UNIVERSIDADE FEDERAL DE MINAS GERAIS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

MODELOS E ALGORITMOS PARA PROBLEMAS DE ATRIBUIÇÃO DE CAPACIDADES E ROTEAMENTO EM REDES DE COMUNICAÇÃO

Ricardo Poley Martins Ferreira

Orientador: Henrique Pacca Loureiro Luna

TESE APRESENTADA COMO REQUISITO PARCIAL
PARA OBTENÇÃO DO TÍTULO DE
DOUTOR EM CIÊNCIA DA COMPUTAÇÃO
NA
UNIVERSIDADE FEDERAL DE MINAS GERAIS
BELO HORIZONTE, MINAS GERAIS

Março 2003

Sumário

Lista de Tabelas

Lista de Figuras

Sumário

Abstract

The joint problem of selecting routing and a capacity for each link in a communication network is considered. We apply an alternative approach for some models that have been addressed for computer networks discrete capacity allocation and routing problems. The network topology and traffic characteristics are assumed to be given. The goal is to obtain a feasible solution with minimum total cost, where the total cost include both leasing capacity and congestion costs. Heuristic algorithms with performance guarantees based on lower bounds and on the separability of the objective function are proposed. A heuristic algorithm is proposed to solve the problem with nonbifurcated routing constraints. An implicit enumeration approach is proposed to solve exactly the capacity expansion problem with two capacity levels for each channel. Experiments were conducted to verify the performance and to confirm the efficiency of the proposed algorithms.

Resumo

O problema conjunto de determinar rotas e as capacidades de cada arco em uma rede de comunicações é tratado. Uma abordagem alternativa para alguns modelos que foram propostos para o problema de atribuição de capacidades discretas e contínuas e roteamento em redes de computadores é adotada. A topologia da rede e as características de tráfego são considerados como sendo dados conhecidos. O objetivo é obter uma solução viável com o mínimo custo total, onde o custo total inclui tanto os custos de instalação de capacidade quanto os custos de congestionamento. Algoritmos heurísticos com desempenho garantido baseados em limites inferiores e na separabilidade da função objetivo são propostos. Um destes algoritmos é adaptado para considerar problemas com roteamento não-bifurcado. Um algoritmo exato baseado no método de enumeração implícita é proposto para resolver o problema de expansão de capacidade. Experimentos foram realizados para verificar o desempenho e confirmar a eficiência dos algoritmos propostos.

Agradecimentos

Eu gostaria de agradecer ao Prof. Henrique Pacca Loureiro Luna, pela sua orientação constante e paciente, pelas muitas sugestões, idéias e críticas e pelas oportunidades que me proporcionou indo muito além de seu papel de orientador.

Ao Prof. Philippe Mahey por ter me recebido no LIMOS/ISIMA e por ter me proporcionado um ambiente de trabalho propício para o florescimento de idéias.

Ao Prof. Ricardo Utsch de Freitas Pinto por me apresentar a programação matemática e pelas valiosas discussões.

Sou muito grato aos meus pais Carlos e Sônia, pelo amor, paciência e suporte sem os quais este trabalho não poderia ter sido realizado.

À minha esposa Nara e minha filha Bárbara pelo amor e pela compreensão das horas de ausência e ansiedade, e por tornarem minha vida doce.

Aos amigos que muito me ajudaram: Gilberto de Miranda Junior, Maurício Cardoso de Sousa, Denilson Laudares, Carlos Frederico, Ilmério, Paulo Cesar do Amaral, Autran e Zenilton Kleber do Patrocínio.

Aos professores, funcionários e colegas do Departamento de Ciência de Computação.

À CAPES e ao CNPQ pelo suporte financeiro desta pesquisa.

Belo Horizonte, Minas Gerais — Ricardo Poley Martins Ferreira
Março, 2003

Capítulo 1

Introdução

Decisões de expansão de capacidade são tomadas diariamente pelo governo, pelas empresas e pelas pessoas comuns. Algumas expansões envolvem grandes somas, como instalar internet de alta velocidade em todas as escolas públicas, outras requerem pequenas quantias, como a aquisição de um disco rígido. Algumas decisões levam anos de estudo e planejamento, outras se baseiam apenas na intuição. Expansão de capacidade é a adição de facilidades para servir a algum propósito: aumento do lucro, melhoria da qualidade de serviço, garantia de atendimento da demanda, aumento de confiabilidade, redução de custos.

A expansão de capacidade é um problema que está intimamente relacionado com o uso racional dos recursos existentes. Antes e depois de uma expansão, espera-se que as facilidades instaladas estejam sendo usadas da maneira mais eficiente possível. Dessa forma, a decisão de expansão e o planejamento do uso dos recursos disponíveis são problemas que precisam ser tratados conjuntamente.

Neste trabalho considera-se que decisões de expansão de capacidade e roteamento de fluxos em redes de comunicação por comutação de pacotes podem ser orientadas por modelos e métodos de programação matemática. Problemas reais de expansão de capacidade em redes são muito complexos. Entretanto, adotando-se hipóteses simplificadoras, modelos de programação matemática, que preservam as principais características de diferentes problemas reais, podem ser aplicados. Por exemplo, uma das principais simplificações adotadas refere-se ao fato de que a demanda por capacidade é determinada independentemente das decisões de expansão de capacidade.

O foco principal deste trabalho é o estudo de alternativas para modelos e algoritmos que têm sido aplicados na solução do problema de atribuição (expansão) de

capacidades e roteamento de fluxos em redes de comunicação. Uma rede de comunicação é assumida como sendo modelada por um grafo no qual os nós representam fontes ou sumidouros do tráfego ou nós de passagem, e os arcos representam os canais de comunicação da rede. A expansão de capacidades e roteamento é um caso particular do problema de atribuição de capacidades e roteamento, para o qual a capacidade inicial de cada arco é conhecida. O problema de atribuição de capacidades e roteamento de fluxos (conhecido como *Capacity and Flow Assignment problem — CFA problem*) é clássico no planejamento de redes de comunicação e tem tido uma importância crescente com o aumento da demanda nas redes de comunicação modernas [29][30][26][24] [4][5][52][22] [12][64]. Esse problema é modelado como um problema de otimização em que, se por um lado, deve-se definir as capacidades dos arcos com o mínimo custo de instalação, por outro, os fluxos de comunicação devem ser roteados de forma que o menor custo operacional seja obtido, satisfazendo restrições de rede, restrições orçamentárias e restrições de qualidade de serviço. Os custos de capacidade normalmente são referentes ao custo de instalar ou alugar níveis discretos de capacidade para cada arco da rede. O custo operacional é medido em termos da perda de qualidade de serviço e em custos relacionados com o transporte da informação através da rede. O problema de rede estudado não inclui aspectos topológicos como no problema geral de planejamento de redes (do inglês, *network design problem*), mas permanece sendo um problema desafiador porque as variáveis de projeto são interdependentes e a decisão de onde e qual capacidade instalar é de natureza combinatória, logo, resultando em problemas de otimização não convexos e combinatórios [52].

