Painted network flows with weighted divergence

Abdelghani Mehailia
Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/rtd

Part of the Mathematics Commons

Recommended Citation
https://lib.dr.iastate.edu/rtd/8792
INFORMATION TO USERS

The most advanced technology has been used to photograph and reproduce this manuscript from the microfilm master. UMI films the original text directly from the copy submitted. Thus, some dissertation copies are in typewriter face, while others may be from a computer printer.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyrighted material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each oversize page is available as one exposure on a standard 35 mm slide or as a 17" × 23" black and white photographic print for an additional charge.

Photographs included in the original manuscript have been reproduced xerographically in this copy. 35 mm slides or 6" × 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.
Painted network flows with weighted divergence

Mehailia, Abdelghani, Ph.D.
Iowa State University, 1988
PLEASE NOTE:

In all cases this material has been filmed in the best possible way from the available copy. Problems encountered with this document have been identified here with a check mark ✓.

1. Glossy photographs or pages
2. Colored illustrations, paper or print
3. Photographs with dark background
4. Illustrations are poor copy
5. Pages with black marks, not original copy
6. Print shows through as there is text on both sides of page
7. Indistinct, broken or small print on several pages ✓
8. Print exceeds margin requirements
9. Tightly bound copy with print lost in spine
10. Computer printout pages with indistinct print
11. Page(s) lacking when material received, and not available from school or author.
12. Page(s) seem to be missing in numbering only as text follows.
13. Two pages numbered. Text follows.
14. Curling and wrinkled pages
15. Dissertation contains pages with print at a slant, filmed as received ✓
16. Other

UMI
Painted network flows with weighted divergence

by

Abdelghani Mehailia

A Dissertation Submitted to the
Graduate Faculty in Partial Fulfillment of the
Requirements for the Degree of
DOCTOR OF PHILOSOPHY

Department: Mathematics
Major: Applied Mathematics

Approved:

In Charge of Major Work

For the Major Department

For the Graduate College

Iowa State University
Ames, Iowa
1988
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>NOMENCLATURE</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>vi</td>
</tr>
</tbody>
</table>

## 1. INTRODUCTION

## 2. GENERAL BACKGROUND

### 2.1. Definition of a Network

#### 2.1.1. Definitions

### 2.2. The Painted Path Problem

#### 2.2.1. Definitions

#### 2.2.2. Statement of the problem

### 2.3. The Painted Cut Problem

#### 2.3.1. Definitions

#### 2.3.2. Statement of the problem

### 2.4. The Painted Network Algorithm

#### 2.4.1. Definitions

#### 2.4.2. Statement of the algorithm

#### 2.4.3. The painted network theorem

## 3. MAXIMUM WEIGHTED FLOW PROBLEM

### 3.1. Description of the Network

#### 3.1.1. Assumptions and definitions

### 3.2. The Maximum Weighted Flow Problem

#### 3.2.1. The total weighted divergence principle

#### 3.2.2. Description of the problem

#### 3.2.3. Statement of the problem

#### 3.2.4. The fundamental weighted divergence principle

#### 3.2.5. The maximum weighted flow-min weighted cut problem

### 3.3. The Maximum Weighted Flow Algorithm

## 4. Conclusion
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4</td>
<td>Justification of the Algorithm</td>
<td>21</td>
</tr>
<tr>
<td>3.5</td>
<td>Example</td>
<td>25</td>
</tr>
<tr>
<td>4.</td>
<td>THE FEASIBLE DISTRIBUTION PROBLEM</td>
<td>31</td>
</tr>
<tr>
<td>4.1</td>
<td>Statement of the Problem</td>
<td>31</td>
</tr>
<tr>
<td>4.2</td>
<td>The Feasible Distribution Theorem</td>
<td>31</td>
</tr>
<tr>
<td>4.3</td>
<td>The Feasible Distribution Algorithm</td>
<td>33</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Description of the algorithm</td>
<td>33</td>
</tr>
<tr>
<td>4.4</td>
<td>Justification of the Algorithm</td>
<td>34</td>
</tr>
<tr>
<td>4.5</td>
<td>Examples</td>
<td>36</td>
</tr>
<tr>
<td>4.5.1</td>
<td>A feasible distribution problem which has no solution</td>
<td>36</td>
</tr>
<tr>
<td>4.5.2</td>
<td>A feasible distribution problem which has a solution</td>
<td>41</td>
</tr>
<tr>
<td>5.</td>
<td>THE MAXIMUM WEIGHTED TENSION PROBLEM AND ITS DUAL PROBLEM</td>
<td>46</td>
</tr>
<tr>
<td>5.1</td>
<td>The Maximum Weighted Tension Problem</td>
<td>46</td>
</tr>
<tr>
<td>5.1.1</td>
<td>Assumptions and definitions</td>
<td>46</td>
</tr>
<tr>
<td>5.1.2</td>
<td>Description of the problem</td>
<td>47</td>
</tr>
<tr>
<td>5.1.3</td>
<td>Statement of the problem</td>
<td>47</td>
</tr>
<tr>
<td>5.2</td>
<td>The Maximum Weighted Tension Algorithm</td>
<td>48</td>
</tr>
<tr>
<td>5.3</td>
<td>Justification of the Algorithm</td>
<td>49</td>
</tr>
<tr>
<td>5.4</td>
<td>Example</td>
<td>50</td>
</tr>
<tr>
<td>5.5</td>
<td>The Minimum Path Problem</td>
<td>54</td>
</tr>
<tr>
<td>5.5.1</td>
<td>Assumptions and definitions</td>
<td>54</td>
</tr>
<tr>
<td>5.5.2</td>
<td>Statement of the problem</td>
<td>54</td>
</tr>
<tr>
<td>5.5.3</td>
<td>Max weighted tension-min path theorem</td>
<td>54</td>
</tr>
<tr>
<td>5.6</td>
<td>The Minimum Path Problem</td>
<td>55</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>6. THE WEIGHTED TENSION RECTIFICATION ALGORITHM</td>
<td>56</td>
<td></td>
</tr>
<tr>
<td>6.1. The Feasible Differential Problem</td>
<td>56</td>
<td></td>
</tr>
<tr>
<td>6.1.1. Description of the problem</td>
<td>56</td>
<td></td>
</tr>
<tr>
<td>6.1.2. Statement of the problem</td>
<td>56</td>
<td></td>
</tr>
<tr>
<td>6.2. The Weighted Tension Rectification Algorithm</td>
<td>56</td>
<td></td>
</tr>
<tr>
<td>6.3. Justification of the Algorithm</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>6.4. Example</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>7. THE WEIGHTED LINEAR OPTIMAL DISTRIBUTION AND APPLICATIONS</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td>7.1. The Weighted Linear Optimal Distribution Problem</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td>7.1.1. Description of the problem</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td>7.1.2. Statement of the problem</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td>7.1.3. A duality theorem</td>
<td>67</td>
<td></td>
</tr>
<tr>
<td>7.2. Bechhofer and Kulkarni's Problem</td>
<td>71</td>
<td></td>
</tr>
<tr>
<td>7.2.1. Description of the problem</td>
<td>71</td>
<td></td>
</tr>
<tr>
<td>7.2.2. Formulation of Bechhofer and Kulkarni's problem</td>
<td>73</td>
<td></td>
</tr>
<tr>
<td>8. CONCLUSION</td>
<td>81</td>
<td></td>
</tr>
<tr>
<td>8.1. Summary</td>
<td>81</td>
<td></td>
</tr>
<tr>
<td>8.2. Suggestions for Future Research</td>
<td>81</td>
<td></td>
</tr>
<tr>
<td>9. BIBLIOGRAPHY</td>
<td>92</td>
<td></td>
</tr>
<tr>
<td>10. ACKNOWLEDGMENTS</td>
<td>84</td>
<td></td>
</tr>
<tr>
<td>11. APPENDIX A: THE WEIGHTED FLOW RECTIFICATION ALGORITHM</td>
<td>85</td>
<td></td>
</tr>
<tr>
<td>11.1. Description of the Algorithm</td>
<td>85</td>
<td></td>
</tr>
<tr>
<td>11.2. Justification of the Algorithm</td>
<td>86</td>
<td></td>
</tr>
<tr>
<td>11.3. Example</td>
<td>88</td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX B: THE WEIGHTED LINEAR OPTIMAL DISTRIBUTION ALGORITHM

12. Description of the Algorithm
12.2. Justification of the Algorithm
12.3. Example
NOMENCLATURE

G  Network
A  Set of all arcs of G
N  Set of all nodes of G
i  Node
j~(i,i')  Arc incident to i and i'
P  Path
Q  Cut
e_P  Incidence function of P
e_Q  Incidence function of Q
e_N  Incidence function of N
e(i,j)  Node-arc incidence function of G
E  Node-arc incidence matrix of G
x(j)  flux in the arc j
y(i)  Divergence of the flow at node i
u(i)  Potential at node i
v(j)  Tension (differential) across j
θ  Routing
m  Number of nodes in N
n  Number of arcs in A
P^+  Positive arcs of P
P^-  Negative arcs of P
Q^+  Positive arcs of Q
Q^-  Negative arcs of Q
C  
C+  
C-  
p(j)  
\tilde{y}(i)  
\tilde{y}(S)  
S  
e_Q \cdot \tilde{x}  
C^+(Q)  
C^-(Q)  
x'  
u'  
v'  
\tilde{y}_x  
\tilde{y}_{x'}  
b(i)  
b(S)

Capacity interval
Upper bound of C
Lower bound of C
Weight on arc j
Weighted divergence at node i
Weighted divergence from S
Node set
Weighted flux of x across Q
Upper bound of $e_Q \cdot c^+$
Lower bound of $e_Q \cdot c^-$
Augmented flow
Augmented potential
Augmented tension
Weighted divergence of x
Weighted divergence of $x'$
Supply at node i
Net supply in S
1. INTRODUCTION

In the period following World War II, it began to be recognized that there were a large number of interesting and significant activities which could be classified as multistage decision processes. It was soon seen that the mathematical problems that arose in their study stretched the conventional confines of analysis, and required new methods for their successful treatment. The classical techniques of calculus and the calculus of variations were occasionally valuable and useful in these new areas, but were clearly limited in range and versatility, and were definitely lacking as far as furnishing numerical answers was concerned.

The recognition of these facts led to the creation of a number of novel mathematical theories and methods. Among these were the theories of linear programming, dynamic programming and network flows. The theory of linear programming is designed to treat processes possessing certain features of linearity, and the elegant "simplex method" of George Dantzig (1948, 1963) to a large extent solves the problem for these processes. Dynamic programming is designed to treat multistage processes possessing certain invariant aspects. We mention Bellman (1957) as a reference on the subject. The theory of network flows deals with problems which can be represented as networks and like linear programming, provides a general framework for formulating and solving a considerable number of optimization problems. We mention Busacker

The flow $x$ is a function on the arcs. Most of the classical theory of network flows like it is treated in Rockafellar (1984) assumes a conservation property for the flow which holds at every node of the network. Namely, the net flow at each node $i$ is zero. In this study, we allow the flow to be weighted so that losses and gains of the flow may occur at each node of the network.

Chapter 2 consists of the basic concepts and definitions needed and that will be used in the later chapters. The notation and terminology are from Rockafellar (1984).

Chapter 3 treats a maximum weighted flow problem. An algorithm to solve the problem is described. At the end of the chapter, an example of how the algorithm can be implemented is presented.

Chapter 4 contains an algorithm which consists of the feasibility part (phase 1) of the maximum weighted flow algorithm described in the preceding chapter. An example of how the algorithm works is included at the end of the chapter. A different algorithm but which also solves the feasibility part of the max weighted flow algorithm is presented in Chapter 11 (Appendix A) along with an illustrative example.

In Chapter 5, a maximum weighted tension problem is discussed
and the maximum weighted tension algorithm is presented as a way to solve the problem. An illustrative example is included.

Chapter 6 treats the feasibility part of Chapter 5. The weighted tension rectification algorithm is presented with an example.

Statisticians have been developing the theory of ordering (or ranking) and selection procedures for over 25 years. Among the early selection procedures proposed was the single-stage procedure for selecting the best one of several binomial populations by Sobel and Huyett (1957). In the last 10 to 15 years, the rate of development has increased considerably. Many publications in the statistical literature have been devoted to various theoretical aspects of ordering and selection problems. As a good bibliography of these publications, we suggest Gibbons et al. (1977). Bechhofer (1985) and Kulkarni (1981) developed an optimal multistage procedure for selecting the best one among k populations. Unlike the single-stage procedure of Sobel and Huyett, this multistage procedure minimizes the expected number of observations that need to be sampled before the stopping rule is reached. It has the same probability of selecting the best population as the single-stage procedure. Kulkarni (1981) and Jennison (1985) proved by using dynamic programming that in the case of two populations, the multistage procedure is optimal if and only if a given condition on the probabilities of success of the two populations was met. In the case when k \((k \geq 3)\) populations are considered, that condition is only sufficient and is not
necessary to have the multistage procedure optimal.

In Chapter 7, a weighted divergence linear optimal distribution problem is discussed. A proof of duality theorem is provided. We formulate the decision-making problem of Bechhofer and Kulkarni as a special linear optimal distribution problem. An algorithm which solves the weighted linear optimal distribution problem is described in Chapter 12 (Appendix B) with an illustrative example.
2. GENERAL BACKGROUND

A fairly brief introductory background to the theory of the network flows is necessary for developing the results that will be discussed in the next chapters.

