc_i		Paths		Reroutings	
Base Case	5 847 594	0		1 176		15 438	
Springfield	6 187 343	275 436	5%	60	5%	4 236	27%
Rochester	5 866 243	-32 516	-1%	72	6%	926	6%
Richmond	5 868 149	38 054	1%	108	9%	908	6%
Atlanta	5 982 964	141 820	2%	132	11%	1 866	12%
Jacksonville	5 891 389	66 948	1%	90	8%	764	5%
Columbus	Infeasible						
Memphis	Infeasible						
Dallas	6 285 127	431 742	7%	96	8%	3 666	23%
San Bernardino	6 457 497	590 416	10%	96	8%	3 314	21%
San Jose	6 042 645	174 122	3%	102	9%	2 484	16%
Salt Lake City	5 858 109	9 124	0%	120	10%	590	4%

Table 4.7: The new importance measures for local distribution centres, when Columbus' capacity is increased to 900 units.

reduced. Therefore, lowering the cost of arcs for a given node can increase its importance and decrease the importance of neighbouring nodes, which can change the relative rankings of other nodes in the tier.

Raising the cost of incoming or outgoing arcs for a particular node can have the exact opposite effect (which is expected). The increased cost can lead to diminished flow through the altered node, which lowers the network cost and the number of reroutings. Neighbouring nodes may become more attractive, and thus may now handle enhanced flow. Raising the cost of arcs for a particular node can decrease its importance and increase the importance of neighbouring nodes, which may alter the relative rankings of the other nodes in the tier.

Here we decrease the cost of incoming arcs to the Indianapolis node by roughly 25%, in order to illustrate this type of change. Table 4.8 gives the data for the regional distribution centres. Although the ranking does not change, there is a significant increase in the importance of the Indianapolis node, as well as noticeable decrease for all other nodes.

Node Removed	x^*	nc_i		Paths		Reroutings	
Base Case	5 803 801	0		1 176		15 438	
Indianapolis	6 337 184	414 546	7%	261	22%	2 950	19%
Pensecola	7 012 197	1 084 152	19%	360	31%	8 030	52%
Tucson	7 062 503	1 093 649	19%	294	25%	9 728	63%
Meadowlands	7 194 073	1 192 337	21%	261	22%	8 112	53%

Table 4.8: The new importance measures for regional distribution centres, when incident arcs to Indianapolis have their costs reduced by 25%

Arc Addition and Removal

Adding or removing arcs leading to or away from a particular node directly impacts the number of paths contributed by that node. Additionally, the total number of paths is altered as well, thus the path contribution measure for all nodes in the network is changed in every case.

When removing arcs, the path contribution of the affected node is lowered, and the total flow going through that node may decrease. The altered node is therefore less important to the network. Excess capacity is lost when arcs are removed, since the altered node is no longer capable of supplying certain other nodes. This can lead to an increase in network cost for neighbouring nodes. Moreover, neighbouring nodes will have their path contributions increase relative to the altered node, and may handle more flow, thus becoming more important.

Adding arcs has the opposite effect. The altered node can become more important, and all other nodes in the tier could become less so. As with previous variations, the relative ranking of nodes in the altered node's tier can change.

Supply and Demand

We inspect the effect of altering the supply or demand of suppliers and customers. The result can vary greatly based on circumstance. Here we present two examples which give contradictory outcomes when we alter the demand of customer nodes.

To illustrate the contradictory outcomes, we increase the supply for the Newark node. Table 4.9 shows the importance measures for the regional distribution centres. Notice that the Meadowlands and Pensecola nodes have a larger amount of reroutings, and that all other nodes have less reroutings and lower network costs, despite the fact that the ranking has not changed. Investigating the optimal solution, we see that Meadowlands is handling more flow, due to its proximity to the Newark node. This increases the importance of Meadowlands at the expense of the other regional distribution centres. It is important to note that spending this much effort identifying why Meadowlands' importance has increased while the other nodes have seen decreases is equivalent to calculating the importance of nodes by hand.