Instâncias significativas desses problemas de otimização são difíceis de serem resolvidas, e esta é uma das principais razões pelas quais atualmente a maior parte das abordagens de solução existentes na literatura se contentam com a utilização de algoritmos de aproximação, heurísticas e modelos de envoltórios para obter soluções satisfatórias ou ótimos locais [26][29][52][68].

Este trabalho pesquisa uma nova formulação integrada que associa custos operacionais relacionados com a qualidade de serviço oferecida pela rede com os custos de instalação de capacidades nos arcos. O problema *CFA* é formulado em termos de um único critério de custo [49]. Os métodos de solução desenvolvidos selecionam as capacidades dos arcos e as rotas, minimizando simultaneamente os custos de instalação de capacidades e os custos operacionais associados ao fluxo total em cada

arco. O objetivo deste trabalho é demonstrar que uma nova metodologia (formulação + algoritmos) adotada é eficiente e competitiva em relação a outras abordagens.

Uma revisão bibliográfica sobre as formulações e os métodos de solução do problema de atribuição de capacidades e roteamento de fluxos em redes de comunicação é feita no capítulo 2.

No capítulo 3 é apresentada uma extensão da formulação integrada para o problema proposta em [49], e são apresentados também dois algoritmos com desempenho garantido baseados em limites inferiores. A separabilidade da nova função objetivo é utilizada na obtenção de uma função de aproximação convexa cujo erro de aproximação pode ser calculado explicitamente. A formulação proposta considera que o fluxo de comunicação entre dois nós pode ser bifurcado, ou seja, os pacotes podem seguir caminhos distintos, não necessariamente disjuntos, entre cada par de nós que se comunicam. Os algoritmos desenvolvidos calculam fluxos e capacidades simultaneamente de forma a garantir um nível aceitável de desempenho com um mínimo custo total e usam como sub-rotinas métodos para resolver problemas de fluxo multiproduto com funções objetivo convexas. Experiências numéricas são apresentadas para verificar a eficiência dos algoritmos e a qualidade das soluções obtidas.

No capítulo 4 o problema de atribuição de capacidades e roteamento é formulado considerando que os pacotes devem seguir caminhos únicos entre cada par origem-destino. O problema de roteamento não-bifurcado é um problema combinatório NP-difícil. Métodos de roteamento com o método de desvio de fluxo (*Flow Deviation*) ou o método de decomposição proximal não são adequados para resolver problemas desse tipo. Entretanto, quando a rede possui algumas propriedades como uma demanda balanceada em um bom número de nós ($|n| \geq 25$), uma heurística gulosa inspirada no método de desvios de fluxos pode ser adotada [29]. Experiências numéricas são apresentadas para verificar a eficiência dos algoritmos e a qualidade das soluções obtidas e comparar os resultados com outros obtidos por diferentes métodos para resolver alguns dos problemas exemplos adotados.

Um algoritmo exato para a solução do problema de atribuição de capacidades e roteamento de fluxos é apresentado no capítulo 5. Esse algoritmo é inspirado no método de enumeração implícita [7][27] e é próprio para resolver problemas que possuam apenas uma opção de expansão de capacidade para cada arco e onde os fluxos adotem uma política de roteamento bifurcada. No Apêndice A um modelo contínuo

de atribuição de capacidades e roteamento de fluxos é proposto.

O capítulo 6 fecha o trabalho, com comentários, conclusões e sugestões para estudos futuros.

As principais contribuições desta tese são as seguintes:

- extensão da formulação proposta por Luna e Mahey [49] para o problema de expansão de capacidades e roteamento de fluxos de uma para múltiplas expansões de capacidades;
- extensão da formulação proposta por Luna e Mahey para considerar estratégias de roteamento não-bifurcado;
- extensão da formulação proposta por Luna e Mahey considerando que a capacidade assume valores contínuos;
- desenvolvimento de dois algoritmos com desempenho garantido para solução do problema de atribuição de capacidades e roteamento de fluxos;
- desenvolvimento de um algoritmo competitivo quando comparado a outros presentes na literatura na obtenção de bons limites superiores para o problema de atribuição de capacidades e roteamento de fluxos não-bifurcados;
- desenvolvimento de um algoritmo exato baseado no método de enumeração implícita apropriado para resolução do problema de atribuição de capacidades e roteamento em redes de comunicação;
- demonstração, através de experimentos, de que as metodologias propostas, envolvendo tanto a formulação como o algoritmo, são eficientes na solução dos problemas propostos;
- discussão sobre a aplicação de métodos de programação matemática em problemas de rede, principalmente considerando que o dinamismo da evolução das redes modernas não tem facilitado o desenvolvimento de formulações matemáticas consistentes e que sejam capazes de caracterizá-las sem a necessidade de se construir modelos de simulação ou mesmo realizar experimentos em redes físicas.

Capítulo 2

Problemas de Atribuição de Capacidades e Roteamento em Redes - Modelos, Formulações e Métodos de Solução

Neste capítulo apresentam-se os principais modelos e formulações para o problema de atribuição de capacidades e roteamento de fluxos em redes de comunicação e alguns dos algoritmos de solução existentes na literatura.

2.1 Modelos e formulações para problemas de atribuição de capacidades e roteamento de fluxos

O problema de atribuição de capacidades e roteamento de fluxos possui três aspectos concorrentes: o custo de investimento para instalação ou aluguel das linhas de comunicação, os custos operacionais associados aos custos de transmissão dos dados e a qualidade de serviço oferecida pela rede.

Redes de comunicação por pacotes são concebidas para transmitirem mensagens entre os usuários da rede. Uma mensagem é uma unidade de comunicação que deve ser transmitida completamente (preferencialmente sem perdas) de um usuário a outro ou de um a vários usuários. Para ser transmitida, a mensagem é codificada em uma cadeia de bits que é decomposta em seqüências menores de bits denominadas pacotes. Esses pacotes são enviados ao destinatário por caminhos eventualmente independentes. O roteamento consiste em definir por quais caminhos cada pacote deve seguir na rede, permitindo que a mensagem original seja remontada no seu destino

final. Toda vez que um pacote alcança um nó (roteador), esse pacote é atribuído a uma fila de espera para ser retransmitido ao próximo canal de comunicação de seu caminho (redes do tipo armazena e envia). Este processo é a principal causa de atrasos no envio de uma mensagem.