2.1. Definition of a Network

2.1.1. Definitions

a) Definition: A network is defined as a triple consisting of two abstract sets A and N, and a function that assigns to each \( j \in A \) a pair \((i, i') \in N \times N\) such that \(i \neq i'\). The elements of A are called arcs (edges or links), and those of N nodes; it is assumed \(N \neq \emptyset\). The function is just described as \(j \sim (i, i')\), i and i' are called the initial node and the terminal node of j, respectively. The arc j is said to be incident to i and i'.

b) Definition: A path \( P \) in a network G is a finite sequence \(i_0, j_1, i_1, j_2, i_2, \ldots, j_k, i_k\) \((k > 0)\) where \(i_m\) denotes a node, \(j_m\) denotes an arc and either \(j_m \sim (i_{m-1}, i_m)\) or \(j_m \sim (i_m, i_{m-1})\). When \(i_0 = i_k\), \(P\) is called a circuit.

c) Definition:

i) An arc \(j_m\) in \(P\) is said to be traversed positively (or simply positive) or negatively (or simply negative) according to whether \(j_m \sim (i_{m-1}, i_m)\) or \(j_m \sim (i_m, i_{m-1})\).
ii) An elementary path is a path which uses no node more than once, except for the initial and terminal nodes. $P^+$ is defined to be the set of positive arcs and $P^-$ the set of negative arcs.

An important question arises in the theory of network flows; that is, how a network $G$ and its associated flows and potentials may be represented numerically. That leads to the following series of definitions.

d) Definition: The node-arc incidence function of $G$ is defined by

\[
e(i,j) = \begin{cases} 
1 & \text{if } i \text{ is the initial node of the arc } j \\
-1 & \text{if } i \text{ is the final node of the arc } j \\
0 & \text{in all other cases}
\end{cases}
\]

This function is often expressed in terms of the node-arc incidence matrix. For instance, if we let $N = \{i_1, i_2, \ldots, i_m\}$ and $A = \{j_1, j_2, \ldots, j_n\}$, the incidence matrix $E$ has in each column exactly one +1 and one -1. (Each arc of $A$ has exactly one initial node and exactly one final node.) In matrix notation, $E = (e_{ik})_{1 \leq k \leq m \atop 1 \leq i \leq n}$. Conversely, any $m \times n$ matrix $E$ with this property can be interpreted as the incidence matrix for a certain uniquely determined network $G$ with $|N| = m$ and $|A| = n$.

e) Definition: Similarly, for an elementary path $P$, the incidence function for $P$ is defined by

\[
e_P(j) = \begin{cases} 
1 & \text{if } j \in P^+ \\
-1 & \text{if } j \in P^- \\
0 & \text{otherwise.}
\end{cases}
\]
f) Definition: A flow in a network $G$ is meant to be nothing more than an arbitrary function $x$ on the arcs $x: A \rightarrow \mathbb{R}$. The value $x(j)$, called the flux in the arc $j$ is interpreted in most applications as the amount of material flowing in the arc $j$.

g) Definition: The divergence of the flow at $i$ is the quantity denoted by $y(i)$ and defined by $y(i) = \sum_{j \in A} e(i,j)x(j)$.

h) Definition: A node $i$ is said to be a source for the flow if $y(i) > 0$ and a sink if $y(i) < 0$. If $y(i) = 0$, the flow is said to be conserved at $i$.

i) Definition: A flow is called a circulation if the flow is conserved at every node, i.e., $y(i) = 0$ for all $i \in N$.

REMARK: Sums and scalar multiples of circulations are circulations as well.

j) Definition: A potential in $G$ is an arbitrary real-valued function defined on the node set $N$. The potential at node $i$ is the value $u(i)$.

k) Definition: Let $j$ be an arc of $G$, say $j \sim (i, i')$, the potential difference $v(j) = u(i') - u(i)$ is called the tension across the arc $j$. In other words, the tension function $v$ on $A$ is the differential of the potential $u$.

All the paths in this dissertation are elementary.
2.2. The Painted Path Problem

2.2.1. Definitions

a) Definition: A network G is said to be connected if for every pair of distinct nodes i and i', there exists a path P: i → i'.

A network may have a more restrictive property than connectedness, namely, strong connectedness.

b) Definition: A network G is said to be strongly connected if for every pair of distinct nodes i and i', there exists a positive path P: i → i'.

The next theorem could be easily considered the backbone of the theory of network flows. What is the efficient way to test whether a given network is connected or strongly connected?

The fundamental idea due to G. J. Minty (1960) is to partition the arc set A of the network G into four disjoint subsets, some of which might be empty. In any compatible path, some arcs can be in $P^+$ or $P^-$ (two-way), some in $P^+$ only (one-way forward), some in $P^-$ only (one-way backward), and some forbidden (no-way). The arcs of each one of the four categories are painted one of the four possible colors. The colors green, white, black and red correspond respectively to the four categories.
2.2.2. Statement of the problem

Let G be a network, $N^+$ and $N^-$ be two nonempty disjoint node sets of G. The arcs of G are painted by the colors green, white, black and red. The problem consists of finding a path $P: N^+ \rightarrow N^-$ (i.e., with initial node in $N^+$, and terminal node in $N^-$) which is compatible with the painting.

2.3. The Painted Cut Problem

As in most of modern optimization, duality plays an important role. From the painted path problem follows the painted cut problem which is based upon the concept of a "cut" which plays a role dual to that of "path".

2.3.1. Definitions

a) Definition: Let S be a given node set in a network G. The two parts of arcs $Q^+$ and $Q^-$ are defined as follows:

$$Q^+ = [S, N - S]^+ = \{j \in A / j \sim (i, i') \text{ with } i \in S, i' \notin S\}$$

$$Q^- = [S, N - S]^-[i \in A / j \sim (i', i) \text{ with } i \in S, i' \notin S]$$

But a cut Q in the network G, we mean the arc set $Q = Q^+ \cup Q^-$. The notation $Q = [S, N - S]$ is adopted.

The terminology "cut" for $Q = [S, N - S]$ is derived from the idea that any path $P$ with initial node in $S$ and terminal node in $N - S$ must, at some stage, traverse one of the arcs in Q; the suppression of the
arcs Q would thus "cut" all such paths.

b) Definition: The incidence function for a cut Q is defined by

\[ e_Q(j) = \begin{cases} 
  1 & \text{if } j \in Q^+ \\
  -1 & \text{if } j \in Q^- \\
  0 & \text{otherwise.} 
\end{cases} \]

2.3.2. Statement of the problem

Let G be a network. \( N^+ \) and \( N^- \) are two nonempty disjoint node sets. The painting of the painted path problem is used on the arcs. The problem is to determine a cut \( Q: N^+ \cap N^- \) such that every arc in \( Q^+ \) is red or black, whereas every arc in \( Q^- \) is red or white. We write \( Q: N^+ \cap N^- \) and say that \( Q \) is a cut that separates \( N^+ \) from \( N^- \) if \( Q = [S, N - S] \) for some node set \( S \) such that \( S \supseteq N^+ \) and \( S \cap N^- = \emptyset \). A cut is illustrated in Figure 2.1.

2.4. The Painted Network Algorithm

2.4.1. Definitions

a) Definition: For a given node set \( S \supseteq N^+ \), define a function \( \theta: S - N^+ \to A \) satisfying the following:

i) For each \( i \in S - N^+ \), \( \theta(i) \) is an arc joining \( i \) to another node in \( S \).

ii) Whenever a sequence of the form \( i_1, \theta(i_1), i_2, \theta(i_2), \ldots \), etc. is generated, a node in \( N^+ \) is eventually reached.
Figure 2.1. A cut that separates $N^+$ from $N^-$

The function $\theta$ is called a routing of $S$ with base $N^+$. When the sequence in (b) is generated and $N^+$ is eventually reached, the reverse of the sequence is a path from $N^+$ to $i$ that does not use any nodes outside $S$. It is called the path to it associated with the routing $\theta$, or simply the $\theta$-path from $N^+$ to $i$. Figure 2.2 illustrates a particular routing in a given network.

2.4.2. Statement of the algorithm

Step 0: Given $S \supseteq N^+$ with $S \cap N^- = \emptyset$ and a routing $\theta$, examine the cut $Q = [S, N - S]$. 
Step 1: If $Q^+$ has an arc $j$ which is green or white, then let $\theta(i) = j$ where $i$ is the end node of $j$ not in $S$. Replace $S$ by $S \cup \{i\}$ and go to step 4.

Step 2: If $Q^-$ has an arc $j$ which is green or black, then let $\theta(i) = j$ where $i$ is the end node of $j$ not in $S$. Replace $S$ by $S \cup \{i\}$ and go to step 4.

Step 3: If neither of the above steps, then stop. There is no solution to the painted path problem.

$\theta(i_4) = j_1, \theta(i_5) = j_3, \theta(i_3) = j_2$

Figure 2.2. A routing $\theta$ of $S$ with base $N^+$
Step 4: If $i \in N^-$, then stop. The reverse of the sequence in the routing $\theta$ is a path $P$ from $N^+$ to $i$ which solves the painted path problem.

Step 5: If $i \notin N^-$ ($S \cap N^- = \emptyset$ again) go to step 0.

Initially, the algorithm starts with $S = N^+$ and $\theta = \emptyset$.

2.4.3. The painted network theorem

Let $N^+$ and $N^-$ be disjoint node sets in the network $G$ painted with the four colors. Exactly one of the following problems has a solution:

1. The painted path problem
2. The painted cut problem

In other terms, the painted network algorithm always ends up with a path or with a cut.

Proof: The painted network algorithm has only two possible alternatives either to stop at step 3 or step 4. If the algorithm stops at step 3, there is a cut $Q$ which solves the painted cut problem. Otherwise, if step 4 is reached, a solution to the painted path problem is found. So, alternatives 1 and 2 are mutually exclusive as claimed.
3. MAXIMUM WEIGHTED FLOW PROBLEM

3.1. Description of the Network

3.1.1. Assumptions and definitions

Let $G$ be a network with a finite number of arcs and nodes, $x(j)$ is the flux in the arc $j$ such that $x(j) \in C(j) = [c^-(j), c^+(j)]$ where $c^+(j)$ could be $+\infty$ and $c^-(j)$ could be $-\infty$; in which case, we write $C(j) = (-\infty, \infty)$. Otherwise, if both $c^+(j)$ and $c^-(j)$ are finite, then the notation $C(j) = [c^-(j), c^+(j)]$ will be used.

For each arc $j \in A$, we assume that there exists a positive weight $p(j)$ attached to it. The weight $p(j)$ could have many practical meanings such as a toll a traveling salesman might have to pay if he chooses to go over a particular arc $j$, or a discount rate that a person may receive if they choose to travel the arc $j$ as opposed to another different arc. The weight $p(j)$ could also mean the probability of crossing the arc $j$ or also the reliability of a given system once it reaches the initial node of the arc $j$.

As a matter of fact, there are no restrictions put on the weights $p(j)$ except the one that they are positive weights in the true sense of the word, that is, $p(j) \in (0, \infty)$ for all $j \in A$. In Chapter 7, the weights $p(j)$ will be assumed to be conditional probabilities. Despite the stochastic nature of the problems treated in the present and upcoming chapters, we will refrain from using that terminology. The terminology "weighted" will be adopted instead.
The "net weighted flow out of node i" denoted by $\tilde{y}(i)$ is defined by $\tilde{y}(i) = \sum_{j \in A} p(j)x(j)e(i,j)$. It follows that the weighted divergence of the flow $x$ from the node set $S$ is defined by $\tilde{y}(S) = \sum_{i \in S} \tilde{y}(i)$. The quantity represents the net weighted amount of material originating in $S$.

### 3.2. The Maximum Weighted Flow Problem

#### 3.2.1. The total weighted divergence principle

It states that the total weighted divergence of the flow $x$ from $N$ is zero.

Proof:

$$\tilde{y}(N) = 1^T \cdot y = \sum_{i \in N} \tilde{y}(i) = \sum_{i \in N} \left( \sum_{j \in A} x(j)p(j)e(i,j) \right)$$

$$= \sum_{j \in A} p(j)x(j)\left( \sum_{i \in N} e(i,j) \right)$$

Each column of the node-arc incidence matrix $E$ has exactly one +1 and one -1 and all the rest of the entries are zeros. Therefore, it follows directly that $\sum_{i \in N} e(i,j) = 0$ for all $j \in A$ which implies $\tilde{y}(N) = 0$.

#### 3.2.2. Description of the problem

Let $G$ be a network with capacity intervals, and let $N^+$ and $N^-$ be nonempty disjoint sets of nodes of $G$. Let $x$ be any flow such that its weighted divergence is conserved at all nodes outside $N^+$ and $N^-$. That
is, \( \tilde{y}(i) = 0 \) for all \( i \not\in N^+ \cup N^- \). By the total weighted divergence principle, we have \( 0 = \sum_{i \in N^-} \tilde{y}(i) = \sum_{i \in N^+} \tilde{y}(i) + \sum_{i \in N^-} \tilde{y}(i) \) which yields 
\[ \tilde{y}(N^+) = -\tilde{y}(N^-). \]
\( \tilde{y}(N^+) \) represents the net weighted amount of source in \( N^+ \), whereas \( -\tilde{y}(N^-) \) is the net weighted amount of sink in \( N^- \). The common value is interpreted as the net weighted amount flowing from \( N^+ \) to \( N^- \). It will be called the weighted flux of \( x \) from \( N^+ \) to \( N^- \).

### 3.2.3. Statement of the problem

Maximize the weighted flux from \( N^+ \) to \( N^- \) over the set of all flows \( x \) that are conserved at all nodes outside \( N^+ \) and \( N^- \) as well as feasible with respect to capacities.

**REMARK:** The max weighted flow problem may be formulated as a linear programming problem as follows:

Maximize 
\[
\sum_{i \in N^+} \sum_{j \in A} e(i,j) x(j)p(j) = \text{Maximize} \sum_{j \in A} \left[ \sum_{i \in N^+} e(i,j) \right] p(j)x(j)
\]

subject to the constraints:

\[
\sum_{j \in A} e(i,j)p(j)x(j) = 0 \quad \text{for every } i \not\in N^+ \cup N^-
\]

\[
x(j) \leq c^+(j) \quad \text{for every } j \in A
\]

\[
x(j) \geq c^-(j) \quad \text{for every } j \in A
\]

Inequalities where \( c^+(j) = \infty \) or \( c^-(j) = -\infty \) are redundant and may be omitted.
For a cut $Q$ and flow $x$, the quantity $e_Q \cdot \bar{x} = \sum_{j \in Q^+} p(j)x(j) - \sum_{j \in Q^-} p(j)x(j)$ is called the weighted flux of $x$ across the cut $Q$. It may be interpreted as the net weighted amount of material across $Q$ in the direction of the orientation of $Q$.