Node Removed	x^*	nc_i		Paths		Reroutings	
Base Case	5 803 801	0		1 176		15 438	
Indianapolis	5 861 184	5 669	0%	261	22%	2 788	18%
Pensecola	6 707 184	846 262	15%	360	31%	8 468	55%
Tucson	6 827 837	926 106	16%	294	25%	9 548	62%
Meadowlands	7 194 073	1 259 460	22%	261	22%	10 262	66%

Table 4.9: The new importance measures for regional distribution centres, when Newark's supply is increased to 1 815 units.

In fact, analysing this kind of change is closely related to trying to determine the impact a change has on the other tiers in the network. In previous analyses, we have mentioned that the ranking in the altered node's tier may be revised, but we omitted the impact on other tiers. This is because of the phenomenon observed above: Some nodes in different tiers may have their importance increased, while others may have their importance decreased. Outlining simple rules that explain the impact of a change in tier Y on nodes in tier X is therefore too complex. The rules would need too many clauses, hence we simply state that most rankings for the rest of the network will change, without explaining those changes.

Review of Impacts

Table 4.10 gives a summary of all the parameter changes, and the impacts they have on the rankings. It is important to note that in most cases, there is no guarantee that the change's impact will be reflected on all nodes. We can only say that any impact is possible with each parameter change.

We have given clear applications of the node-importance algorithm, as well as analysis of the effect of each parameter on the importance of a node. Chapter 5 explores some ideas for furthering research and applications of our node-importance algorithm.

Parameter Change	Importance of the altered node	Importance of neighbouring nodes	Relative ranking of other nodes
Increasing Node Cost	Decreases	No change	No reordering
Decreasing Node Cost	Increases	No change	No reordering
Increasing Node Capacity (non-binding)	No change	Decreases	Reordered
Decreasing Node Capacity (non-binding)	No change	Increases	Reordered
Increasing Node Capacity (binding)	Increases	Decreases	Reordered
Decreasing Node Capacity (binding)	Decreases	Increases	Reordered
Increasing Arc Cost	Decreases	Increases	Reordered
Decreasing Arc Cost	Increases	Decreases	Reordered
Adding Arcs	Increases	Decreases	Reordered
Removing Arcs	Decreases	Increases	Reordered

Table 4.10: Summary of the possible impacts each parameter change has on the altered node, as well as neighbouring nodes, and whether or not the nodes are reordered in the ranking. Each impact can cause the listed change, but is not guaranteed to do so.

Chapter 5

Conclusion

5.1 Summary of Results

We developed a ranking method for the nodes in physical distribution networks. Our method measures three metrics for each intermediate node in the network, which are combined to rank the nodes. In Chapter 3 we gave algorithms to compute each of our measures. Combining the algorithms from Chapter 3, we presented Algorithm 4 which gives the importance measures for all nodes in a network. Applying weights to the data from this algorithm, we rank nodes. We developed supporting algorithms to enumerate our networks, as well as solve large numbers of similar optimization problems, using the revised simplex method.

We demonstrated how the ranking method is used, as well as the impact each parameter has on the ranking of a node, in Chapter 4. By varying each parameter in turn, we showed that the impact of a parameter change is most often not restricted to the node directly affected by the parameter. Of particular interest is the difference between increasing the capacity of a node when the new capacity constraint is either binding or not. In Table 4.10 we see that altering the capacity of a node to a new, non-binding, value has no effect on the node itself, whereas a change to a binding value has a direct impact on the importance of the node. Both of these changes affect the neighbouring nodes in similar ways. Alternatively, changing the cost of a node has no impact on the neighbouring nodes, but directly affects the importance of the node in question.

5.2 A Transhipment Formulation and the Network Simplex Method

When doing our computations, we used a formulation that was not a proper transhipment problem. Through the use of a simple transformation, we can restate our problem as a transhipment problem. This gives the advantage of being able to use the network simplex method. The network simplex method is much more efficient at solving transhipment problems than the simplex method. Our stated interest in modifying the revised simplex method in order to warm start the optimization of each sub-network, was efficiency. We concede that using a formulation other than the transhipment problem is not ideal since the network simplex method is much more efficient than the simplex method.

Further work should be done to restate the problem as a transhipment problem, and to utilise the network simplex method to obtain optimal solutions. It would also be interesting to adapt the idea of a warm start to the network simplex method. It is not possible to directly apply the same concepts we developed in Section 3.3, since we used the point at which a constraint becomes active in the basis as the restarting point of the simplex method. This works because the initial basic feasible solution is made up entirely of slack variables. In the case of the network simplex method, the initial basic feasible solution is a spanning tree of all nodes in the network, along with flows. Many, although not all, nodes will be active in this initial solution, thus a warm start would be equivalent to restarting the network simplex method.