As redes por comutação por pacotes (como a Internet) não foram concebidas inicialmente para levar em conta parâmetros modernos de qualidade de serviço. Essas redes por comutação de pacotes foram desenvolvidas em uma época em que a capacidade dos arcos (banda passante) era escassa. A estratégia então era obter uma ocupação máxima dos canais de conexão mesmo que isso introduzisse atrasos adicionais de transmissão. Neste contexto, nas abordagens clássicas do problema de atribuição de capacidades e roteamento de fluxos em rede por comutação de pacotes, minimizar o atraso médio na rede era o principal critério de medida da qualidade de serviço. O atraso pode ser ocasionado por quatro causas principais [9][42]:

- atraso de processamento: tempo gasto para processar um pacote a cada nó intermediário e prepará-lo para retransmissão. Esse atraso é determinado pela complexidade do protocolo e pela capacidade computacional disponível em cada nó intermediário;

- atraso de propagação: tempo que leva para um pacote percorrer o canal de comunicação. Esse atraso é determinado pela distância ou comprimento do caminho percorrido. Ele pode ser significativo principalmente em canais via satélite, e em canais de alta velocidade quando o atraso de propagação pode ser uma parcela significativa do atraso total;

- atraso de transmissão: tempo gasto na transmissão de uma mensagem completa, do primeiro ao último bit, em um canal de comunicação. O atraso de transmissão é principalmente definido pela velocidade de transmissão do canal, por exemplo, um canal de 256 kbps acarreta somente a metade do atraso de um canal de 128 kbps;

- atraso de congestionamento (ou atraso de fila de espera): tempo que um pacote tem que esperar em uma fila para ser atendido. O atraso de congestionamento é ocasionado pelo congestionamento nos nós intermediários. Filas são naturais em redes do tipo comutadas por pacotes. Os pacotes chegam de maneira

assíncrona nos nós intermediários e são processados e reenviados de acordo com algum protocolo de serviço. Como, normalmente, um nó intermediário não é capaz de manipular simultaneamente todo o tráfego que chega, os pacotes que chegam são armazenados temporariamente (no *buffer*) enquanto aguardam a vez de serem processados e retransmitidos. O atraso de congestionamento é determinado geralmente pelo congestionamento nos nós intermediários, o qual é governado pela estatística de chegada de pacotes e pela disciplina de serviço. Mais especificamente, o congestionamento (uma fial) em um nó é determinado pelos seguintes fatores:

chegada de um número excessivo de pacotes devido às flutuações estatísticas da geração de tráfego;

tamanho da memória temporária (*buffer size*), os pacotes são descartados quando a memória se esgota;

número de servidores (número de filas);

o tempo de serviço que é determinado pelo tamanho do pacote e pela taxa de transmissão do canal.

Juntos, esses parâmetros determinam o atraso de um pacote. Uma fila é especificada por uma sêxtupla $A/B/c/d/e/f$ (notação de Kendall), em que os dois primeiros especificadores A e B denotam as estatísticas de chegada de serviço, respectivamente, e c e d denotam o número de servidores e a capacidade do sistema (tamanho da memória), respectivamente e e e f denotam o tamanho da polução e a disciplina de atendimento. Em uma fila do tipo $M/M/1$ as estatísticas de chegada e de serviço são distribuições exponenciais, e, em cada nó, existe apenas um servidor (uma fila). Como a capacidade do sistema não foi especificada, considera-se que o sistema possui memória temporária infinita (*buffer* infinito). Uma rede pode ser caracterizada como sendo uma rede de filas $M/M/1$ onde cada fila representa um canal de saída em cada um dos nós da rede.

Nas formulações clássicas do problema *CFA*, redes de filas $M/M/1$ são consideradas adequadas para descrever o comportamento das redes. As formulações mais modernas têm introduzido outros modelos procurando acompanhar as inovações tecnológicas. Por exemplo, a probabilidade de ocorrência de estouro das filas nos roteadores (*buffer overflow*) tem sido usada como critério de qualidade de serviço da rede em alguns dos trabalhos mais recentes. Esse critério é baseado no fato de que os atrasos de propagação e transmissão são dominantes nas redes de comunicação moderna, e a perda de um pacote durante a transmissão afeta negativamente a qualidade de serviço da rede [55][13][53][40].

Há na literatura diversos modelos que tratam do problema de atribuição de capacidades e roteamento de fluxos. Uma das características que distinguem esses modelos é a natureza das capacidades dos arcos que podem assumir valores discretos ou contínuos. Os custos destas facilidades adicionais podem ou não possuir economia de escala (isto é, seu custo é menor ou não que o custo proporcional à capacidade). As estratégias de roteamento adotadas (bifurcadas ou não bifurcadas) também caracterizam as formulações e os métodos de solução. A seguir serão apresentados alguns desses modelos e as suas respectivas formulações.

2.1.1 Modelos contínuos

As versões clássicas do problema *CFA* de atribuição de capacidades e roteamento de fluxos definem as capacidades dos canais de comunicação e roteiam os fluxos de forma a garantir que o atraso médio da rede seja mantido em um patamar aceitável. Isso acontece devido à procura de uma solução de mínimo custo. Outra abordagem adotada define limitações orçamentárias e procura obter o menor atraso possível com os recursos disponíveis [29][30][26][24][4][5]. Descrições:

Modelo contínuo - minimizando o custo de instalação de capacidades *CFA1*:

Conhecendo: a topologia da rede,
o custo de instalação de capacidades em cada arco $\theta_i(c_i)$,
o vetor de demandas máximas.

Determinar: as capacidades que minimizem o custo total de instalação $\varphi(c) = \sum_{i=1}^{n} \theta_i(c_i)$,
onde $\theta_i(c_i)$ é uma função contínua (normalmente côncava).

Variáveis: os fluxos f_i e as capacidades c_i nos arcos.

Sujeito a: atraso médio $T(f, c)$ da rede que deve ser inferior a um valor máximo estabelecido,
fluxo total em cada arco não pode ultrapassar a capacidade do arco,
satisfazer todas as demandas,
restrições de fluxo através da rede.

Modelo contínuo - minimizando o atraso médio da rede (*CFA2*):

Conhecendo: a topologia da rede,
o custo de instalação de capacidades em cada arco $\theta_i(c_i)$,
o vetor de demandas máximas.