3.2.4. The fundamental weighted divergence principle

\[ \text{[weighted divergence of } x \text{ from } S] = \text{[weighted flux of } x \text{ across } Q] \]

**Proof:**

\[
\bar{y}(S) = \sum_{i \in S} \bar{y}(i) = \sum_{i \in S} \sum_{i \in A} e(i,j)x(j)p(j) = \sum_{i \in S} \sum_{i \in A} e(i,j)x(j)p(j) \\
= \sum_{j \in A_1} \sum_{i \in S} e(i,j)x(j)p(j) + \sum_{j \in A_2} \sum_{i \in S} e(i,j)x(j)p(j) \\
+ \sum_{j \in Q} \sum_{i \in S} e(i,j)x(j)p(j)
\]

where $A_1 = \{j \in A \text{ such that both ends of } j \text{ are in } S\}$

$A_2 = \{j \in A \text{ such that both ends of } j \text{ are not in } S\}$

\[
\sum_{j \in A_1} \sum_{i \in S} e(i,j)x(j)p(j) = 0 \quad \text{and} \quad \sum_{j \in A_2} \sum_{i \in S} e(i,j)x(j)p(j) = 0
\]

which leads to

\[
\bar{y}(S) = \sum_{j \in Q} \sum_{i \in S} e(i,j)x(j)p(j) = \sum_{j \in Q^+} \sum_{i \in S} e(i,j)x(j)p(j) \\
+ \sum_{j \in Q^-} \sum_{i \in S} e(i,j)x(j)p(j)
\]
For each \( j \in Q^+ \), there is only one nonzero term \( e(i,j) = 1 \), and for each \( j \in Q^- \), there is only one nonzero term \( e(i,j) = -1 \). Therefore,
\[
\tilde{y}(S) = \sum_{j \in Q^+} p(j)x(j) - \sum_{j \in Q^-} p(j)x(j).
\]

For a flow \( x \) across \( Q \), we have
\[
c^-(j) \leq x(j) \leq c^+(j)
\]
and
\[
-c^+(j) \leq -x(j) \leq -c^-(j)
\]

Multiplying the above inequalities by \( p(j) \) and adding up, we obtain
\[
\sum_{j \in Q^-} c^-(j)p(j) - \sum_{j \in Q^+} c^+(j)p(j) \leq e \cdot \tilde{x} \leq \sum_{j \in Q^-} c^+(j)p(j) - \sum_{j \in Q^+} c^-(j)p(j)
\]
or
\[
C^-(Q) \leq e \cdot \tilde{x} \leq C^+(Q),
\]
where
\[
C^+(Q) = \sum_{j \in Q^+} c^+(j)p(j) - \sum_{j \in Q^-} c^-(j)p(j)
\]
and
\[
C^-(Q) = \sum_{j \in Q^-} c^-(j)p(j) - \sum_{j \in Q^+} c^+(j)p(j).
\]

**Corollary 1.** Let \( x \) be a feasible flow such that \( \tilde{y}(i) = 0 \) \( \forall i \notin N^+ \cup N^- \).

Then, the weighted flux across the cut \([N^+, N - N^+]\) = weighted flux across the cut \([N^-, N - N^-]\).

**Proof:** By the fundamental weighted divergence principle, \( y(N^+) \) = weighted flux across \([N^+, N - N^+]\) and \( y(N^-) \) = weighted flux across \([N^-, N - N^+]\). The result follows directly from the total weighted divergence principle.
Let $Q$ be a cut such that $Q = [S, N - S]$ where $S \supset N^+$ and $S \cap N^- = \emptyset$.

**Corollary 2.** Weighted flux of $x$ from $N^+$ to $N^-$ $\leq C^+(Q)$.

**Proof:** $\bar{y}(i) = 0$ for all $i \in S - N^+$ implies $\bar{y}(N^+) = \bar{y}(S)$

We know $\bar{y}(S) \leq C^+(Q)$

By the fundamental weighted divergence principle, $\bar{y}(S) = e_Q \cdot \bar{x}$

$\therefore e_Q \cdot \bar{x} \leq C^+(Q)$.

This leads to the minimum weighted cut problem which is to minimize $C^+(Q)$ over all cuts $Q = [S, N - S]$ such that $S \supset N^+$ and $S \cap N^- = \emptyset$.

This problem always has a solution since there are only finitely many cuts and by Corollary 2, $\bar{y}(N^+) \leq \min_{Q: N^+ \to N^-} C^+(Q)$.

3.2.5. The maximum weighted flow-minimum weighted cut theorem

We assume there is at least one flow satisfying the constraints of the max weighted flow problem. Then,

$$\left[ \sup \text{ in max weighted flow problem} \right] \left[ \min \text{ in min weighted cut problem} \right]$$

If there is an elementary path $P: N^+ \to N^-$ of unlimited capacity, the common value is $+\infty$, whereas if there is no such path, it is finite and the max weighted flow problem has a solution.

There are only a finite number of cuts, so the min weighted cut problem has a solution.
Proof: By Corollary 2, $\sup$ in max weighted flow problem $\leq$ $\min$ weighted cut problem.

To show equality, we construct an algorithm whose final stage yields a flow $x$ and a cut $Q$ for which the equality in Corollary 2 holds.

3.3. The Maximum Weighted Flow Algorithm

The following assumptions are made:

i) There is no path of unlimited capacity from $N^+$ to $N^-$ (Otherwise, the problem is trivial.)

ii) There exists a flow that satisfies the constraints of the max weighted flow problem.

Algorithm:

Step 0: The painting of the painted network algorithm is used.

- green if $c^-(j) < x(j) < c^+(j)$
- white if $c^-(j) = x(j) < c^+(j)$
- black if $c^-(j) < x(j) = c^+(j)$
- red if $c^-(j) = x(j) = c^+(j)$

Step 1: If a solution $Q: N^+ \rightarrow N^-$ to the painted cut problem is found, stop. The algorithm terminates and $x$ and $Q$ are solutions to the respective problems. Otherwise, go to step 3.

Step 2: Let $P: N^+ \rightarrow N^-$ be a solution to the painted path problem.
Let $a = \min[c^+(j) - x(j)]p(j)$ for $j \in P^+$ and

$b = \min[x(j) - c^-(j)]p(j)$ for $j \in P^-$. 

Let $\alpha = \min\{a, b\}$

Define $\alpha(j) = \frac{\alpha}{p(j)}$ for $j \in P$.

It is important to mention that $\alpha > 0$ from which it follows that $\alpha(j) > 0$

for every $j \in A$.

Define the new flow $x'(j)$ by

$$x'(j) = \begin{cases} x(j) + \alpha(j) & \text{if } j \in P^+ \\ x(j) - \alpha(j) & \text{if } j \in P^- \\ x(j) & \text{if } j \notin P. \end{cases}$$

Step 3: Repeat step 0 with $x'$.

3.4. Justification of the Algorithm

Claim 1: $x'$ is feasible with respect to the capacity intervals.

Proof: $\alpha(j) \leq c^+(j) - x(j)$ for $j \in P^+$

and $\alpha(j) \leq x(j) - c^-(j)$ for $j \in P^-$

For $j \notin P$, $x'(j) = x(j) = c^-(j) \leq x'(j) \leq c^+(j)$

For $j \in P^+$, $c^-(j) \leq x(j) < x'(j) = x(j) + \alpha(j) \leq c^+(j)$,

For $j \in P^-$, $x(j) > x'(j) = x(j) - \alpha(j) \geq c^-(j)$.

Therefore, $c^-(j) \leq x'(j) \leq c^+(j)$ for all $j \in P$. 
Claim 2: \( x' \) is conserved at all nodes outside \( N^+ \) or \( N^- \).

Proof:

For \( i \notin N^+ \cup N^- \),

\[
\tilde{y}_{x'}(i) = \sum_{j \in A} e(i,j) x'(j)p(j) = \sum_{j \in P} e(i,j) p(j) x'(j)
\]

\[
+ \sum_{j \in P^-} e(i,j) x'(j)p(j) + \sum_{j \notin P} e(i,j) p(j) x'(j)
\]

\[
\tilde{y}_{x'}(i) = \sum_{j \in P} e(i,j) x(j)p(j) + \sum_{j \notin P} e(i,j) p(j) x(j)
\]

\[
+ \sum_{j \in P^-} e(i,j) x(j)p(j) - \sum_{j \notin P} e(i,j) p(j) x(j)
\]

\[
\tilde{y}_{x'}(i) = \sum_{j \in A} e(i,j) x(j)p(j) + \alpha \left[ \sum_{j \in P} e(i,j) - \sum_{j \notin P} e(i,j) \right]
\]

\[
= 0 + \alpha \cdot 0 = 0
\]

Claim 3: The weighted flow from \( N^+ \) to \( N^- \) increases.

\[
\tilde{y}_{x'}(N^+)^+ = \sum_{i \in N^+} \sum_{j \in A} e(i,j) x'(j)p(j) = \sum_{i \in N^+} \sum_{j \in P} e(i,j) x(j)p(j)
\]

\[
+ \sum_{i \in N^+} \sum_{j \notin P} e(i,j) p(j) x(j)
\]

\[
+ \sum_{i \in N^+} \sum_{j \in P^-} e(i,j) x(j)p(j)
\]

\[
+ \sum_{i \in N^+} \sum_{j \notin P} e(i,j) p(j) x(j)
\]

\[
- \sum_{i \in N^+} \sum_{j \in P} e(i,j) p(j) x(j)
\]

\[
+ \sum_{i \in N^+} \sum_{j \notin P} e(i,j) p(j) x(j)
\]

\[
+ \sum_{i \in N^+} \sum_{j \in P^-} e(i,j) x(j)p(j)
\]

\[
+ \sum_{i \in N^+} \sum_{j \notin P} e(i,j) p(j) x(j)
\]

\[
- \sum_{i \in N^+} \sum_{j \in P} e(i,j) p(j) x(j)
\]

\[
+ \sum_{i \in N^+} \sum_{j \notin P} e(i,j) p(j) x(j)
\]

\[
- \sum_{i \in N^+} \sum_{j \in P^-} e(i,j) x(j)p(j)
\]

\[
+ \sum_{i \in N^+} \sum_{j \notin P} e(i,j) p(j) x(j)
\]

\[
- \sum_{i \in N^+} \sum_{j \in P} e(i,j) p(j) x(j)
\]

\[
+ \sum_{i \in N^+} \sum_{j \notin P} e(i,j) p(j) x(j)
\]
which implies

\[ \tilde{y}_x'(N^+) = \tilde{y}_x(N^+) + \alpha \sum_{i \in N^+} \left( \sum_{j \in P^+} e(i,j) - \sum_{j \in P^-} e(i,j) \right) \]

\[ \tilde{y}_x'(N^+) = \tilde{y}_x(N^+) + 2 \alpha n^+ > \tilde{y}_x(N^+) \quad \text{where } n^+ = |N^+|. \]

Claim 4: The weighted flow is optimal if a cut is found.

Proof: If a solution \( Q: N^+ \to N^- \) to the painted cut problem is obtained, it satisfies

\[ c^+(j) = x(j) \quad \text{for all } j \in Q^+, \]

and \( c^-(j) = x(j) \quad \text{for all } j \in Q^- \).

Consequently,

\[ \tilde{y}(N^+) = \text{weighted flux of } x \text{ across the cut } Q = \]

\[ = \sum_{j \in Q^+} x(j)p(j) - \sum_{j \in Q^-} x(j)p(j) \]

\[ = \sum_{j \in Q^+} x(j)c^+(j) - \sum_{j \in Q^-} x(j)p(j) = c^+(Q). \]

So, the weighted divergence of \( x \) from \( N^+ = C^+(Q) \). Thus, it must be optimal since \( C^+(Q) \) is an upper bound that is attained.

Step 1 of the algorithm is described in Figure 3.1.

\( Q = [S, N - S] \) is the cut found in step 1.
Figure 3.1. The flux values in the arcs of a cut

A question that also needs to be answered is: Is the algorithm finite? The answer in general is no. However, the algorithm is finite if arc discrimination is used. When using the painted network algorithm, green paths are given a choice preference among all the arcs in step 2. We look for green paths, if one is found and the flow is augmented, some green arcs will change to black or white. (At least the one yielding the minimum in the calculation of $\alpha$.) Whereas all the arcs that were already white or black remain so. Therefore, a stage must come after finitely many interactions when to make further progress toward a flow-augmenting path (if the algorithm has not yet terminated), one must resort to white or black arcs. Now, one has arrived at a situation $S \supset N^+$ and such that there are no green arcs in the cut $Q = [S, N-S]$.
The value of the current weighted flow \( x \) is

\[
\sum_{j \in \mathcal{Q}^+} x(j)p(j) - \sum_{j \in \mathcal{Q}^-} x(j)p(j) = \sum_{j \in \mathcal{Q}} c^+(j)p(j) - \sum_{j \in \mathcal{Q}} c^-(j)p(j) - \sum_{j \in \mathcal{Q}} c^+(j)p(j) - \sum_{j \in \mathcal{Q}} c^-(j)p(j) - \sum_{j \in \mathcal{Q}} c^+(j)p(j) - \sum_{j \in \mathcal{Q}} c^-(j)p(j)
\]

So, the value of the weighted flow is of the form

\[
\sum_{j \in \mathcal{Q}} c^+(j)p(j) + \sum_{j \in \mathcal{Q}} c^-(j)p(j) - \sum_{j \in \mathcal{Q}} c^+(j)p(j) - \sum_{j \in \mathcal{Q}} c^-(j)p(j)
\]

There are only finitely many sums of such a special form; since there are only finitely many arcs. Thus, it follows that there are only finitely many distinct values that the weighted flux from \( N^+ \) to \( N^- \) can possibly take on when there are no more green arcs that belong to a path \( P \) from \( N^+ \) to \( N^- \). No one of the values of the weighted flux from \( N^+ \) to \( N^- \) can ever be repeated since the flux is increased at each iteration. Therefore, step 1 of the algorithm must be reached sooner or later yielding a pair of solutions \((x, Q)\) to the max weighted flow and the min weighted cut problems, respectively. This completes the proof of the max weighted flow-min weighted cut theorem.