Adapting the network simplex method to exploit the similarity of each of our sub-networks is an area that could yield interesting results, with further research.

5.3 Future Research

In this work we were able to incorporate the importance of cost, network structure (through path counting) and disruptions (through reroutings). With a basic ranking method established, we can attempt to expand on the existing importance measures by including more complexity in our model. An example would be to consider lead time as a constraint, and measure the average lead time in the network when a node is present, and when it is removed. Another possible addition could be to incorporate multiple shipment methods, which would be represented as multiple arcs between certain nodes.

Since facility location is a large decision, due to the heavy cost of opening a new facility, incorporating the ideas of this importance algorithm into facility location models is another aspect that can be considered. Whereas the questions we asked were in relation to removing a node from a network, we can turn the question into adding a node to a network with some minor adjustments. Incorporating such an adjusted importance measure into facility location models could be beneficial to a company whose main concern is not just the strict economic gains of a certain expansion, but also the gains in integrity and robustness such an expansion would permit.

A related question would be whether it is better to have many somewhat important nodes in the network, or a small number of very important nodes. Even though our method gives relative importance, we still get an idea of absolute importance by looking at the spread of values obtained when compiling the final ranking. In cases where the reliability of a network is desired, it might be preferable to have a diffused network structure, where no one node is most important. Designing networks with this idea in mind could yield more secure structures with a high resilience to breakdowns or failures.

APPENDICES

Appendix A

Data for the Large Example

Here we present the data that makes up the large realistic problem from Section 4.2. Any data that has a value of 0 is omitted, to save space. We give the supply and demand values, the node capacity values, and the arc incidence and cost matrix. This constitutes all the data used for the realistic problem.

A.1 Supply and Demand

Negative values represent supply, positive values represent demand.

Node	Value
KCT	-3500
NOR	-2085

continued on next page

continued from previous page

Node	Value
NWK	-1415
NY	839
LA	383
CH	285
HO	226
PH	159
PI	155
SA	137
SD	131
DA	130
SJ	96
DT	91
SF	82
JC	81
IN	81
AU	79
CO	77
FW	73
CR	71
ME	68
BO	65

continued on next page

continued from previous page

Node	Value
BA	64
EP	62
SE	62
DE	61
NA	61
MI	61
WA	60
LV	57
LO	57
PO	57
OK	56
TU	54
AT	54
AL	53
KC	48
FR	48
MS	47
SM	47
LB	46
OM	45
VB	43

continued on next page

continued from previous page

Node	Value
MM	43
CL	43
OA	41
RA	41
CS	40
TL	39
MN	39
AG	38
WI	37
SL	36
NO	35
TA	34
ST	34
AH	34
CI	33
BK	32
AO	32
TD	32
PT	31

Table A.1: Supply and demand of nodes in the realistic example

A.2 Node Capacity

Here the capacity of source and customer nodes is assumed to be greater than their respective supply or demand value.

Node	Value
IND	1800
PEN	1900
TUC	2500
MED	2300
SPR	1300
ROC	1200
RIC	400
ATL	550
JAC	325
COL	600
MEM	650
DAL	750
SBR	800
SJO	750
SLC	700

Table A.2: Capacity of nodes in the realistic example

A.3 Arc incidence and cost

Here, we give the cost of arcs, only if they exist in the network. We break it down into three tables, to cut down on excess space. Note that an entry of 0 indicates a zero arc cost and not the absence of an arc. We denote absent arcs by a hyphen.