Determinar: o roteamento que minimize o atraso médio da rede $T(f, c)$.

Variáveis: os fluxos f_i e as capacidades c_i nos arcos.

Sujeito a: custo total de instalação e operação de capacidades em cada arco $\theta_i(c_i)$ que não deve ultrapassar um limite máximo preestabelecido,
fluxo total em cada arco não pode ultrapassar a capacidade do arco,
satisfazer todas as demandas,
restrições de fluxo através da rede.

Formulações

Esses modelos podem ser formulados considerando uma rede como sendo um grafo orientado $G = (V, A)$ com n nós e m arcos pelos quais passam $k = 1, ..., K$ produtos. Para cada par origem O^k destino D^k de um produto k, é associada uma demanda d^k, onde:
c_i, capacidade de um arco i,
f_i, fluxo total no arco i,
A, matriz de incidência do grafo $G = (V, A)$,
d^k, vetor de demandas,
x_i^k, fluxo do produto k passando pelo arco i,
$T(f, c)$, atraso médio da rede,
$\theta_i(c_i)$, custo de instalação de capacidades em cada arco,

$\varphi(c)$, custo total de instalação de capacidades na rede,
T_{max}, atraso médio máximo permitido,
Tem-se então:

$$[CFA1] \begin{cases} minimizar: & \varphi(c) = \sum_{i=1}^{m} \theta_i(c_i) & \\ sujeito\ a: & T(f,c) \leq T_{max} & \\ & f_i = \sum_{k=1}^{K} x_i^k & \forall i = 1,...,m \\ & f_i \leq c_i, & \forall i = 1,...,m \\ & Ax^k = d^k, & \forall k = 1,...,K \\ & x \in R^{Km+} & \\ & f \in R^{m+} & \\ & c_i \in R^{m+} & \end{cases}$$

$$[CFA2] \begin{cases} minimizar: & T(f,c) & \\ sujeito\ a: & \varphi(c) = \sum_{i=1}^{m} \theta_i(c_i) \leq \varphi_{max} & \\ & f_i = \sum_{k=1}^{K} x_i^k & \forall i = 1,...,m \\ & f_i \leq c_i, & \forall i = 1,...,m \\ & Ax^k = d^k, & \forall k = 1,...,K \\ & x \in R^{Km+} & \\ & f \in R^{m+} & \\ & c_i \in R^{m+} & \end{cases}$$

Essas formulações foram propostas por Gerla em [29]. Em princípio, não há nada que indique qual das formulações é mais apropriada [47]. Uma questão natural que surge é se as soluções obtidas são equivalentes. Humes em [38] apresenta algumas respostas como, por exemplo: quando os dois problemas são viáveis, eles possuem solução ótima, e as seguintes conclusões são válidas:

- $\forall\ T_{max} > 0$, (T_{max}, φ^*) corresponde a um projeto eficiente, onde φ^* é o valor

ótimo de $[CFA1]$.

- $\forall\ \varphi_{max} > \varphi_0$, (T^*, φ_{max}) corresponde a um projeto eficiente, onde T^* é o valor ótimo de $[CFA2]$.

Essas formulações são dependentes das funções que descrevem o atraso médio na rede $T(f, c)$ e o custo de instalação de capacidades em cada arco $\theta_i(c_i)$. A seguir é apresentada uma breve discussão sobre essas funções.

O atraso médio de congestionamento $T(f, c)$

Um dos resultados adotados pelas formulações apresentadas acima é uma expressão *analítica* para o atraso médio de congestionamento em uma rede de computadores [42].

Essa expressão baseia-se no fenômeno de fila que é observado em cada arco como um servidor cuja taxa de serviço é determinada pela sua capacidade e considerando as mensagens nos arcos como clientes competindo por esse serviço. O modelo resultante que caracteriza o atraso de congestionamento é de uma rede de filas do tipo M/M/1. Várias hipóteses simplificadoras foram necessárias para tornar tratável o modelo de redes de filas resultante. As hipóteses são: chegadas nos nós obedecendo a uma distribuição de Poisson, independência dos processos de chegada nos nós, roteamento determinístico, memória do nó infinita, e o atraso de propagação do pacote sendo desprezado e principalmente distribuição exponencial no tamanho das mensagens. Esta última hipótese afirma que toda vez que uma mensagem chega a um nó da rede, um novo tamanho é escolhido para esta mensagem independentemente seguindo uma distribuição exponencial. Sem a hipótese de independência o problema de obter o atraso médio em uma rede é intratável. Sob essas hipóteses o atraso médio no canal de comunicação i é dado por:

$$T_i = \frac{1}{\mu c_i - \lambda_i}, \tag{2.1.1}$$

onde: $\frac{1}{\mu}$, tamanho médio das mensagens $[bits/menssagem]$,
c_i, capacidade do canal $[bits/segundo]$
λ_i, tráfego médio de mensagens no canal i $[menssagens/segundo]$.

O atraso médio na rede é então dado por:

$$T(f,c) = \sum_{i=1}^{m} \frac{\lambda_i}{\Gamma} T_i = \frac{1}{\Gamma} \sum_{i=1}^{m} \frac{\lambda_i/\mu}{c_i - \lambda_i/\mu}, \tag{2.1.2}$$

onde Γ é o somatório do tráfego médio de mensagens entre cada par origem-destino da rede: $\Gamma = \sum_{k=1}^{K} d^k$.

Sendo $\lambda_i/\mu = f_i$, tem-se, então, o atraso médio de congestionamento:

$$T(f,c) = \frac{1}{\Gamma} \sum_{i=1}^{m} \frac{f_i}{c_i - f_i} \tag{2.1.3}$$

Custos de instalação de capacidades em cada arco $\theta_i(c_i)$

Um caso especial do problema $[CFA1]$ que ilustra bem as dificuldades para a solução deste foi proposto por Kleinrock [41] que sugere uma formulação em que o problema $CFA1$ é reduzido a um problema de roteamento de fluxos em uma rede multiproduto não capacitada.

Para obter essa formulação, Kleinrock considera que a função de custo de capacidades $\theta_i(c_i)$ é uma função contínua e linear $\theta_i(c_i) = \alpha_i c_i + \beta_{i0}$ e considera que o atraso médio da rede [9][43] é dado por 2.1.3.

Adotando essas funções e aplicando as condições de otimalidade de Kuhn-Tucker ao problema original $CFA1$, foi possível obter uma formulação explícita da função objetivo do problema somente em função dos fluxos no arco.