3.5. Example

The following is an illustration of how the maximum weighted flow algorithm can be implemented to find a solution to the maximum weighted flow problem. We start the algorithm with \( x = 0 \) which is a legal flow and use arc discrimination at step 2 of every iteration of the algorithm to insure termination.
\[ N^+ = \{A\} \text{ and } N^- = \{G\}. \]

**Iteration 0**

The arcs of the network are assigned the following order:

AB (1), BE (2), EG (3), AC (4), CF (5), FG (6), BC (7),
BD (8), CD (9), DE (10), DF (11), and EF (12).
Iteration 1

The path $P$: $A \rightarrow B \rightarrow E \rightarrow G$ is selected

$\alpha = \min\{2, \frac{3}{4}, \frac{6}{7}\} = \frac{3}{4}$

Thus $\alpha(1) = \frac{3}{2}$, $\alpha(2) = 3$, $\alpha(3) = \frac{21}{4}$

$x(1), x(2) = -3$, $x(3) = \frac{21}{4}$

and $x(j)$ remains unchanged for every arc $j$ not in $P$. 
Iteration 2

The path $P: A \rightarrow C \rightarrow F \rightarrow G$ is selected

$\alpha = \left\{ \frac{7}{3}, 2, 2 \right\} = 2$

It follows that $\alpha(4) = 6, \alpha(5) = 2, \alpha(6) = 4$

$x(4) = 6, x(5) = -2, x(6) = 4$

and $x(j)$ remains the same for each $j$ not in $P$. 
The path P: A → C → D → E → G is selected

\[ \alpha = \min\left(\frac{1}{3}, \frac{4}{3}, 3, \frac{3}{28}\right) = \frac{3}{28} \]

Thus \( \alpha(3) = \frac{3}{4}, \alpha(4) = \frac{9}{28}, \alpha(9) = \frac{9}{56}, \alpha(10) = \frac{3}{28} \)

\[ x(3) = 6, x(4) = \frac{177}{28}, x(9) = \frac{9}{56}, x(10) = \frac{3}{28} \]

and \( x(j) \) remains the same for each arc \( j \) not in P.
There exists a cut $Q = [N - G, G]$ so we stop. The weighted flux across the cut $Q$ equals
\[ \sum_{j \in Q^+} x(j)p(j) - \sum_{j \in Q^-} x(j)p(j) = c^+(3) + c^+(6) = \frac{6}{7} + 2 = \tilde{y}(A) \]

Hence, the maximum weighted flow from $A$ to $G$ is $\frac{20}{7}$.

The flow $x$ of the network at iteration 4 solves the maximum weighted flow problem whereas the cut $Q$ obtained is a solution to the minimum weighted cut problem.
4. THE FEASIBLE DISTRIBUTION PROBLEM

4.1. Statement of the Problem

Given capacity intervals $C(j) = [c^-(j), c^+(j)]$ for all arcs $j$ and supply values $b(i)$ for all nodes $i$, find a flow $x$ such that

$$c^-(j) \leq x(j) \leq c^+(j) \quad \text{for all } j \in A,$$

$$y(i) = b(i) \quad \text{for all } i \in N.$$

The vector $b$ is called the supply function. If $b = 0$, one has the feasible circulation problem. A flow satisfying the supply and capacity constraints is called a solution to the feasible distribution problem. For example, finding a flow that satisfies the constraints of the maximum weighted flow problem corresponds to solving a feasible distribution problem with $b(i) = 0$ for all $i \notin N^+ \cup N^-$ and $b(i) \in \mathbb{R}$ for all $i \in N^+ \cup N^-$. As mentioned in Chapter 3, the max weighted flow problem can be formulated as a linear programming problem to which the simplex method may be applied to obtain the optimal solution. In the context of this formulation, the feasible distribution problem consists precisely of solving phase 1 of the above linear programming problem.

4.2. The Feasible Distribution Theorem

The feasible distribution problem has a solution if and only if $b(N) = 0$ and $b(S) \leq C^+(Q)$ for all cuts $Q = [S, N - S]$. Note that for
S being an arbitrary node set, \( b(S) = \sum_{i \in S} b(i) \) (it represents the net supply in \( S \)).

**Proof:** We first prove the necessity of the condition.

\( b(N) = 0 \) is always satisfied by the total weighted divergence principle. Also, \( b(S) = \sum_{i \in S} \tilde{y}(i) = \) weighted divergence of \( x \) from \( S = \) weighted flux of \( x \) across \( Q \) by the fundamental weighted divergence principle. But

\[
\sum_{i \in S} \tilde{y}(i) = \sum_{i \in S} \sum_{j \in A} e(i,j)x(j)p(j) = \sum_{j \in Q^+} x(j)p(j) = \sum_{j \in Q^-} x(j)p(j).
\]

Using the capacity constraints, namely

\[
c^-(j) \leq x(j) \leq c^+(j) \quad \text{for all } j \in Q^+
\]

\[
-c^+(j) \leq -x(j) \leq -c^-(j) \quad \text{for all } j \in Q^-.
\]

We have

\[
\sum_{j \in Q^+} c^-(j)p(j) - \sum_{j \in Q^-} c^+(j)p(j) \leq \sum_{j \in Q^+} x(j)p(j) - \sum_{j \in Q^-} x(j)p(j)
\]

\[
\leq \sum_{j \in Q^+} c^+(j)p(j) - \sum_{j \in Q^-} c^-(j)p(j)
\]

which leads to \( b(S) \leq \sum_{j \in Q^+} c^+(j)p(j) - \sum_{j \in Q^-} c^-(j)p(j) = C^+(Q) \).

This completes the proof of the necessary condition.

**REMARK:** The inequality \( b(S) \geq c^-(Q) \) is also necessary since

\[
0 = b(N) = b(S) + b(N - S) \quad \text{which implies } b(S) = -b(N - S).
\]

At the same time, the cut \( Q' = [N - S, S] \) is the reverse of
the cut \( Q \) which implies \( C^+(Q') = -C^-(Q) \) from which it follows 
\[ b(N - S) \leq C^+(Q') \text{ or } -b(S) \leq -C^-(Q), \text{ i.e., } b(S) \geq C^-(Q). \]
Again, the sufficiency is proved by an algorithm.

4.3. The Feasible Distribution Algorithm

4.3.1. Description of the algorithm

We start with the assumption \( b(N) = 0 \). Pick any flow \( x \) that is feasible with respect to the arc capacities. \( x(j) \) could be chosen to be any number in \([c^-(j), c^+(j)]\). Then, let \( N^+ = \{i \in N/b(i) > \bar{y}(i)\} \) and \( N^- = \{i \in N/b(i) < \bar{y}(i)\} \).

Step 0: If \( N^+ = \phi = N^- \), then \( x \) is a solution to the feasible distribution problem, and the algorithm terminates. If not, then both \( N^+ \) and \( N^- \) are nonempty (because \( \sum_{i \in N} [b(i) - \bar{y}(i)] = 0 \), due to \( b(N) = 0 = \bar{y}(N) \)) and go to step 1.

Step 1: The painted network algorithm is applied with the same painting of the arcs as in the maximum weighted flow algorithm:
- green if \( c^-(j) < x(j) < c^+(j) \)
- white if \( c^-(j) = x(j) < c^+(j) \)
- black if \( c^-(j) < x(j) = c^+(j) \)
- red if \( c^-(j) = x(j) = c^+(j) \)

Step 2: If the outcome is a cut \( Q = [S, N - S] \), then stop there is no solution to the feasible distribution problem.

Step 3: If the outcome is a path \( P: N^+ \rightarrow N^- \), then define the numbers \( a, b, c \) and \( d \) as follows:
\[ a = \min_j p(j)[c^+(j) - x(j)] \quad \text{for } j \in P^+ \]

\[ b = \min_j p(j)[x(j) - c^-(j)] \quad \text{for } j \in P^- \]

\[ c = \frac{b(i) - \tilde{y}(i)}{p(\text{initial})} \]

where initial is the first arc of the path \( P \) and \( i \) is the initial node of that arc in \( N^+ \).

\[ d = \frac{\tilde{y}(f) - b(f)}{p(\text{final})} \]

where final is the last arc of the path \( P \) and \( f \) is the terminal node of that arc in \( N^- \).

Let \( \alpha = \min\{a, b, c, d\} \), and \( \alpha(j) = \begin{cases} \frac{\alpha}{p(j)} & \text{if } \alpha = a \text{ or } \alpha = b \\ \frac{\alpha}{p(\text{initial})} & \text{if } \alpha = c \\ \frac{\alpha}{p(final)} & \text{if } \alpha = d \end{cases} \)

Then define the flow \( x' \) such that \( x'(j) = x(j) + \alpha(j)e_p \) (\( e_p \) is the incidence function of the path \( P \)), and go to step 0.

### 4.4. Justification of the Algorithm

Suppose step 2 is reached, that is, there is a cut \( Q: N^+ \cup N^- \)

which solves the painted cut problem and corresponds to the set \( S \)

(\( S \supseteq N^+ \) and \( S \cap N^- = \emptyset \)). Figure 4.1 illustrates such a cut.

Then \( C^+(Q) = \sum_{j \in Q} c^+(j)p(j) - \sum_{j \in Q} c^-(j)p(j) = \sum_{j \in Q} x(j)p(j) \)

\[ - \sum_{j \in Q} x(j)p(j) = \tilde{y}(S) \]

by the fundamental weighted divergence principle.

Also, \( C^+(Q) = b(S) - [b(S) - \tilde{y}(S)] = b(S) - [b(N^+) - \tilde{y}(N^+)] \).
$b(S) - \tilde{y}(S) = b(N^+) - \tilde{y}(N^+)$ follows from the fact that

$$b(S) - \tilde{y}(S) = b(S - N^+) + b(N^+) - \tilde{y}(S - N^+) - \tilde{y}(N^+) = b(N^+) - \tilde{y}(N^+).$$

Therefore, $C^+(Q) = b(S) - [b(N^+) - \tilde{y}(N^+)] < b(S)$ which violates the necessary condition. On the other hand, if $P: N^+ + N^-$ solves the painted path problem in step 3, the numbers whose minimum define $\alpha$ are all positive and $\alpha(j)$ is positive for every $j \in A$ as well since the weights (or probabilities) are all positive. Furthermore, the numbers $c$ and $d$ are finite and so are $\alpha$ and $\alpha(j)$ for every $j \in A$.

The flow $x'$ defined by $x'(j) = x(j) + \alpha(j)e_P$ ($e_P$ is the incidence function of the path $P$) satisfies the capacity constraints as well. For its weighted divergence $\tilde{y}'$, one has $b(i) - \tilde{y}'(i) = b(i) - \tilde{y}(i)$.
for all nodes, except that

\[ 0 \leq b(i) - \tilde{y}'(i) = b(i) - \tilde{y}(i) - \alpha \quad \text{where } i \text{ is the initial node of } P \]

\[ 0 \geq b(i) - \tilde{y}'(i) = b(i) - \tilde{y}(i) + \alpha \quad \text{where } i \text{ is the final node of } P \]

This shows, indeed, that \( x' \) is an improvement over \( x \), since \( b - \tilde{y}' \) is nearer to 0 than \( b - \tilde{y} \) was.

As in the case of the max weighted flow algorithm, the algorithm must terminate after finitely many iterations if arc discrimination is used.

There is a second procedure called the weighted flow rectification algorithm which also solves the feasible distribution problem. A detailed description and justification of that algorithm is given in Appendix A.

4.5. Examples

4.5.1. A feasible distribution problem which has no solution

The feasible distribution algorithm is applied to a network which has no solution to the feasible distribution problem. We start with a feasible flow that is \( x = 0 \). The outcome of the algorithm will be a cut.
Iteration 1
\[ \alpha = \min \{ \frac{1}{2}, \frac{1}{5}, 2, 5 \} = \frac{1}{5} \]

\[ \alpha(1) = \frac{1}{5} = \frac{2}{5}, \quad \alpha(2) = \frac{1}{5} = 1 \]

\[ x(1) = -\frac{2}{5}, \quad x(2) = -1 \]
Iteration 2

\[
\begin{array}{c}
\text{Path} \\
A \rightarrow D \rightarrow E
\end{array}
\]

\[
\alpha = \min \left\{ \frac{1}{3}, 1, \frac{12}{5}, \frac{4}{5} \right\} = \frac{1}{3}
\]

\[
\alpha(1) = 1, \quad \alpha(2) = \frac{1}{3}, \quad x(1) = 1, \quad x(2) = \frac{1}{3}.
\]
Thus, there is no solution to the feasible distribution problem for this particular network.
4.5.2. A feasible distribution problem which has a solution

In the following network, the numbers inside the nodes are the supplies. The numbers next to the arcs are the weights.

By simple inspection of the network, we found that the flow $x$ given by $x(AB) = -4$, $x(DA) = -8$, and $x(BE) = x(CA) = x(DC) = x(CE) = x(BC) = x(ED) = 0$, is a solution to the feasible distribution problem.

