1 Introdução

1.1. Motivação

A programação linear, ao menos na modelagem matemática que se conhece hoje, foi desenvolvida durante a segunda grande guerra quando foi utilizada no planejamento e execução de operações militares dos aliados. Com o fim da guerra foi publicado, em 1947, por George Dantzig, que havia servido o departamento de controle estatístico da força aérea americana, um método de resolução de problemas de programação linear que foi batizado de Simplex e veio a ser considerado por muitos como uma das maiores contribuições à matemática computacional de todo o século XX. No mesmo ano de 1947 John Von Neumann desenvolvia a teoria da dualidade. Dantzig e Neumann, juntamente com o cientista russo vencedor do prêmio Nobel de economia em 1975, Leonid Kantorovich, são considerados os pais da programação linear.

A partir de suas primeiras formulações, a programação linear ganhou a indústria em inúmeras aplicações práticas e vem se tornado o padrão em muitos processos produtivos. Entre muitos exemplos possíveis pode-se citar alguns, tais quais o roteamento de aeronaves e a atribuição da tripulação às mesmas ou a otimização do despacho em sistemas elétricos hidrotérmicos.

De uma maneira bem genérica pode-se definir como programação linear qualquer tipo de otimização de uma função linear (maximização ou minimização) quando sujeita a restrições dadas por equações ou inequações lineares. Este trabalho irá focar principalmente em problemas do tipo Fluxo de Custo Mínimo, uma classe de problemas bastante estudado pela comunidade científica e pela indústria devido a sua grande aplicabilidade em problemas práticos e que, conseqüentemente, apresenta diversas abordagens para a sua resolução, como exemplo pode-se citar o out-of-kilter de

Fulkerson (1961), Primal-Dual de Ford e Fulkerson (1957), Path de Busackerr e Gowen (1961), Dual de Balas e Hammer (1962), Negative-Cycle de Klein (1967) e Scaling de Edmonds e Karp (1972), além do próprio método Simplex, sem contar os inúmeros algoritmos desenvolvidos para problemas específicos.

Apesar de, conforme observado por Ahuja e Magnatti (1993), o método Simplex não ser exatamente uma das abordagens mais competitivas na resolução de Problemas de Fluxo de Custo Mínimo, este pode ser adaptado de maneira a explorar as características específicas da estrutura em rede apresentada por esta classe de problemas gerando um dos mais poderosos algoritmos disponíveis para este fim, o Simplex para Redes, aplicado pela primeira vez em problemas de caminho mais curto por Dantzig (1957) e Minty (1958) será o objeto de estudo deste trabalho.

1.2. Objetivos

Neste trabalho será desenvolvida a teoria necessária para o estudo do algoritmo Simplex para Redes, algoritmo que, como já foi adiantado anteriormente, apresenta grandes vantagens sobre outras abordagens ao explorar a estrutura das redes subjacentes na busca da solução ótima em Problemas de Fluxo de Custo Mínimo. Para tal será estudada a formulação deste tipo de problema além de apresentar um breve histórico de sua evolução e, a partir daí, será introduzida a solução dada pelo método Simplex padrão e será demonstrado como a especialização deste dá origem ao Simplex para Redes, aproveitando para clarear alguns pontos que são comumente negligenciados ou pouco explorados pela literatura de maneira geral.

No intuito de demonstrar a eficácia do algoritmo em questão e de promover um maior aprofundamento no assunto, o Simplex para Redes será implementado de uma forma computacionalmente eficiente e de fácil entendimento. A implementação será feita através de estruturas de dados especialmente desenvolvidas para melhorar o desempenho do algoritmo se comparado a outras abordagens bastante utilizadas na prática.

1.3. Roteiro dos capítulos

No segundo capítulo será feita uma revisão teórica dos Problemas de Fluxo de Custo Mínimo aonde serão formuladas algumas instâncias deste tipo de problema, a saber, os problemas de Transbordo, Caminho Mais Curto, Atribuição e Transporte.

No terceiro capítulo o Simplex para Redes será apresentado e toda a teoria necessária para o seu desenvolvimento será abordada em detalhes. Desta maneira serão revisados alguns conceitos de álgebra linear que serão utilizados na construção do algoritmo simplex e de sua especialização para problemas em redes, o Simplex para Redes. Durante este capítulo, paralelamente à apresentação dos conceitos teóricos, serão discutidos pontos relativos à implementação do algoritmo.