Nesse caso, existe uma função explícita da capacidade ótima em um arco c_i em função do fluxo no arco f_i:

$$c_i = f_i + \frac{\sum_{i=1}^{m} \sqrt{\alpha_i f_i}}{\lambda T_{max}} \sqrt{\frac{f_i}{\alpha_i}} \tag{2.1.4}$$

Obteve-se, então, uma formulação específica para o problema $CFA1$:

$$[CFA3] \begin{cases} minimizar: & \Phi(f) = \sum_{i=1}^{m}(\alpha_i f_i + \beta_{i0}) + \frac{(\sum_{i=1}^{m}\sqrt{\alpha_i f_i})^2}{\lambda T_{max}} \\ sujeito\ a: & f_i = \sum_{k=1}^{K} x_i^k \ \ \forall i = 1, ..., m \\ & Ax^k = d^k, \ \ \forall k = 1, ..., K \\ & x \in R^{Km+} \\ & f \in R^{m+} \end{cases}$$

A função objetivo $\Phi(f)$ obtida nessa formulação é côncava.

Um modelo mais realista do problema de atribuição de capacidades e roteamento é o que considera a função de custo das capacidades nos arcos $\theta_i(c_i)$ como sendo contínua e côncava (um modelo mais próximo ainda da realidade deve considerar capacidades discretas). Nesse caso não é possível expressar o custo ótimo de atribuição de capacidades explicitamente em função do fluxo f. Mesmo assim, Gerla [29] demonstrou que $\Phi(f)$ continuava sendo uma função côncava. Esse fato permitiu caracterizar os ótimos locais dessa função. Um caso particular da função que descreve o custo das capacidades ocorre quando ela é do tipo "*power law cost*" $\theta_i(c_i) = \kappa_i(c_i)^\alpha$ onde $0 \leq \alpha \leq 1$. Os problemas de atribuição de capacidades e roteamento (caso contínuo) são de minimização com funções objetivos côncavas; uma breve discussão das propriedades desse problema de otimização global é, então, justificada [31][35][37][57].

O objetivo de um problema de minimização côncava é o de minimizar uma função côncava contínua $F(.) : \Omega \rightarrow R$, definida em algum conjunto $\Omega \subset R^n$, com um domínio convexo $Dom \subset \Omega$ definido como: $Dom = \{x \in R^n : g_j(x) \leq 0, \ j = 1, ..., J\}$, onde $g_j(x)$ são funções contínuas e convexas definidas em R^n.

A existência de muitos extremos (ótimos locais) é certamente uma das principais características de um problema de otimização côncava. De fato é possível construir funções $F(.)$ e poliedros Dom que possuam a propriedade de que todo vértice de D seja um ótimo local de $F(.)$. Por exemplo, no caso de minimizar uma função côncava quadrática sobre um politopo simples como um "hipercubo". Assim, do ponto de vista de complexidade computacional, freqüentemente os problemas de minimização côncava são NP-difíceis. Uma das principais dificuldades enfrentadas para a solução de problemas multiextremais é a dificuldade de se obter condições de otimalidade

(necessárias e suficientes) capazes de garantir que um determinado ponto viável seja um ponto de máximo ou de mínimo global da função objetivo [58][36]. O desconhecimento dessas condições dificulta a elaboração de algoritmos capazes de resolver problemas de otimização global [59].

Diante dessa dificuldade, uma alternativa possível é explorar propriedades que caracterizem os ótimos locais para propor algoritmos que sejam capazes de determinar o ótimo global . Por exemplo, entre essas propriedades, uma que está entre as mais utilizadas surge quando a função objetivo $F(.)$ é côncava; o ótimo global de um problema côncavo em um domínio convexo Dom é um ponto extremo deste domínio (geralmente vértice de um poliedro)[58]. Assim, uma maneira de determinar o ótimo global é enumerar todos os pontos extremos do domínio. Entretanto, no caso do poliedro que descreve as restrições de um problema de multiprodutos, o conjunto de pontos extremos pode ser extremamente grande mesmo para problemas com apenas algumas variáveis [60].

Abordagem alternativa para a medida de atraso

LeBlanc e Simmons em [46] propõem usar como medida de atraso na rede uma função empírica de atraso derivada de uma função utilizada pela *U.S. Federal Highway Administration* [1] para medir o atraso no tráfego de suas rodovias.

$$T(f_i, c_i) = \frac{a f_i^{n+1}}{(\epsilon + b c_i)^n} + e f_i \tag{2.1.5}$$

O termo ϵ foi adicionado para que a função seja definida quando $f = c = 0$. As constantes a, b e e são empíricas. Essa função é convexa em f e em c simultaneamente. Eles propõem adotar essa função no lugar de 2.1.3 partindo do argumento de que a função de atraso convencional não representa um modelo consistente com a prática, na medida em que suas hipóteses básicas são muito restritivas.

Considerando que a função que descreve os custos de instalação de capacidades é linear $\theta_i(c_i) = \alpha_i c_i$, eles adotaram a formulação $CFA1$. O problema obtido possui uma função objetivo convexa e restrições convexas. Instâncias significativas dessa formulação podem ser resolvidas sem maiores dificuldades pelos métodos da programação matemática.

2.1.2 Modelos discretos

Quando as capacidades são discretas, foram encontrados basicamente dois modelos principais. O primeiro modelo é similar ao modelo contínuo $CFA1$. Nele o atraso da rede é tratado como uma restrição que limita o atraso médio máximo da rede. O segundo modelo considera como sendo conhecido o custo de uma unidade de atraso e trata o atraso da rede como parte integrante da função objetivo a ser minimizada.

Primeiro modelo discreto [$DCFA1$]:

Conhecendo: a topologia da rede,
os custos de instalação e ou operação de capacidades em cada arco $\theta_i(c_i)$,
os valores discretos c_i,
o vetor de demandas máximas.

Determinar: as capacidades que minimizem o custo total de intalação $\varphi(c) = \sum_{i=1}^{n} \theta_i(c_i)$.

Variáveis: os fluxos f_i e as capacidades c_i nos arcos.

Sujeito a: atraso médio da rede que deve ser inferior a um valor máximo estabelecido,
satisfazer todas as demandas,
as restrições de fluxo através da rede.

Segundo modelo discreto [$DCFA2$]:

Conhecendo: a topologia da rede,
os custos de instalação e operação de capacidades em cada arco $\theta_i(c_i)$,
o vetor de demandas máximas.
o custo de fluir uma unidade de fluxo por um arco υ_i,
o custo de uma unidade de atraso ρ (unidade de custo/tempo).