Now, we assume that no solution is known or can be obtained by simple inspection of the network and apply the feasible distribution algorithm to find it. We start with $x \equiv 0$. 
Iteration 1

The network is painted.

\[ a(1) = \min\{2, 1, 2, 2\} = 1 \]

\[ a(2) = 1, \ x(1) = 1, \ \text{and} \ x(2) = -1 \]

Update the flow.
Iteration 2

The network is painted.

\[ a = \min\{\frac{1}{6}, \frac{1}{5}, 6, 5\} = \frac{1}{6} \]

\[ \alpha(1) = 1, \ \alpha(2) = \frac{1}{3}, \ x(1) = 1, \ x(2) = \frac{1}{3}. \] Update the flow.
Iteration 3

The network is painted.

\[ a = \min\{\frac{5}{2}, 2, \frac{10}{8}, \frac{20}{6}\} = 2 \]

\[ \alpha(1) = 4, \ \alpha(2) = 8, \ x(1) = -4, \ x(2) = -8 \]

Update the flow.
Iteration 4

The network is painted.

A solution is obtained.

\[ x(AB) = -4, \ x(DA) = -8, \ x(BC) = 1, \ x(CA) = 0 \]

\[ x(DC) = -1, \ x(BE) = 1, \ x(DE) = \frac{5}{6}, \ x(CE) = 0. \]

This solution is different than the one guessed initially by simple inspection. We draw the conclusion that there is not a unique solution to the feasible distribution problem.
5. THE MAXIMUM WEIGHTED TENSION PROBLEM
AND ITS DUAL PROBLEM

5.1. The Maximum Weighted Tension Problem

5.1.1. Assumptions and definitions

Let $G$ be a network and $u$ be a potential function defined on the nodes. For each node $i \in N$, we assume there exists a positive weight $p(i)$ attached to it. For instance, the weight $p(i)$ could represent the probability of reaching the node $i$ or leaving it. Similarly, as in Chapter 3, there are practically no restrictions on the weights $p(i)$ except that they are assumed to belong to the interval $(0, \infty)$. Let $v$ be a tension defined on the arcs by

$$v(j) = u(i')p(i') - u(i)p(i) \quad \text{where } j \sim (i, i')$$

It is assumed that the tension $v$ satisfies some feasibility constraints, namely, $v(j) \in [d^-(j), d^+(j)]$ for all $j \in A$ where $d^-(j)$ and $d^+(j)$ are prescribed for all $j \in A$. $D(j) = [d^-(j), d^+(j)]$ is called a span interval.

For an arbitrary tension $v$ and path $P$, the weighted spread of $v$ relative to $P$ is defined by

$$[\text{weighted spread of } v \text{ relative to } P] = \sum_{j \in P^+} v(j) - \sum_{j \in P^-} v(j)$$
5.1.2. Description of the problem

It follows directly from the definition of the weighted spread of $v$ relative to $P$ that

$$[\text{weighted spread of } v \text{ relative to } P] = u(i')p(i') - u(i)p(i)$$

if

$$v = \Delta u \text{ and } P: i \to i'.$$

This formula, called the integration rule, holds simply due to the fact that the terms $v(j)$ present in the sum of the definition telescope. As a consequence of the integration rule, the weighted spread of $v$ relative to $P$ depends only on the endpoints of $P$ and not on the nodes that come in between. We consider now a problem for potentials that we call the maximum weighted tension problem. It is analogous to the max weighted flow problem and likewise concerns two nonempty disjoint node sets $N^+$ and $N^-$ of a network $G$. We assume that $u$ is a potential such that $u(i)p(i)$ is constant on $N^+$ and likewise for $u(i')p(i')$ on $N^-$. The value $u(i')p(i') - u(i)p(i)$ is then the same for any $i \in N^+$ and $i' \in N^-$, and is called the weighted spread of $u$ from $N^+$ to $N^-$. We note that due to the integration rule, this value equals also the weighted spread of the differential $v = \Delta u$ relative to any path $P: N^+ \to N^-$.  

5.1.3. Statement of the problem

Maximize the weighted spread of $u$ from $N^+$ to $N^-$ over all potentials $u$ such that $u(i)p(i)$ is constant on $N^+$ and $N^-$, and such that $v = \Delta u$ is feasible with respect to spans.
5.2. The Maximum Weighted Tension Algorithm

We assume there is a potential $u$ which satisfies the constraints of the max weighted tension problem. The procedure for finding such a potential will be discussed in the next chapter.

**Algorithm**:

**Step 0:** Paint the arcs $j$ of the network in terms of $v$ as follows:
- **Red** if $d^-(j) < v(j) < d^+(j)$
- **Black** if $d^-(j) = v(j) < d^+(j)$
- **White** if $d^-(j) < v(j) = d'^-(j)$
- **Green** if $d^-(j) = v(j) = d'^+(j)$

and apply the painted network algorithm.

**Step 1:** If a path $P: N^+ \to N^-$ is obtained, stop; $u$ solves the max weighted tension problem.

**Step 2:** If a cut $Q: N^+ \downarrow N^-$ is obtained, calculate
\[
\alpha = \min \left\{ d^+(j) - v(j) \text{ for } j \in Q^+ \\
 v(j) - d^-(j) \text{ for } j \in Q^-. \right\}
\]
If $\alpha = \infty$, stop. The value of the weighted spread of $u$ from $N^+$ to $N^-$ is $+\infty$. Otherwise define the potential $u'$ as follows:
\[
u'(i) = u(i) + \frac{\alpha}{p(i)} e_{N^-S} \text{ for all } i \in N
\]
where $Q = [S, N-S]$ and repeat step 0 with the potential $u'$.
5.3. Justification of the Algorithm

**Claim 1:** The differential $v'$ of the potential $u'$ is feasible with respect to spans.

**Proof:** Let $j \sim (i, i') \in Q$. Then,

$$v'(j) = u'(i')p(i') - u'(i)p(i)$$

$$= u(i')p(i') + \frac{\alpha}{p(i')} p(i') - u(i)p(i)$$

$$v'(j) = v(j) + \alpha$$

which shows that the differential $v'$ of $u'$ is feasible with respect to spans as well because of the way $\alpha$ is designed.

**Claim 2:** The weighted spread of $u'$ from $N^+$ to $N^-$ increases.

**Proof:** For $i \in N^+$ and $i' \in N^-$,

$$v'(j) = u'(i')p(i') - u'(i)p(i)$$

$$= v(j) + \alpha > v(j) \text{ since } \alpha > 0.$$
j must be either red or white. Thus, if j gives the minimum value of α, then at step 0 of the next iteration, j will turn black or green. Hence, at every iteration of the algorithm, either step 1 is reached or the number of arcs of Q is reduced by at least one arc. Consequently, after a finite number of iterations one must obtain a path P from $N^+$ to $N^-$ and, hence, a solution $u$ to the max weighted tension problem.

5.4. Example

The following example illustrates how the max weighted tension algorithm is implemented on a given network and with a potential $u$ satisfying the constraints of the max weighted tension problem.

$p(A) = \frac{1}{2}$, $p(B) = \frac{1}{3}$, $p(C) = \frac{1}{2}$, $p(D) = \frac{1}{4}$.

$N^+ = \{A\}$, $N^- = \{D\}$. The algorithm is started with the following potential: $u(A) = 1$, $u(B) = 3$, $u(C) = 2$, and $u(D) = 4$.

**Iteration 0**
Iteration 1

Paint the network

There exists a cut $Q = [S, N - S]$ where $S = \{A\}$ and $N - S = \{B,C,D\}$

$$\alpha = \min \left\{ \frac{5}{2}, \frac{1}{2} \right\} = \frac{1}{2}$$

$$\alpha(B) = \frac{1}{2} \cdot \frac{1}{3} = \frac{3}{2}, \quad \alpha(C) = \frac{2}{1} = 1 \quad \text{and} \quad \alpha(D) = 2.$$
Update the potentials

\[ u(B) = 3 + \frac{3}{2} = \frac{9}{2}, \quad u(C) = 2 + 1 = 3, \quad u(D) = 4 + 2 = 6; \]

\[ u(A) \text{ is unchanged}. \]

Iteration 2

Paint the network

There exists a cut \( Q = [S, N - S] \) where \( S = \{A, C\} \).

The arcs in the cut are AB and CD.
\( \alpha = \min \{2, 1\} = 1. \) \( \alpha(B) = 3, \alpha(D) = 4. \)

Update the potentials: \( u(B) = \frac{9}{2} + 3 = \frac{15}{2}, \ u(D) = 6 + 4 = 10; \)
\( u(A) \) and \( u(C) \) are unchanged.

\[ \begin{align*}
A & \quad 1 \quad \text{white} \\
B & \quad 2 \quad \text{red} \\
C & \quad 3 \quad \text{black} \\
D & \quad 10 \quad \text{black}
\end{align*} \]

\textbf{Iteration 3}

Paint the network

\[ \begin{align*}
A & \quad 1 \quad \text{white} \\
B & \quad 2 \quad \text{red} \\
C & \quad 3 \quad \text{white} \\
D & \quad 10 \quad \text{white}
\end{align*} \]

There is a path \( P \) from \( A \) to \( D \).

The value of the weighted spread of \( u \) from \( A \) to \( D \) is \( \frac{10}{4} - \frac{1}{2} = 2 \).

The solution of the problem is the potential \( u \) such that \( u(A) = 1, \)
\( u(B) = \frac{15}{2}, u(C) = 3, \) and \( u(D) = 10. \)

Also, the weighted spread equals \( d^+(AC) + d^+(CD). \)
REMARK: When the algorithm terminates and a solution $u$ to the max weighted tension problem is found, we note that the weighted spread of $u$ from $N^+$ to $N^-$ is equal to 2 which is also the value of $\sum_{j \in P^+} d^+(j) - \sum_{j \in P^-} d^-(j)$. Indeed, the path $P: N^+ \rightarrow N^-$ obtained at step 1 in the last iteration of the algorithm is a solution to the minimum path problem. The minimum path problem is a dual problem to the maximum weighted tension problem.

5.5. The Minimum Path Problem

5.5.1. Assumptions and definitions

Given the spans $D(j) = [d^-(j), d^+(j)]$ for the tension $v$, and a path $P: N^+ \rightarrow N^-$ where $N^+$ and $N^-$ are two nonempty disjoint node sets of a network $G$, $d^+(P)$ is defined by

$$d^+(P) = \sum_{j \in P^+} d^+(j) - \sum_{j \in P^-} d^-(j).$$

5.5.2. Statement of the problem

Minimize $d^+(P)$ over all paths $P: N^+ \rightarrow N^-$.  

5.5.3. Max weighted tension-min path theorem (Minty 1960)

Suppose there is at least one potential satisfying the constraints of the max weighted tension problem. Then

$$[\text{sup in max weighted tension problem}] = [\text{min in min path problem}].$$
Both Minty (1960) and Rockafellar (1984) contain a proof of the theorem.

5.6. The Minimum Path Problem

There are many algorithms which solve the minimum path problem. Among them, we mention Dijkstra's algorithm. Dreyfus (1969) contains an excellent analysis of the traditional methods for optimal routing problems. Golden (1976) contains numerical comparisons of some optimal routing problems.
6. THE WEIGHTED TENSION RECTIFICATION ALGORITHM

6.1. The Feasible Differential Problem

6.1.1. Description of the problem

The maximum weighted tension algorithm makes use of an initial potential satisfying the constraints described in 5.1.3. In fact, this leads to the question of how such a potential might be constructed in general, when none seems evident.

6.1.2. Statement of the problem

In a network \( G \) with span intervals \( D(j) = [d^-(j), d^+(j)] \), determine a potential \( u \) whose differential \( v \) satisfies \( v(j) \in D(j) \) for all \( j \in A \).

6.2. The Weighted Tension Rectification Algorithm

The algorithm starts with an arbitrary potential \( u \) and its differential \( v \). Let

\[
A^+ = \{ j \in A / v(j) < d^-(j) \} \quad \text{and} \quad A^- = \{ j \in A / v(j) > d^+(j) \}
\]

Step 0: If \( A^+ = A^- = \emptyset \), then \( u \) is a solution to the feasible differential problem, and the algorithm terminates. Otherwise, let \( \bar{j} \) be an arc in \( A^+ \cup A^- \).

Step 1: Apply Minty's algorithm to \( \bar{j} \) and the following painting of the arcs:
red if \( d^-(j) < v(j) < d^+(j) \)
black if \( v(j) < d^-(j) < d^+(j) \)
white if \( v(j) > d^+(j) > d^-(j) \)
green if \( d^-(j) = v(j) = d^+(j) \)

REMARK: The arcs in \( A^+ \) are all black, whereas those in \( A^- \) are white; in particular, \( j \) is either black or white.

Step 2: If the outcome is a circuit \( P \) containing \( j \) and compatible with the above painting, then stop there is no solution to the feasible differential problem.

Step 3: If a compatible cut \( Q \) containing \( j \) is the outcome, define

- \( a \), \( b \), \( c \), and \( d \) as follows:

  \[
  a = \min \{ d^+(j) - v(j) \} \quad \text{for } j \in Q^+ \\
  b = \min \{ v(j) - d^-(j) \} \quad \text{for } j \in Q^- \\
  c = \min \{ v(j) - d^+(j) \} \quad \text{for } j \in A^- \\
  d = \min \{ d^-(j) - v(j) \} \quad \text{for } j \in A^+ 
  \]

Let \( \alpha = \min \{a, b, c, d\} \) and \( \alpha(i) = \frac{\alpha}{p(i)} \) for \( i \in N \). Then, define the potential \( u' \) by \( u'(i) = u(i) + \alpha(i)e_{N-S} \) where \( Q = [S, N - S] \) and go to step 0.
6.3. Justification of the Algorithm

**Claim 1:** If step 2 is reached, then there is no solution to the feasible differential problem.

**Proof:** We assume Minty's algorithm results in a circuit $P$ containing the arc $j$ and compatible with the painting. Then, one must have

\[ v(j) > d^+(j) \text{ for } j \in P^+ \text{ and } v(j) \leq d^-(j) \text{ for } j \in P^- \]

with at least one strict inequality since $P$ contains $\bar{j}$.