O quarto capítulo irá tratar da implementação e trará uma discussão sobre a estrutura de dados utilizada e sua representação lógica além de algumas das principais funções utilizadas no programa e da apresentação do programa em si e suas funcionalidades.

O quinto e último capítulo trará as conclusões deste trabalho e discutirá os resultados obtidos.

1.4. Definições

Antes de iniciar a teoria de grafos algumas definições precisam ser estabelecidas. Aqui, serão seguidas aquelas mostradas por Bazaraa *et al.* (1990) e por Ahuja e Magnatti (1993). Considere então, um grafo $G(N, A)$ onde N e A são respectivamente um conjunto de nós de cardinalidade $|N|$ = n e um conjunto de arcos de cardinalidade $|A|$ = m.

Adjacência: Dois nós i e j serão adjacentes se houver algum arco (i, j) ou (j,i).

Nós origem e destino: Para um arco direcionado (i, j), i será chamado de nó-origem e j de nó destino

Grafos direcionados: Um grafo é dito direcionado se contiver apenas arcos direcionados, que são arcos que permitem o fluxo em apenas uma direção.

Grafo bipartido: Grafo conectado e não direcionado tal que seja possível dividir seus nós em dois conjuntos, X e Y, de maneira que nenhum arco tenha início e fim no mesmo conjunto. Este conceito será de grande relevância para o desenvolvimento deste trabalho uma vez que a cada iteração do Simplex para Redes será criado um Grafo bipartido.

Corrente: Uma corrente em um grafo é uma seqüência de nós distintos tais que, com exceção do primeiro, todos os nós sejam adjacentes ao nó anterior.

Caminho: Um caminho em um grafo é uma corrente em que um nó qualquer é origem de, no máximo, um arco e destino de, no máximo, um arco.

Circuito: Um circuito é um caminho fechado, ou seja, um caminho em que o nó inicial e o nó final sejam coincidentes.

Ciclo: Um ciclo é um circuito em uma corrente.

Grafo conectado: Um grafo é dito conectado se existe uma corrente entre quaisquer pares de nós do grafo.

Grafo fortemente conectado: Um grafo fortemente conectado é aquele em há um caminho direcionado entre cada par de nós do grafo.

Subgrafo: $G(N', A')$ é um subgrafo de $G(N, A)$ se $N' \subseteq N$ e $A' \subseteq A$ e desde que se uma arco $(i, j) \in A'$, tanto i quanto j pertencem a N'.

Subgrafo gerador: Um subgrafo é dito gerador se contém todos os nós do grafo original, ou seja, se $N' = N$

Subgrafo induzido: Um subgrafo é dito induzido pelo conjunto de nós N' se contiver todas as arestas de A que tenham origem e destino dentro do conjunto N'

Árvore: Uma árvore é um grafo conectado que não contém ciclos. Se a árvore contiver todos os nós do grafo de onde ela se originou é chamada de árvore geradora.

Arco livre: Um arco é dito livre se for possível, respeitando-se as restrições de fluxo máximo e de fluxo mínimo, tanto um acréscimo como um decréscimo no seu fluxo.

Arco restringido: Um arco que não for livre será um arco restringido.

Rede Residual: Uma rede residual (figura 1.1b) é, com respeito a uma rede original (figura 1.1a), uma rede cujos arcos representam uma capacidade de aumento ou de diminuição do fluxo entre dois nós da rede original.

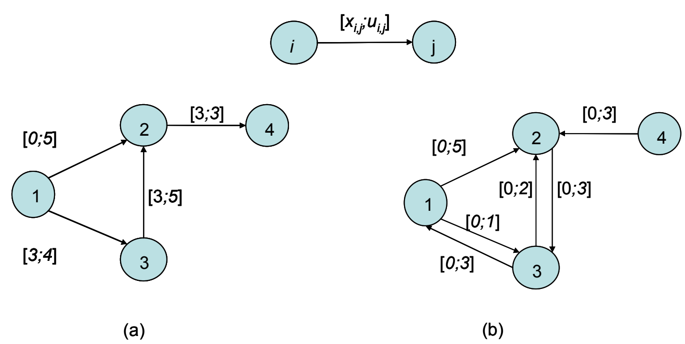

Figura 1.1 - Exemplo de rede residual.