Determinar: as capacidades que minimizem $\varphi(c, f) = \sum_{i=1}^{n} \theta_i(c_i) + \rho T(f, c) + \sum_{i=1}^{n} \upsilon_i f_i$,

Variáveis: os fluxos f_i e as capacidades c_i nos arcos.

Sujeito a: fluxo no arco que não pode ultrapassar a capacidade do arco,
satisfazer todas as demandas,
as restrições de fluxo através da rede.

Formulações

As formulações desses modelos seguem a mesma linha dos modelos contínuos. Serão designados por $DCFA$ - *Discret Capacity and Flow Assignment*, e considerando uma rede como um grafo orientado $G = (V, A)$ com n nós e m arcos pelos quais passam $k = 1, ..., K$ produtos. Para cada par origem O^k destino D^k, é associada uma demanda d^k, onde:
c_i, a capacidade de um arco i;
f_i, fluxo total no arco i;
A, matriz de incidência;
d^k, vetor de demandas;
x_i^k, fluxo do produto k passando pelo arco i;
$T_i(f_i, c_i)$, contribuição ao atraso médio do arco i;
$T(f, c)$, atraso médio da rede;
$\theta_i(c_i)$, custo de instalação de capacidades em cada arco;
$\varphi(c)$, custo total de instalação de capacidades na rede;
T_{max}, atraso médio máximo permitido;
υ_i, custo de fluir uma unidade de fluxo por um arco;
c_i, capacidade do arco i;
C_i, conjunto das capacidades que estão disponíveis para serem instaladas no arco i.

Primeiro problema:

$$DCFA1 \begin{cases} minimizar: & \varphi(c) = \sum_{i=1}^{m} \theta_i(c_i) & \\ sujeito\ a: & T(f, c) \leq T_{max} & \\ & f_i = \sum_{k=1}^{K} x_i^k & \forall i = 1, ..., m \\ & f_i \leq c_i, & \forall i = 1, ..., m \\ & Ax^k = d^k, & \forall k = 1, ..., K \\ & x \in R^{Km+} & \\ & f \in R^{m+} & \\ & c_i \in C_i & \end{cases}$$

Segundo problema:

$$
DCFA2 \begin{cases}
minimizar: & \varphi(c,f) = \sum_{i=1}^{n} \theta_i(c_i) + \rho T(f,c) + \sum_{i=1}^{n} \upsilon_i f_i & \\
sujeito\ a: & f_i = \sum_{k=1}^{K} x_i^k & \forall i = 1,...,m \\
 & f_i \leq c_i, & \forall i = 1,...,m \\
 & Ax^k = d^k, & \forall k = 1,...,K \\
 & x \in R^{Km+} & \\
 & f \in R^{m+} & \\
 & c_i \in C_i &
\end{cases}
$$

A segunda formulação segue uma estratégia clássica de transformar um problema multicritério em um problema monocritério, através de uma combinação linear [47]. Os modelos e as formulações discretas e contínuas são similares. Boa parte das estratégias de solução se baseiam em resolver a formulação contínua e posteriormente obter a solução inteira a partir das respostas obtidas. Nessas duas formulações apresentadas, cada pacote de uma mensagem pode seguir seu próprio caminho ao longo da rede.

Gavish e Neuman em [26] propõem uma formulação que adiciona restrições fazendo com que o roteamento de um pacote siga um caminho único entre a sua origem e o seu destino.

Terceiro problema:

$$
[DCFA3] \begin{cases}
minimizar: & \varphi(c,f) = \sum_{i=1}^{m} \theta_i(c_i) + \rho T(f,c) + \sum_{i=1}^{m} \upsilon_i f_i & & \\
sujeito\ a: & f_i = \sum_{k=1}^{K} x_i^k & & \forall i = 1,...,m \\
& \sum_{r \in \Upsilon} \delta_{ri} y_r^k d_i^k \leq c_i, & \forall k = 1,...,K, & \forall i = 1,...,m \\
& \sum_{r \in \Upsilon} y_r^k = 1 & & \forall k = 1,...,K \\
& y_r^k = 0,1 & \forall r \in \Upsilon & \forall k = 1,...,K \\
& Ax^k = d^k, & & \forall k = 1,...,K \\
& x \in R^{Km+} & & \\
& f \in R^{m+} & & \\
& c_i \in C_i & &
\end{cases}
$$

Onde:

Υ, o conjunto de rotas disponíveis,

y_r^k, uma variável de decisão que informa se a rota r foi escolhida pelo produto k,

δ_{ri}, uma função de indicação, que assume o valor unitário se o arco i é utilizado pelo rota r, e zero caso contrário.

Formulação similar a esta foi utilizada também em [4].

Diversos autores propuseram formulações desconsiderando as economias de escala na atribuição das capacidades dos arcos. Suas formulações resultam em problemas lineares ou convexos que podem ser resolvidos de forma a se determinar o ótimo global [46] [55] [40] [13].

2.2 Métodos de solução

Nesta seção, serão apresentados alguns dos algoritmos de solução propostos na literatura para resolver os problemas formulados na seção anterior.

2.2.1 Métodos de solução para problemas com capacidades contínuas

Método do ponto fixo

Uma das primeiras tentativas de resolver o problema de roteamento em uma rede multiproduto com uma função objetivo côncava estritamente crescente e derivável (como $CFA3$) foi feita por Yaged ([66]). Yaged estendeu os resultados que Zangwill ([68]) tinha obtido para o problema com um só produto (uniproduto) e custos côncavos. De maneira geral, o método trabalha realizando sucessivas aproximações lineares do problema até alcançar um ótimo local.

Esse método se baseia em algumas propriedades que caracterizam um ótimo local:

- devido à concavidade da função objetivo, uma solução ótima local possui a propriedade de que o fluxo de um produto k entre cada par origem O^k e destino D^k pode seguir um caminho único.