It follows

\[ \sum_{j \in P^+} v(j) - \sum_{j \in P^-} v(j) > \sum_{j \in P^+} d^+(j) - \sum_{j \in P^-} d^-(j) \]

which gives

\[ d^+(P) = \sum_{j \in P^+} d^+(j) - \sum_{j \in P^-} d^-(j) < 0. \]

\[ \sum_{j \in P^+} v(j) - \sum_{j \in P^-} v(j) = 0 \text{ since } P \text{ is a circuit.} \]

Therefore, the necessary condition of the feasible differential problem is violated and, hence, there is no solution.

**Claim 2:** If a compatible cut $Q$ containing $\bar{j}$ is the outcome of Minty's algorithm, then there is a solution to the feasible differentiable problem.
Proof: With a cut $Q$ as the outcome in step 2, one has that all the numbers $a, b, c,$ and $d$ must be positive. Furthermore, either $c$ or $d$ must be finite since the cut $Q$ contains the arc $\bar{j}$ which itself belongs to either $A^+$ or $A^-$. It follows that $\alpha$ is finite and then so is $\alpha(i)$ for all $i \in N$. The differential $v'$ of the potential $u'$ satisfies the following:

For $j \in Q^+$, $v'(j) = u'(i')p(i') - u'(i)p(i)$

$$= u(i')p(i') - u(i)p(i) + \alpha$$

$$v'(j) = v(j) + \alpha \leq d^+(j)$$

and $d^-(j) - v'(j) = d^-(j) - v(j) - \alpha$

For $j \in Q^-$, $v'(j) = v(j) - \alpha \geq d^-(j)$

and $v'(j) - d^+(j) = v(j) - d^+(j) - \alpha$

whereas $v'(j) = v(j)$ for all $j \notin Q$.

The only arcs which violate the feasibility with respect to spans are those in $A^+ \cup A^-$. Again, with a similar argument as for the weighted flow rectification algorithm, the algorithm must terminate by reaching a solution if the same arc $\bar{j}$ is always picked at step 0, as long as it is in $A^+ \cup A^-$. 


6.4. Example

Here, we apply the weighted tension rectification algorithm to the same network treated in Chapter 5. We know that the potential \( u(A) = 1, u(B) = 3, u(C) = 2 \) and \( u(D) = 4 \) is a solution to the feasible differential problem.

**Iteration 1**

We start with any potential \( u \) since \( N^+ = \{A\} \) and \( N^- = \{D\} \) (singleton node sets). We recall that \( p(A) = \frac{1}{2}, p(B) = \frac{1}{3}, p(C) = \frac{1}{2}, \) and \( p(D) = \frac{1}{4} \). We start with the potential \( u \) such that \( u(A) = 1, u(B) = \frac{3}{2}, u(C) = \frac{4}{3}, \) and \( u(D) = \frac{5}{3} \).
Iteration 2

\[ A^+ = \{BD, CD\}, \quad A^- = \emptyset. \quad \text{We note that the arc AB left } A^+ \cup A^- \]

Select \( j = CD \) and paint the network.

A cut \( Q = [N, N - S] \) where \( N = \{A, C\} \) \quad \( j = CD \in Q^+ \)

\[
\begin{align*}
    a &= \min\{3, \ 1 + \frac{1}{4}, \ 1 - \frac{1}{6}\} = \frac{5}{6} \\
    u'(B) &= \frac{3}{2} + \frac{1}{3} = \frac{11}{6} = \frac{3}{2} + \frac{1}{4} = \frac{7}{4} \\
    b &= \min\{\frac{1}{6}\} = \frac{1}{6}, \quad c = \infty \\
    u'(D) &= \frac{5}{3} + \frac{1}{4} = \frac{7}{3} + \frac{1}{3} = \frac{1}{2} \\
    d &= \min\{\frac{1}{12}, \frac{1}{4}\} = \frac{1}{12} \\
    \alpha &= \min\{\frac{5}{6}, \frac{1}{6}, \frac{1}{12}, \infty\} = \frac{1}{12} \quad \text{Update the potentials.}
\end{align*}
\]
Iteration 3

\[ A^+ = \{BD, CD\}, A^- = \emptyset. \text{ Select } \bar{j} = CD \text{ again and paint the network.} \]

\[
\begin{array}{c}
\text{A cut } Q = [N, N - S] \text{ where } N = \{A, B, C\} \\
\bar{j} = CD \in Q^+ \\
a = \min\{3 - \frac{1}{12}, 1 - \frac{1}{6}, 3 + \frac{1}{12}, 1 + \frac{1}{6}\}, \quad b = \min\{\frac{1}{6}, \frac{1}{12}\} = \frac{1}{2} \\
c = \infty, \quad d = \min\{\frac{1}{12}, \frac{1}{6}\} = \frac{1}{12} \\
\alpha = \frac{1}{12}. \quad u'(D) = 2 + \frac{1}{2} \cdot \frac{1}{4} = 2 + \frac{1}{3} = \frac{7}{3}. \text{ Update the potentials.}
\end{array}
\]
Iteration 4

\[ A^+ = \{CD\}, \ A^- = \emptyset. \] Select \( \bar{j} = CD \) (no choice at this time) and paint the network.

A cut \( Q = [N, N - S] \) where \( N = \{A, B, C\} \)

\( j = CD \in Q^+ \)

\( a = \min\{3 - \frac{1}{12}, 1 - \frac{1}{6}, 1 + \frac{1}{12}, 3\} \)

\( a = \min\{\frac{10}{12}, \frac{11}{12}, \infty, \frac{1}{12}\} = \frac{1}{12} \)

\( b = \min\{\frac{1}{6}, \frac{1}{12}\} = \frac{1}{12}, \ c = \infty \)

\( u'(D) = \frac{7}{3} + \frac{1}{4} = \frac{8}{3} \)

Update the potentials.
Iteration 5

\[ A^+ = \{CD\}, \quad A^- = \emptyset. \] Select \( j = CD \) again and paint the network.

\[ \text{Cut } Q = [N, N - S] \text{ where } N = \{A,B,C\} \]

\[ a = \min\{3 - \frac{1}{12}, 1 - \frac{1}{6}, 1 + \frac{1}{12}, 3\} \]

\[ b = \min\{\frac{1}{6}, \frac{1}{12}\} = \frac{1}{12}, \quad c = \infty \]

\[ d = \frac{1}{12} \quad \Rightarrow \alpha = \frac{1}{12} \]

Update the potentials.
**Iteration 6**

A solution is obtained.

The potential u: \( u(A) = 1, u(B) = \frac{7}{4}, u(C) = \frac{4}{3} \) and \( u(D) = 3 \) satisfies the constraints of the feasible differential problem. We note this solution is different than the one we used in Chapter 5 to start the max weighted tension algorithm. Similarly, like the feasible distribution problem, the feasible differential problem doesn't have a unique solution if any solution exists.
7. THE WEIGHTED LINEAR OPTIMAL DISTRIBUTION PROBLEM
AND APPLICATIONS

7.1. The Weighted Linear Optimal Distribution Problem

7.1.1. Description of the problem

Let G be a connected network where each arc has a capacity interval \([c^-(j), c^+(j)]\) and each node \(i\) has a supply \(b(i)\), where \(b(N) = 0\). We assume that the cost of the flux \(x(j) \in [c^-(j), c^+(j)]\) is given by a linear expression \(d(j)x(j) + r(j)\), where \(d(j)\) and \(r(j)\) are constants associated with the arc \(j\). The cost of a solution \(x\) to the feasible distribution problem is then the sum of these linear expressions.

7.1.2. Statement of the problem

Minimize 
\[
\sum_{j \in A} [d(j)x(j) + r(j)] = d.x + \text{(constant)}
\]
over all flows \(x\) such that

\[
c^-(j) \leq x(j) \leq c^+(j) \quad \text{for all } j \in A
\]

\[
\sum_{j \in A} e(i, j)p(j)x(j) = b(i) \quad \text{for all } i \in N
\]

where \(p(j)\) is a constant associated with the arc \(j\).

Without loss of generality, all the constants \(r(j)\) are assumed to be zero. We will assume also that both \(c^+(j)\) and \(c^-(j)\) are finite for all \(j \in A\) for reasons of simplicity.
7.1.3. A duality theorem

The dual problem of the weighted linear optimal distribution problem is

$$\text{maximize } - \sum_{i} b(i)u(i) + \sum_{j} \min \left\{ c^+(j)[d(j) - v(j)], c^-(j)[d(j) - v(j)] \right\}$$

over all potentials $u$, where $v(j) = [u(i') - u(i)]p(j)$ where $j \sim (i, i')$.

Proof: Let

$$A_j = \left\{ (x(j), v(j) \in \mathbb{R}^2, \text{satisfying } c^-(j) \leq x(j) \leq c^+(j) \text{ and having} \right\}$$

$$v(j) \leq d(j) \text{ if } x(j) < c^+(j), \text{ and } v(j) \geq d(j) \text{ if } x(j) > c^-(j)$$

Lemma 1. For $x(j) \in C(j)$ and $v(j) \in \mathbb{R}$, one has

$$x(j)[d(j) - v(j)] \geq \min \left\{ c^+(j)[d(j) - v(j)], c^-(j)[d(j) - v(j)] \right\}$$

Equality holds in (1) if and only if $(x(j), v(j)) \in A_j$.

Sufficiency: Let $(x(j), v(j)) \in A_j$

Case 1. If $x(j) = c^+(j) = c^-(j)$, then equality holds.

Case 2. If $c^-(j) \leq x(j) \leq c^+(j)$, then $v(j) \leq d(j)$ which implies either $v(j) = d(j)$ or $x(j)$ must equal $c^-(j)$.

- If $v(j) = d(j)$, the equality is obvious.
- If $x(j) = c^-(j)$ ($d(j) > v(j)$) then

$$\min \left\{ c^+(j)[d(j) - v(j)], c^-(j)[d(j) - v(j)] \right\}$$

$$= c^-(j)[d(j) - v(j)]$$

which equals the left-hand side of inequality (1).
Case 3. If $c^-(j) < x(j) \leq c^+(j)$, then $v(j) \geq d(j)$ which implies either $v(j) = d(j)$ or $x(j)$ must equal $c^+(j)$. Therefore, if $v(j) = d(j)$, then the equality is evident. Otherwise, if $x(j) = c^+(j) \left( v(j) > d(j) \right)$, then

$$\min \left\{ c^+(j)[d(j) - v(j)], c^-(j)[d(j) - v(j)] \right\} = c^+(j)[d(j) - v(j)]$$

which equals the left-hand side of (1).

**Necessity:** We assume that

$$x(j)[d(j) - v(j)] = \min \left\{ c^+(j)[d(j) - v(j)], c^-(j)[d(j) - v(j)] \right\}$$

Thus, $x(j)[d(j) - v(j)] \leq c^+(j)[d(j) - v(j)]$, and

$$x(j)[d(j) - v(j)] \leq c^-(j)[d(j) - v(j)].$$

Equality holds for one inequality.

If $d(j) > v(j)$, then $x(j) = c^-(j)$ since $c^+(j) > c^-(j)$, and if $d(j) < v(j)$, then $x(j) = c^+(j)$.

So, $x(j) < c^+(j) \Rightarrow d(j) \geq v(j)$, and

$$x(j) > c^-(j) \Rightarrow d(j) \leq v(j).$$

This completes the proof of lemma 1.

**Lemma 2:** The inequality

$$\sum_{j \in A} x(j)d(j) \geq - \sum_{i \in N} b(i)u(i) + \sum_{j \in A} \min \left\{ c^+(j)[d(j) - v(j)], c^-(j)[d(j) - v(j)] \right\}$$

(2)
is valid for every feasible solution $x$ and every potential $u$ and its differential $v$, and equality holds if and only if $(x(j), v(j)) \in A_j$ for every $j \in A$.

For a given potential, we define the differential $v$ of $u$ by $v(j) = [u(i') - u(i)]p(j)$ where $j \sim (i, i') \in A$.

**Proof:** From lemma 1,

$$x(j)[d(j) - v(j)] \geq \min \left\{ c^+(j)[d(j) - v(j)], c^-(j)[d(j) - v(j)] \right\}$$

(1)

By summing over all the arcs of $A$ in (1), we have

$$\sum_{j \in A} x(j)d(j) \geq \sum_{j \in A} x(j)v(j)$$

$$+ \sum_{j \in A} \min \left\{ c^+(j)[d(j) - v(j)], c^-(j)[d(j) - v(j)] \right\}$$

$$= \sum_{j \in A} x(j)v(j) = \sum_{j \in A} x(j) \left[ e(i', j)u(i') + e(i, j)u(i)p(j) \right]$$

if $j \sim (i, i')$

$$= \sum_{j \in A} x(j) \left[ - \sum_{i \in N} e(i, j)u(i)p(j) \right]$$

$$= \sum_{i \in N} \left[ - \sum_{j \in A} e(i, j)p(j)x(j) \right] u(i) = - \sum_{i \in N} b(i)u(i)$$

If $(x(j), v(j)) \in A_j$, then equality holds in (1) by lemma 1 from which the equality in (2) follows. Conversely, if $(x(j), v(j)) \notin A_j$, then equality in (1) cannot hold (lemma 1)
from which it follows also that equality cannot hold in (2).

By the linear optimality theorem for flows, a flow $x$ is an optimal solution to the weighted linear optimal distribution problem if and only if $\text{div } x = b$, and there is a potential $u$ whose differential $v$ satisfies $(x(j), v(j)) \in A_j$ for every arc $j \in A$. (The sufficiency is the duality theorem and the necessity is the weighted optimal distribution algorithm which will be treated in Appendix B.)