A.3.1 Source nodes to regional distribution centres

	Source		
Destination	KCT	NOR	NWK
IND	482	818	698
PEN	900	200	1198
TUC	1 237	1 407	2428
MED	1 190	1 300	0

Table A.3: Arc costs from source nodes to regional distribution centres

A.3.2 Regional distribution centres to local distribution centres

	Source			
Destination	IND	PEN	TUC	MED
SPR	893	-	-	152
ROC	569	-	-	324
RIC	619	854	-	328
ATL	533	322	1 733	877
JAC	849	356	-	931
COL	175	-	-	524
MEM	464	459	1 402	1 085
DAL	-	696	955	-
SBR	-	2 017	436	-
SJO	-	2 407	826	-
SLC	-	1 985	773	-

Table A.4: Arc costs from regional distribution centres to local distribution centres

A.3.3 Local distribution centres to customer nodes

	Source										
Destination	SPR	ROC	RIC	ATL	JAC	COL	MEM	DAL	SBR	SJO	SLC
NY	150	314	332	-	-	-	-	-	-	-	-
LA	-	-	-	-	-	-	-	-	60	340	689
CH	-	603	-	-	-	356	-	-	-	-	-
HO	-	-	-	-	871	-	570	242	-	-	-
PH	-	-	-	-	-	-	-	-	321	713	658
PI	245	321	243	-	-	-	-	-	-	-	-
SA	-	-	-	-	-	-	727	278	-	-	-
SD	-	-	-	-	-	-	-	-	107	460	750
DA	-	-	-	-	-	-	451	0	-	-	-
SJ	-	-	-	-	-	-	-	-	394	0	769
DT	-	338	-	-	-	191	-	-	-	-	-
SF	-	-	-	-	-	-	-	-	435	48	736
JC	-	-	-	317	0	-	-	-	-	-	-
IN	-	-	-	-	-	176	-	-	-	-	-

continued on next page

continued from previous page

Destination	Source										
	SPR	ROC	RIC	ATL	JAC	COL	MEM	DAL	SBR	SJO	SLC
AU	-	-	-	-	-	-	648	199	-	-	-
CO	687	395	477	-	-	0	-	-	-	-	-
FW	-	-	-	-	-	-	485	36	-	-	-
CR	-	-	292	244	383	-	620	-	-	-	-
ME	-	-	-	384	697	589	0	-	-	-	-
BO	90	393	-	-	-	-	-	-	-	-	-
BA	336	343	146	-	-	-	-	-	-	-	-
EP	-	-	-	-	-	-	-	638	-	-	863
SE	-	-	-	-	-	-	-	-	-	838	840
DE	-	-	-	-	-	-	-	797	-	-	536
NA	-	-	-	250	566	379	211	-	-	-	-
MI	-	-	-	-	-	442	-	-	-	-	-
WA	375	388	107	-	-	-	-	-	-	-	-
LV	-	-	-	-	-	-	-	-	221	522	425
LO	-	-	-	422	-	206	-	-	-	-	-
PO	-	-	-	-	-	-	-	-	-	665	766

continued on next page

continued from previous page

Destination	Source SPR	ROC	RIC	ATL	JAC	COL	MEM	DAL	SBR	SJO	SLC
OK	-	-	-	-	-	-	466	208	-	-	-
TU	-	-	-	-	-	-	-	-	436	829	773
AT	-	-	532	0	317	-	383	-	-	-	-
AL	-	-	-	-	-	-	-	648	742	-	-
KC	-	-	-	-	-	-	451	509	-	-	-
FR	-	-	-	-	-	-	-	-	273	152	818
MS	-	-	-	-	-	-	-	-	337	729	675
SM	-	-	-	-	-	-	-	-	438	118	649
LB	-	-	-	-	-	-	-	-	67	363	702
OM	-	-	-	-	-	-	646	661	-	-	-
VB	508	-	110	-	630	-	-	-	-	-	-
MM	-	-	-	663	345	-	-	-	-	-	-
CL	556	258	-	-	-	143	-	-	-	-	-
OA	-	-	-	-	-	-	-	-	424	41	730
RA	-	-	157	409	-	-	753	-	-	-	-
CS	-	-	-	-	-	-	-	726	-	-	603

continued on next page

continued from previous page

Destination	Source										
	SPR	ROC	RIC	ATL	JAC	COL	MEM	DAL	SBR	SJO	SLC
TL	-	-	-	-	-	-	402	256	-	-	-
MN	-	-	-	-	-	762	829	-	-	-	-
AG	390	-	105	-	-	-	-	-	-	-	-
WI	-	-	-	-	-	-	578	365	-	-	-
SL	-	-	-	-	-	420	285	-	-	-	-
NO	-	-	-	469	546	-	396	509	-	-	-
TA	-	-	-	458	201	-	-	-	-	-	-
ST	-	-	-	-	-	-	-	-	50	373	-
AH	-	-	-	-	-	-	-	-	47	366	682
CI	-	501	519	461	-	107	-	-	-	-	-
BK	-	-	-	-	-	-	-	-	165	242	704
AO	-	-	-	-	-	-	-	795	-	-	541
TD	-	373	-	-	-	144	-	-	-	-	-
PT	496	284	326	-	-	183	-	-	-	-	-

Table A.5: Arc costs from local distribution centres to customer nodes

Bibliography

Walmart: A leader in logistics, 2009. http://walmartstores.com/download/2336.pdf, accessed March 3rd, 2011.