- se dois nós da rede, respectivamente origem e destino do fluxo de um produto k pertencem ao caminho entre a origem e o destino do fluxo de um outro produto $k^{'}$, o caminho utilizado pelo fluxo de k pertence necessariamente ao subcaminho utilizado por $k^{'}$ entre a origem e o destino do produto k;

- se um roteamento é ótimo local, ele é também $\varepsilon-$ótimo local. Ou seja, além de satisfazer as duas propriedades precedentes, se para cada produto k é feita uma perturbação ε da demanda d^k, esta pertubação não provoca mudança no caminho utilizado por k;

Considere o roteamento dos fluxos dos produtos $\bar{f}^{'}$ como o roteamento que segue os caminhos mais curtos da rede. Os caminhos mais curtos são obtidos atribuindo às distâncias entre os nós da rede l_i os valores das derivadas da função objetivo nos arcos $l_i = \Phi_i^{'}$ ($l_i \geq 0$ uma vez que $\Phi(f)$ é estritamente crescente). Considere a transformação L que mapeia $\bar{f}$ em $\bar{f}^{'}$, $\bar{f}^{'} = L(\bar{f})$. Yaged demonstrou que $\bar{f}$ é um ponto que satisfaz as condições de otimalidade de Kuhn-Tucker se e somente se ele for um ponto fixo de L, ou seja: $\bar{f} = L(\bar{f})$. A função objetivo sempre diminui aplicando a transformação L, $\Phi(\bar{f}^{'}) < \Phi(\bar{f})$ a menos que $\bar{f} = \Phi(\bar{f})$. Aplicando este conjunto de idéias, Yaged desenvolveu um algoritmo que possui convergência finita e que consiste em determinar um ponto fixo da função $\Phi(f)$.

Algoritmo do ponto fixo

Passo 1- Seja $it = 0$.
Seja f^0 uma solução inicial.

Passo 2- Na iteração it, calcule $\bar{f}^{it+1} = L(\bar{f}^{it})$.

Passo 3- Se $\bar{f}^{it+1} = \bar{f}^{it}$, interrompa.
Senão, faça $it = it + 1$ e volte ao **Passo 2**.

Como se trata de um problema de minimização côncava, este algoritmo converge para um ótimo local que depende do roteamento inicial adotado.

Método de separação e avaliação

Minoux [51][52] desenvolveu um algoritmo de separação e avaliação para resolver o problema de roteamento em redes multiproduto que possuam uma função objetivo côncava explícita estritamente crescente e derivável (como $CFA3$). O método proposto é baseado em condições necessárias para otimalidade local.

Seja f^0 qualquer fluxo multiproduto na rede, considere um arco i por onde passa um fluxo $f_i^0 > 0$. Defina o comprimento do arco da rede $l_i > 0$ da seguinte maneira:

$$\begin{cases} l_i = \infty \\ l_j = \Phi_j(f_i^0 + f_j^0) - \Phi_j(f_j^0), \quad \forall j \neq i. \end{cases}$$

Definindo $\zeta(f^0, i)$ como sendo o comprimento de um caminho de comprimento mínimo unindo as extremidades do arco i na rede, $\zeta(f^0, i)$ pode ser interpretado como sendo o custo mínimo extra para rotear todo o fluxo que passa pelo arco i por um caminho alternativo sem bifurcação na rede. A redução de custo resultante deste novo roteamento é definida como sendo o custo $\Delta(i)$.

Teorema 1. Uma condição necessária para f^0 ser uma solução de mínimo custo é:
$\forall$ arco da rede i tal que $f_i^0 > 0$, $\Delta(i) = \zeta(f^0, i) - \Phi_i(f_i^0) \geq 0$.

Esse teorema é o suporte para a formulação do algoritmo guloso proposto por Minoux. Em uma iteração k, f_k é uma solução viável, todas as diferenças $\Delta(i)$ são calculadas (para todo arco i que possua um fluxo positivo $f_k^i > 0$) e o arco que possui o menor valor de Δ ($\Delta < 0$) é eliminado. Um novo roteamento é feito obtendo-se uma

nova solução f_{k+1} com um custo menor. O algoritmo interrompe quando o menor Δ for não negativo. Nesse caso, as condições necessárias do teorema são satisfeitas.

Algoritmo de separação e avaliação

Passo 1- Seja f_0 uma solução inicial viável, $t \leftarrow 0$.

Passo 2- Na iteração it, seja f^{it} a solução corrente.

Passo 3- $\forall i = 1, ..., m$, tal que $f_i^{it} > 0$ calcule $\Delta^{it}(i) = \zeta(f^{it}, i) - \Phi_i(f_i^{it})$.

Passo 4- Determine $\overline{i}$, tal que $\Delta^{it}(\overline{i}) = \min \Delta^{it}(i), \ \ \forall i = 1, ..., m, \ \Phi_i(f_i^{it}) > 0$.

Passo 5- Se $\Delta^{it}(\overline{i}) \geq 0$, pare. Caso contrário, faça Λ^* ser o comprimento do menor caminho que liga os extremos de i obtidos no **Passo 2**. Seja:

$$\begin{cases} f_i^{it+1} \leftarrow f_i^{it} & i \neq \overline{i} \\ f_i^{it+1} \leftarrow f_i^{it} + f_{\overline{i}}^{it} & i \in \Lambda^* \\ f_{\overline{i}}^{it} \leftarrow 0 \end{cases}$$

Faça $it \leftarrow it + 1$ e volte ao Passo 2.

Algoritmo FD (Flow Deviation)

Gerla [29] propõe e demonstra um teorema que permite caracterizar soluções ótimas locais do problema $[CFA1]$.

Teorema 2. As condições necessárias e suficientes para que $(\bar{f}, \bar{c})$ seja uma solução ótima local de $[CFA1]$ são:

- $\bar{f}$ *e* $\bar{c}$ *viáveis* $(\bar{f}^i \leq \bar{c}^i, \ \forall i = 1, ..., m)$.
- $T(\bar{f}, \bar{c}) = T_{max}$.
- $\bar{f}$ *minimiza* $T(f, \bar{c})$.
- $\bar{c}$ *minimiza* $\varphi(c)$ *sujeito a* $T(\bar{f}, c) \leq T_{max}$.

Partindo desse teorema, foi desenvolvido um algoritmo genérico para determinação de ótimos locais de problemas com função objetivo contínua estritamente crescente e diferenciável.

Algoritmo FD Genérico

Passo 1- Seja (f^0, c^0) uma solução viável do problema CFA.
Seja $\varphi_0 = \varphi(c^0)$.
Seja $it = 0$.

Passo 2- Determine $T(f^{it+1}, c^{it}) = \min_f T(f, c^{it})$ onde f satisfaz as demais restrições do problema.

Passo 3- Determine $\varphi^{it+1} \leq \varphi(c^{it+1}) = \min_c \varphi(c)$, satisfazendo $T(f^{it+1}, c) \leq T_{max}$ e as demais restrições do problema.

Passo 4- Se $(\varphi^{it+1} - \varphi^{it}) < \sigma$, onde $\sigma > 0$ é a precisão desejada, interrompa: $T(f^{it+1}, c^{it+1})$ é um ótimo local. Senão faça $it = it + 1$ e retorne ao **Passo 2**.