Therefore, if the weighted linear optimal distribution problem is optimal with solution $x$, then there must be a potential $u$ whose differential $v$ is such that $(x(j), v(j)) \in A_j$ for every arc $j \in A$. And by lemma 2,

\[
\sum_{j \in A} x(j)d(j) = -\sum_{i \in N} b(i)u(i) \\
+ \sum_{j \in A} \min \left\{ c^+(j)[d(j) - v(j)], c^-(j)[d(j) - v(j)] \right\}
\]

which gives that

\[
\min \sum_{j \in A} x(j)d(j) = \max \sum_{i \in N} b(i)u(i) \\
+ \sum_{j \in A} \min \left\{ c^+(j)[d(j) - v(j)], c^-(j)[d(j) - v(j)] \right\}
\]

That completes the proof of the theorem.
7.2. Bechhofer and Kulkarni's Problem

7.2.1. Description of the problem

Bechhofer (1985) is an article which describes a new closed adaptive sequential procedure proposed by Bechhofer and Kulkarni for selecting the Bernoulli population which has the largest success probability. A class of sampling rules $R$ which take no more than $n$ observations from any of the $k$ populations is considered; the single-stage procedure proposed by Sobel and Huyett (1957) is in this class.

Now, we describe the procedure proposed by Bechhofer and Kulkarni in the article mentioned above. It is a multistage sequential procedure in which observations are taken one at a time (instead of in a single-stage). We denote the multistage procedure by $P = (R, S, T)$. The success probabilities of the populations are ordered in an ascending fashion:

$$p_1 \leq p_2 \leq \cdots \leq p_k \quad k \geq 2.$$

We denote a success (failure) from population $\pi_i$ at stage $m$ by

$$s_i^m, f_i^m (1 \leq i \leq k, 1 \leq m \leq kn-1).$$

Let $n_{i,m}$ denote the total number of observations sampled from $\pi_i$ through stage $m$, and let $z_{i,m}$ denote the total number of successes yielded by $\pi_i$ through stage $m$ ($1 \leq i \leq k, 1 \leq m \leq kn-1)$.

**Sampling rule** $(R)$: At stage $m$ ($1 \leq m \leq kn-1$), take the next observation from the population which has the smallest number of failures among all $\pi_i$ for which $n_{i,m} < n$ ($1 \leq i \leq k$). If there is a tie among such equal-number-of failure populations, take the next observation from
that one of them that has the largest number of successes. If there
is a further tie among such equal-number-of success populations, select
one of them at random and take the next observation from it.

Stopping rule (S): Stop sampling at the first stage m at which
there exists at least one population \( \pi_i \) satisfying

\[
    z_{i,m} \geq z_{j,m} + n - n_{j,m} \quad \text{for all } j \neq i (1 \leq i, j \leq k) \quad (1)
\]

Terminal decision rule (T): If \( r \geq 1 \) populations, say \( \pi_{i1}, \pi_{i2}, \ldots, \pi_{ir} \), simultaneously satisfy condition (1), then select one of them at
random as associated with \( p_{[k]} \quad k \geq 2 \).

Remark: Initially, (stage 0), one population is selected randomly
with probability \( \frac{1}{k} \quad (k \geq 2) \) to begin the sampling rule.

Condition (1) means that if at some stage \( m \) of the sampling scheme,
the number of successes already obtained from population \( i \) exceeds or
equals the sum of the number of successes already obtained and those
which can be obtained at most in all the remaining observations from
each one of the other populations, then one can stop sampling.

The sampling scheme of the multistage procedure can be repre­
sented as a network which consists of two different kinds of nodes
and arcs. A node \( i \) is called a decision or deterministic node (denoted
by \( d_i \)) if at that node a decision is made about which population the
next observation is to be sampled from. A node \( i \) is called stochastic
(denoted by \( s_i \)) if it is a successor of a deterministic node. It
follows that an arc \( j \) is called deterministic if its initial node is
deterministic and an arc \( j \) is called stochastic if its initial node is stochastic. For the case of two populations, the following diagram is an illustration of the two kinds of arcs and nodes.

\[d_1 \quad \downarrow \quad s_1 \quad \downarrow \quad s_2 \quad \downarrow \quad d_2 \quad \downarrow \quad d_3 \quad \downarrow \quad d_4 \quad \downarrow \quad d_5\]

\[p \quad 1-p \quad q \quad 1-q\]

\(d_1\) is a decision node. \(s_1\) and \(s_2\) correspond to the results of the decisions made at node \(d_1\). \(d_2\) and \(d_3\) correspond to the outcomes of the experiment made at \(s_1\) which are success or failure, respectively. If there are \(k\) populations, then there are \(k\) stochastic nodes at each level, each with two successors.

7.2.2. Formulation of Bechhofer and Kulkarni's problem

When no switching costs among the \(k\) different populations are taken into consideration, the following equalities and inequalities hold.

\[c(d_1) - c(s_1) \geq 1; \quad c(d_1) - c(s_2) \geq 1\]
\[ c(s_1) - pc(d_2) - (1-p)c(d_3) = 0 ; \quad c(s_2) - qc(d_4) - (1-q)c(d_5) = 0 \]

where \( c(i) \) stands for the expected cost of reaching a stopping node from \( i \).

The inequalities express the cost of doing the experiment, while the equations represent the usual expectation equations for disjoint stochastic events.

The multistage procedure of Bechhofer and Kulkarni consists of minimizing \( c(N^+) - c(N^-) \) subject to the above constraints. Where \( N^+ \) is the set of the first stochastic node of the network and \( N^- \) is the set of the final deterministic nodes of the network (leaves).

One can argue that the minimization makes of the inequalities an equation, so that the decision rule is picking the smallest \( c(s_1) \); the expected costs are unrestricted so that the problem is a network problem that resembles a maximum tension problem.

**Theorem:** The dual problem to the above (primal) problem is to maximize

\[ \sum_{j \in D} x(j) \]

where \( D \) is the set of the deterministic arcs of the network subject to the constraints:

\[ \sum_{j \in A} e(i,j)p(j)x(j) = 0 \quad \text{for all } i \in N \text{ except for } i \text{ starting node} \]
such that $p(j) = 1$ if $i$ is a deterministic node and $p(j) = p_k$ for $1 \leq k \leq k$ ($k \geq 2$) if $i$ is stochastic.

And $x(j) = x(j) = x(j')$ if $j$ and $j'$ are two stochastic arcs which have the same initial (stochastic) node and $j$ is a deterministic arc which has for final node the initial node common to $j$ and $j'$.

$x(j) \geq 0$ for all $j \in A$ and $\sum_{j \in L} p(j)x(j) = 1$ where $L$ is the set of stochastic arcs which have a stopping node as a final node.

The network starts with a special single (deterministic) node which is followed by $k$ stochastic nodes. At those nodes, the condition $x(1) + x(2) + \ldots + x(k) = 1$ holds. This is an extra condition not of the form $\sum_{j \in A} e(i,j)p(j)x(j) = 0$.

**Lemma:** $x(j) \in [0,1]$ for all $j \in D$

We will do an induction argument.

If $i$ is the initial deterministic node, we have the following diagram.
\[ x(1) + x(2) = 1 \]
which implies \( x(1) \leq 1 \)
and \( x(2) \leq 1 \) since \( x(j) \geq 0 \) for all \( j \in D \).

This is the first step of the induction.

\[ y(1) = y(2), x(1) = y(1) \]
\[ x(3) + x(4) = py(1) \]

It follows \( x(3) + x(4) \leq p \leq 1 \) by using the induction hypothesis \( x(1) \leq 1 \). This leads to \( x(3) \leq 1 \) and \( x(4) \leq 1 \). That completes the induction proof.

**Proof:** We will show first that \( c(N^+) - c(N^-) \geq \sum_{j \in D} x(j) \) and equality holds if the complementary slackness condition is met. The necessary condition is an algorithm which solves the dual problem and yields a solution to the primal problem as well. For those two respective solutions, the objective functions
of the primal and the dual problems have the same value.

We consider the following diagram.

\[ \text{The divergence conditions are} \]
\[ x(1) + x(2) = px(0) \]
\[ x(3) = x(1) \]

We multiply \( c(d_1) - c(s_1) \geq 1 \) by \( x(1) \)
\[ c(d_1) - c(s_2) \geq 1 \] by \( x(2) \)
\[ c(s_1) - pc(d_3) - (1-p)c(d_4) = 0 \] by \( x(3) \)
\[ c(s_0) - pc(d_1) - (1-p)c(d_2) = 0 \] by \( x(4) \)

The sum is a linear combination of the \( c \)'s and \( \sum_{j \in D} x(j) \).
To see that the c's corresponding to internal nodes have coefficient zero, consider \( c(d^-) \) whose coefficient is \( [x(1) + x(2) - px(0)] = 0 \) by the divergence condition.

Consider \( c(s^-) \), e.g., whose coefficient is \( x(3) - x(1) = 0 \) by the divergence condition. If \( d^- \) is the starting node, then \( c(d^-) \) has coefficient \( x(1) + x(2) = 1 \) and if \( d^- \) is a leaf, then its coefficient is \(-px(3)\). Since all the final nodes have the same value \( c(N^-) \) then all the coefficients of those add up to \( -\sum_{j \in L} p(j)x(j) \) by the condition \( \sum_{j \in L} p(j)x(j) = 1 \).

If the complementary slackness condition is not met that is at least one of the inequalities \( x(j)[c(d^-) - c(s^-) - 1] > 0 \) is strict for \( j \in D \), then \( \sum_{j \in D} x(j) < c(N^+) - c(N^-) \).

On the other hand, if all the inequalities of the above form are equalities, then one must have \( \sum_{j \in D} x(j) = c(N^+) - c(N^-) \) which gives optimality for both problems. Now, to complete the proof of the theorem we solve the dual problem by the use of an algorithm which is a special version of the weighted linear optimal distribution problem. The algorithm terminates with a pair of vectors \((x,c)\) which are solutions to the dual and primal problems, respectively.

The dual problem can be written as

\[
\text{minimize } \sum_{j \in D} -x(j) \quad \text{where } D \text{ is the set of the deterministic arcs.}
\]

subject to the constraints
\[ \sum_{j \in A} e(i,j)x(j)p(j) = 0 \] for every node \( i \) and \( p(j) \) as described earlier.

In fact, the dual problem is a weighted linear optimal distribution problem with \( d(j) = -1 \) for \( j \in D \) and \( d(j) = 0 \) for \( j \in A - D \).

The assumption \( \sum_{j \in P} e(i,j)p(j) = 0 \) for all \( i \in N \) where \( P \) is any circuit holds and the weighted linear optimal distribution algorithm as described in Chapter 12 is applied. It is initiated with the feasible solution such that \( x(j) = 0 \) for all \( j \in A \). For instance, for this initial feasible solution, the new system of span intervals is

\[ [d^-_x(j), d^+_x(j)] = \begin{cases} (-\infty, -1] & \text{if } j \in D \\ (-\infty, 0] & \text{otherwise.} \end{cases} \]

We implement the weighted linear optimal distribution algorithm in the same manner as it was described in Chapter 7 except for the notion of path which is different. A path in the new context is such that if a stochastic node belongs to an arc, then that arc must include both stochastic arcs which are incident to that stochastic node. At step 2 of the algorithm, that is when the outcome of step 0 is a circuit \( P \) with \( d^+_x(P) < 0 \), \( \alpha \) is defined by

\[ \alpha = \min \begin{cases} c^+(d) - x(j) & \text{for } j \in P^+ \\ x(j) - c^-(j) & \text{for } j \in P^- \end{cases} \]
where $P$ is a circuit such that if it includes a stochastic arc $j$ it also includes the stochastic arc $j'$ adjacent to $j$ and also all the arcs which are successors of $j$ or $j'$. We will adopt the terminology of multipath circuit for $P$.

REMARK: From linear programming, we know that the above weighted linear optimal distribution problem has a solution which means that $\alpha$ is expected to be finite for any multipath circuit $P$. 
8. CONCLUSION

8.1. Summary

The criterion of suitability of a method is often the economy and efficiency with which it can be programmed on a digital computer. In this dissertation, we are not concerned with computer programming. The illustrative examples presented at the end of some of the chapters are small enough to be solved by hand and may not apparently justify the methods recommended to solve them. As a matter of fact, the networks considered are only illustrative and the algorithms are really designed for large problems of the same type.

8.2. Suggestions for Future Research

Shier and Witzgalls (1980) discuss the sensitivity of solutions of the min path problem relative to changes in the spans. A similar study could be done about the sensitivity of solutions to the maximum weighted tension problem relative to changes in the weights defined on the arcs of the network. And likewise about the weighted linear optimal distribution problem.


10. ACKNOWLEDGMENTS

I wish to take this opportunity to express my sincere gratitude to my major professor, Dr. A. M. Fink for his constant encouragement, assistance, and especially for all of his time and guidance. I am grateful to him, also, for introducing me to the theory of network flows and for his constructive comments which led to many improvements of the manuscript. I would like also to thank my committee members Drs. Peter Colwell, Dale D. Grosvenor, Irvin R. Hentzel, Wolfgang H. Kliemann, Stuart A. Nelson, and Vincent A. Sposito. I also wish to record my appreciation of the inspiring guidance and excellent advice rendered by Dr. Wolfgang H. Kliemann during my last four years of graduate study at Iowa State University. Special thanks go to Maggie Wheelock for her time and skill in typing the manuscript. Last, but certainly not least, I wish to express my heartfelt gratitude to my wife, Aicha, for her moral support, continual encouragement and understanding throughout our stay in Ames.

Finally, I wish to thank my daughter Nasrine for her love and the amount of time she has spent without her father and my newborn daughter Soumeya for her smile and being so quiet during the nights.
11. APPENDIX A: THE WEIGHTED FLOW RECTIFICATION ALGORITHM

11.1. Description of the Algorithm

The algorithm to be described now solves the feasible distribution problem in a manner complementary to the feasible distribution algorithm developed in Chapter 4. Instead of maintaining feasibility with respect to capacities, and improving feasibility with respect to supplies by an application of the painted network algorithm at each iteration, it works with flows satisfying the supply constraint and improves feasibility with respect to capacities.