Jussi Antikainen, Sami Repo, Pekka Verho, and Pertti Jarventausta. Possibilities to improve reliability of distribution network by intended island operation. *International Journal of Innovations in Energy Systems and Power*, 4(1), 2009. ISSN 1913-133X.

S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30:107–117, April 1998. ISSN 0169-7552.

P. M. S. Carvalho and L. A. F. M. Ferreira. Urban distribution network investment criteria for reliability adequacy. *IEEE Transactions on Power Systems*, 19(2):1216–1222, May 2004.

Stella Dafermos. The general multimodal network equilibrium problem with elastic demand. *Networks*, 12(1):57–72, March 1982.

Paul Dwyer. Measuring the value of electronic word of mouth and its impact in consumer communities. *Journal of Interactive Marketing*, 21(2):63–79, March 2007.

K. A. Hawick. Exploring data structures and tools for computations on graphs and networks. Technical Report 043, Massey University, New Zealand, 2007. Computational Science Technical Note Series, http://www.massey.ac.nz/ kahawick/cstn/043/cstn-043.html, accessed March 3rd, 2011.

K. A. Hawick and H. A. James. Node importance ranking and scaling properties of some complex road networks. *Research Letters in the Information and Mathematical Sciences*, 11:23–32, 2007.

James J. Kelleher. Tactical communications network modelling and reliability analysis: Overview. Technical report, Survivability Management Office, U.S. Army LABCOM, November 1991. JC-2091-GT-F3 under contract DAAL02-89-C-0040, http://w3.antd.nist.gov/wctg/netanal/gphthy_ kelleher.pdf accessed March 3rd, 2011.

Kazuhiro Kobayashi, Hozumi Morohosib, and Tatsuo Oyamab. Applying path-counting methods for measuring the robustness of the network-structured system. *International Transactions in Operational Research*, 16(3):371–389, May 2009.

Lv Le and Yu Hewei. A new method for evaluating node importance in complex networks based on data field theory. *Proceedings of the 1st International Conference on Networking and Distributed Computing*, pages 133–136, 2010.

Fengwei Li and Xueliang Li. On the integrity of graphs. In T. Gonzalez, editor, *Proceedings of the 16th IASTED International Conference on Parallel and Distributed Computing and Systems*, pages 577–582, November 2004. ISBN 10272658.

Gao Lianxiong, Wu Jianping, and Liu Rui. Key nodes mining in transport networks based on pagerank algorithm. In *2009 Chinese Control and Decision Conference*. IEEE Computer Society, 2009. ISBN 9781424427239.

Hokey Min and Gengui Zhou. Supply chain modeling: past, present and future. *Computers and Industrial Engineering*, 43(1-2):231–249, July 2002.

Michel Minoux. Discrete cost multicommodity network optimization problems and exact solution methods. *Annals of Operations Research*, 106(1):19–46, 20010901 2001.

Anna Nagurney. On the relationship between supply chain and transportation network equilibria: A supernetwork equivalence with computations. *Transportation Research.Part E, Logistics & Transportation Review*, 42E(4):293–316, 20060701 2006.

M.E.J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2): 167–256, June 2003.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. URL `http://ilpubs.stanford.edu:8090/422/`. Accessed March 3rd, 2011.

Q. Qiang and A. Nagurney. A unified network performance measure with importance identification and the ranking of network components. *Optimization Letters*, 2(1):127–142, 2008.

The Boyd Company Inc. Comparative distribution warehousing industry operating costs. Technical report, The Boyd Company, Inc., 2010.

Scott White and Padhraic Smyth. Algorithms for estimating relative importance in networks. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 266–275, 2003.