We start with the assumption \( b(N) = 0 \) and any flow \( x \) such that \( \tilde{y}(i) = b(i) \) for every node \( i \). Let \( A^+ = \{ j \in A / x(j) > c^+(j) \} \) and \( A^- = \{ j \in A / x(j) < c^-(j) \} \).

**Step 0:** If \( A^+ = \emptyset = A^- \), then \( x \) is a solution to the feasible distribution problem, and the algorithm terminates. If not, let \( \tilde{j} \) denote any arc in either \( A^+ \) or \( A^- \).

**Step 1:** Minty's algorithm is applied to \( \tilde{j} \) and the following painting of the arcs:

- green if \( c^-(j) < x(j) < c^+(j) \)
- white if \( x(j) < c^-(j) < c^+(j) \)
- black if \( x(j) > c^+(j) > c^-(j) \)
- red if \( c^-(j) = x(j) = c^+(j) \)
The arcs in $A^+$ are all black, whereas those in $A^-$ are white; in particular, $\overline{j}$ is either black or white.

**Step 2:** If the outcome is a cut $Q = [S, N - S]$ containing $\overline{j}$ and compatible with the above painting, then stop, there is no solution to the feasible distribution problem.

**Step 3:** If a compatible elementary circuit $P$ containing $\overline{j}$ is the outcome, define $a$, $b$, $c$, and $d$ as follows:

\[
a = \min_{j \in P^+} p(j) [c^+(j) - x(j)]
\]

\[
b = \min_{j \in P^-} p(j) [x(j) - c^-(j)]
\]

\[
c = \min_{j \in A^+} p(j) [x(j) - c^+(j)]
\]

\[
d = \min_{j \in A^-} p(j) [c^-(j) - x(j)]
\]

Let $\alpha = \min\{a, b, c, d\}$ and $\alpha(j) = \frac{\alpha}{p(j)}$ for $j \in A$. Then, define the flow $x'$ such that $x'(j) = x(j) + \alpha(j)e_p$, and go to step 0.

**11.2. Justification of the Algorithm**

If Minty's algorithm results in a cut $Q = [S, N - S]$ in step 1, then one has

\[x(j) \geq c^+(j) \text{ for all } j \in Q^+\]

\[-x(j) \geq -c^-(j) \text{ for all } j \in Q^-\]
with strict inequality for at least the arc $\bar{j}$. It follows

$$c^+(Q) < \sum_{j \in Q} x(j)p(j) - \sum_{j \in Q} x(j)p(j) = \text{weighted flux of } x \text{ across } Q$$

$$= \text{weighted divergence of } x$$

$$\text{from } S = b(S)$$

which violates the necessary condition of the feasible distribution problem.

With a circuit $P$ as the outcome in step 1, one has

$$c^+(j) - x(j) > 0 \quad \text{for } j \in P^+$$

$$x(j) - c^-(j) > 0 \quad \text{for } j \in P^-$$

The numbers $a$, $b$, $c$, and $d$ are all positive. And since the arc $\bar{j}$ belongs to either $A^+$ or $A^-$, either $c$ or $d$ must be finite which implies that $\alpha$ is finite and, consequently, so is $\alpha(j)$ for every $j \in A$.

Due to the choice of $\alpha(j)$ for $j \in A$, the flow $x'$ satisfies the following:

$$x'(j) \leq c^+(j) \quad \text{and} \quad c^-(j) - x'(j) = c^-(j) - x(j) - \alpha(j) \quad \text{for all } j \in P^+$$

$$x'(j) \geq c^-(j) \quad \text{and} \quad x'(j) - c^+(j) = x(j) - c^+(j) - \alpha(j) \quad \text{for all } j \in P^-$$

whereas $x'(j) = x(j)$ for $j \not\in P$.

This shows that the flow $x'$ is also feasible with respect to capacities. For divergence feasibility, $\text{div } x' = \text{div } x + \alpha \text{div } e_P = \text{div } x = b$ since
div \ e_p = 0 \text{ when } P \text{ is a circuit.}

The only arcs which violate the feasibility with respect to the capacity intervals, are those in \( A^+ \cup A^- \). The definition of \( x(j) \) is designed to yield improvement in \( j \), without "overshooting". So, at each iteration either at least one element of \( A^+ \cup A^- \) is removed or the value \( x(j) \) gets closer to either \( c^+(j) \) or \( c^-(j) \). So, if \( j \) is always chosen at step 0, as long as it is in \( A^+ \cup A^- \) then arc discrimination works for finiteness.

11.3. Example

![Graph]

Apparent solution \( x(AB) = x(AC) = 0 \)
\( x(CD) = 0 \)
\( x(BD) = -1 \)

We start \( x(AB) = -1 \) \( (p(AB) = p(AC) = p(CD) = p(BD) = 1) \).
\( x(AC) = 1 \)
\( x(CD) = 1 \)
\( x(BD) = -2 \)
Iteration 1

\[ A^+ = \phi, \quad A^- = \{AB, BD\}, \text{ choose } j = AB \text{ and paint the network.} \]

A circuit compatible with the painting is found.

\[
\begin{align*}
\alpha &= 2 \\
x'(AB) &= -1 + 1 = 0 \\
\beta &= 1 \\
x'(BD) &= -2 + 1 = -1 \\
\gamma &= \infty \\
x'(AC) &= 2 \\
\delta &= 1 \\
x'(CD) &= 2 \\
\end{align*}
\]

Update the flow.
Iteration 2

\[ A^+ = \{AC, CD\} \quad A^- = \{BD\} \]

Select \( j = BD \) (no choice) and paint the network.

A circuit compatible with the painting is found.

\[
\begin{align*}
  a &= \min\{1, 2\} = 1 & x'(AB) &= 1 \\
  b &= \min\{2, 2\} = 2 & x'(AC) &= 1 \\
  c &= \min\{1, 1\} = 1 & x'(BD) &= 0 \\
  d &= 2 & x'(CD) &= 1 \\
  \alpha &= 1 & \text{Update the flow.}
\end{align*}
\]
iteration 3

Now, $A^+ = \phi = A^-$. We stop because a solution is reached.

$x(AB) = x(AC) = x(CD) = 1$ and $x(BD) = 0$.

This shows again the nonuniqueness of a solution to the feasible distribution problem if there is any solution at all.
12. APPENDIX B: THE WEIGHTED LINEAR OPTIMAL DISTRIBUTION ALGORITHM

12.1. Description of the Algorithm

The algorithm starts with any feasible solution $x$ to the weighted linear optimal distribution problem.

Step 0: Define the following system of span intervals (see page 46) by

$$
[d_j, d_j] \text{ if } c^-(j) < x(j) < c^+(j)
$$

$$
(-\infty, d_j] \text{ if } c^-(j) = x(j) < c^+(j)
$$

$$
[d_j, \infty) \text{ if } c^-(j) < x(j) = c^+(j)
$$

$$
(-\infty, \infty) \text{ if } c^-(j) = x(j) = c^+(j)
$$

and apply the weighted tension rectification algorithm with these intervals.

Step 1: If the outcome of step 0 is a potential $u$ which differential $v$ satisfies

$$
d_x^-(j) \leq v(j) \leq d_x^+(j) \quad \text{for all } j \in A
$$

(12.1)

where $v(j) = [u(i') - u(i)]p(j)$ with $j \sim (i, i')$ then $x$ is an optimal solution and the algorithm terminates.

Step 2: If the outcome of step 0 is a circuit $P$ with $d_x^+(P) = \Sigma_{j \in P^+} d_x^+(j) - \Sigma_{j \in P^-} d_x^-(j) < 0$, then let

$$
\alpha = \min \begin{cases} 
  c^+(j) - x(j) & \text{for } j \in P^+ \\
  x(j) - c^-(j) & \text{for } j \in P^-
\end{cases}
$$
If $\alpha = \infty$, then the infimum in the problem is $-\infty$ and the algorithm terminates with no solution. Otherwise, let $x'(j) = x(j) + \frac{\alpha}{p(j)} e_p$ for all $j \in A$ and repeat step 0 with $x'$.

An important assumption which needs to be made is that $\sum_{j \in P} e(i,j)p(j) = 0$ for all $i \in N$ where $P$ is any circuit. (12.2)

12.2. Justification of the Algorithm

Claim 1: If the outcome of step 0 is a potential $u$ which differential $v$ satisfies

$$d_x^-(j) \leq v(j) \leq d_x^+(j) \quad \text{for all } j \in A \quad (12.1)$$

where $v(j) = [u(i') - u(i)]p(j)$ with $j \sim (i,i')$, then $x$ is an optimal solution to the problem.

Proof: Indeed, (12.1) is identical to $(x(j), v(j)) \in A_j$ for all $j$ which we demonstrated is a sufficient condition for optimality.

Claim 2: If the outcome of step 0 is a circuit $P$ with $d_x^+(P) < 0$ and

$$\alpha = \infty \text{ where } \alpha = \min \begin{cases} c^+(j) - x(j) & \text{for } j \in P^+ \\
 x(j) - c^-(j) & \text{for } j \in P^- \end{cases},$$

then there is no solution to the problem.

Proof: By assumption (12.2), the flow $x' = x + te_p$ is such that

$\text{div} \ x = \text{div} \ x' = b$ for all $t \in \mathbb{R}$.

Furthermore, $\alpha = \infty$ implies $c^+(j) = \infty$ for all $j \in P^+$ or $c^- (j) = -\infty$ for all $j \in P^-$ which means $x' = x + te_p$ will be
still feasible for all \( t \). If \( t \) is chosen arbitrarily high, then the cost of \( x' \) will be arbitrarily near \(-\infty\).

Conceivably, the algorithm might generate an infinite sequence of feasible solutions with decreasing costs and thus never terminate. This could result from failure to detect a circuit of the type obtained in step 2 with \( \alpha = \infty \), even though the infimum in the problem is \(-\infty\), but it could also occur when the infimum is finite. As a matter of fact, the max weighted flow algorithm may be identified with a special case of the weighted linear optimal distribution problem which means as mentioned earlier in Chapter 3 that the costs might converge to a value short of the minimum. We assume that there are no such circuits so that \( d^+(P) < 0 \) and \( \alpha = \infty \) in step 2, then the termination of the algorithm can be guaranteed by a commensurability condition. The algorithm must terminate if the infimum in the problem is finite, and the values \( c^-(j), c^+(j) \) and \( b(j) \) are all commensurable along with the initial flow values \( x(j) \). Indeed, if these values are all multiples of a certain \( \delta > 0 \), then so are \( \alpha \) and the successor flow values \( x'(j) \). All the flows generated by the algorithm therefore belong to the same commensurability class, and at every iteration the decrease in cost is at least \( \delta \epsilon \), where \( \epsilon \) is the smallest of the values \(|d^*e_p|\) corresponding to the finitely many circuits \( P \) with \( d^*e_p < 0 \). Since costs are bounded below (\( \alpha \) finite), there cannot be an infinite sequence of iterations.
12.3. Example

We consider a network with capacity intervals
\([c^-(j), c^+(j)] \subseteq [0,1]\) and cost coefficients \(d(j)\) all equal to 1.

The weights \(p(j)\) are as follows: \(p(AB) = \frac{1}{2}\), \(p(BC) = \frac{1}{4}\), \(p(AC) = 1\),
\(p(BD) = 2\), \(p(CD) = -1\). The supply values are indicated inside the
nodes. We initiate the algorithm with the flow \(x\) given by

\[
x(AB) = 1, x(AC) = 1, x(CB) = 0, x(BD) = 1 \text{ and } x(CD) = 0.
\]

We construct the new system of spans \([d^-_{x}(j), d^+_{x}(j)]\)
We pick the following potential \( u \) defined by \( u(A) = 0, u(B) = 2, u(C) = 1, u(D) = 3 \). The weighted differential of the potential \( u \) is \( v \) such that \( v(AB) = 1, v(AC) = 1, v(CB) = \frac{1}{4}, v(BD) = 2, v(CD) = -2 \).

This weighted differential is feasible with respect to the spans \( [d^{-}(j), d^{+}(j)] \). The algorithm terminates and the flow \( x \) chosen initially is an optimal solution to the problem.

The value of the objective function of the weighted linear optimal distribution problem is \( 1 + 1 + 0 + 1 + 0 = 3 \). On the other hand, the value of the objective function of the dual problem stated in the duality theorem is computed as follows:

\[
- \left( \frac{3}{2} \right)(0) - (2)\left( \frac{3}{2} \right) - (-1)(1) - (-2)(3) + \min\{1 - 1, 0\} + \min\{1 - 1, 0\}
+ \min\{1 - \frac{1}{4}, 0\} + \min\{1 - 2, 0\}
\]

\[= -3 + 1 + 6 - 1 = 3\]

which matches the value of the primal objective function. Also, \((x(j), v(j)) \in A_j\) for all \( j \in A \) which is easy to verify.

We set \( x(AB) = x_1, (AC) = x_2, x(CB) = x_3, x(BD) = x_4, x(CD) = x_5 \).

This weighted linear optimal distribution problem is the following linear programming problem:

\[
\text{Minimize } x_1 + x_2 + x_3 + x_4 + x_5
\]

subject to the constraints:
\[ 0 \leq x_i \leq 1 \quad \text{for all } i = 1, \ldots, 5 \]

\[ \frac{1}{2} x_1 + x_2 = \frac{3}{2} \]

\[ -x_2 + \frac{1}{4} x_3 - x_5 = -1 \]

\[ - \frac{1}{2} x_1 - \frac{1}{4} x_3 + 2x_4 = \frac{3}{2} \]

\[ -2x_4 + x_5 = -2 \]