Nesse algoritmo, o passo (2) é um problema de roteamento, e o passo (3) é um problema de atribuição de capacidades. A seqüência de φ_{it} é monotonicamente não crescente limitada inferiormente e desta maneira converge para um limite finito. Como conseqüência, a seqüência (f^{it}, c^{it}) converge para um limite $(\bar{f}, \bar{c})$, o qual pelos passos (2) e (3) do algoritmo e pelo teorema proposto é um ótimo local. Esse algoritmo pode ser aplicado para problemas com diferentes funções objetivo desde que elas sejam contínuas e estritamente crescentes. De fato, a idéia de resolver ora o problema de atribuição de capacidades ora o problema de roteamento de fluxos é recorrente entre os diversos algoritmos existentes para resolver o problema $CFA1$.

No seu trabalho, Gerla [21][29] adota esse algoritmo para resolver o problema $CFA1$ e, onde os custos de instalação de capacidades são lineares ou côncavos. Para isso, Gerla propõe o seguinte algoritmo:

Algoritmo FD

Passo 1- Partindo de um roteamento qualquer viável f^0, calcule a atribuição de capacidade ótima c^0 com $f = f^0$.
$(\varphi(c^0) = min\ \varphi(c),\ sujeito\ a\ T(f^0, c) \leq T_{max})$.
Faça $\varphi^0 = \varphi(c^0)$.
Faça $it = 0$.

Passo 2- Seja $\varphi_l^{it}(f)$ o custo da atribuição ótima de capacidades, como uma função

do fluxo, para o problema linearizado em torno de $c = c^0$. Seja f^{it+1} o roteamento de fluxos de caminho mínimo correspondente à métrica $l^{it} = \frac{\partial \varphi^{it}}{\partial f^{it}}$.

Passo 3- Seja c^{it+1} = a atribuição ótima de capacidades em f^{it+1}, e $\varphi^{it+1} = \varphi(c^{it+1})$.

Passo 4- Se $(\varphi^{it} - \varphi^{it+1}) < \epsilon$, onde $\epsilon > 0$ é um erro permitido, interrompa: (f^{it+1}, c^{it+1}) é um ótimo local. Senão faça $it = it + 1$ e retorne ao **Passo 1**.

A convergência desse algoritmo é garantida pelo fato de que existe um número finito de caminhos mínimos, e repetições da mesma solução não são possíveis, uma vez que φ_{it} é estritamente decrescente.

Abordagem alternativa de solução do problema $CFA1$

A restrição no máximo atraso médio de congestionamento da rede $T(f, c) \leq T_{max}$ é não linear tanto na variável de capacidade quanto na variável de fluxo. Dutta e Lim em [16] propõem uma maneira de reformular essa restrição de forma a obter uma restrição linear. Eles observaram que boa parte dos algoritmos propostos procuravam encontrar soluções que permitissem a utilização uniforme dos canais de comunicação. Ou seja, os algoritmos procuravam evitar que na rede houvesse canais sobre ou sub-utilizados. Baseados neste fato, Dutta e Lim reformularam a restrição de atraso médio da rede em uma restrição em função da taxa de utilização do canal distribuindo os requisitos de atraso por todos os arcos.

$$T(f, c) \leq T_{max} \tag{2.2.1}$$

$$T(f, c) = \frac{1}{\Gamma} \sum_{i=1}^{m} \frac{f_i}{c_i - f_i} \leq T_{max} \tag{2.2.2}$$

$$\frac{f_i}{c_i - f_i} \leq \frac{\Gamma T_{max}}{|\acute{A}|}, \forall i = 1, ..., m \tag{2.2.3}$$

ou equivalentemente,

$$\Psi f_i \leq c_i, \forall i = 1, ..., m \tag{2.2.4}$$

onde Ψ é uma constante que define um fator de utilização máxima do canal, e $\acute{A}$ é o conjunto de todos os canais ativos.

$$\Psi = \frac{|\acute{A}|}{\Gamma T_{max}} + 1 \tag{2.2.5}$$

onde $|\acute{A}|$ denota o número de arcos ativos na rede.

Dutta e Lim mostram em seu trabalho [16] que a adoção dessa formulação que considera uma utilização uniforme dos canais causa um pequeno excesso de capacidade. Os resultados obtidos indicam que o atraso médio da rede obtido nos seus experimentos era até 10% inferior ao atraso máximo da rede. Como a restrição obtida é linear, a adoção dessa abordagem facilita a solução do problema $CFA1$. O problema pode, então, ser resolvido utilizando técnicas de solução de problemas côncavos com restrições lineares.

Métodos de otimização global

O problema de rede com função objetivo côncava é um dos problemas mais difíceis da programação matemática, para o qual nenhum método de solução eficiente foi descoberto ainda, apesar de grandes esforços [32] [31] [65].

Em princípio, a dificuldade dos problemas de otimização global se deve ao fato de que muitos ótimos locais podem ocorrer. A limitação dos métodos de otimização local é que eles convergem para um ótimo local e interrompem a busca. Isto se eles não convergirem para um ponto estacionário para o qual não é nem mesmo possível garantir otimalidade local. A existência de múltiplos ótimos locais é uma dificuldade inerente aos problemas de otimização global que é conhecida há muito tempo. Garey, Graham e Johnson [23] demonstram que mesmo o caso particular de minimizar uma função quadrática côncava sobre um hipercubo unitário é um problema NP-difícil. Problemas de otimização global são em geral NP-difíceis.

A primeira proposta para resolver o problema geral de programação côncava foi feita por H.Tuy [37]. Esse algoritmo tem como idéia central um procedimento de corte de certas regiões do poliedro (gerado pelas restrições) nas quais o valor da função objetivo é certamente pior que o valor atualmente conhecido (cortes de concavidade).

Falk e Hoffman [18] propuseram uma abordagem diferente que pode ser interpretada como uma abordagem dual do método de Tuy. O algoritmo proposto utiliza

o conceito de envelope convexo da função objetivo e gera uma seqüência de aproximações convexas do poliedro original. Cada nova faceta gerada no poliedro de aproximação é uma faceta do poliedro original.

Em linhas gerais, esses procedimentos de otimização global enumeram vértices ou facetas do poliedro gerado pelas restrições do problema. Eles realizam uma busca em um conjunto finito de pontos que necessariamente contém o ótimo global do problema. Esses algoritmos possuem convergência finita para o ótimo global do problema, entretanto, do ponto de vista computacional, como são basicamente algoritmos de enumeração, são ineficientes e resolvem apenas problemas pequenos. Exemplos de aplicação desses métodos na solução do problema $CFA3$ podem ser encontrados em [39] e [58].

2.2.2 Métodos de solução para problemas com capacidades discretas

Quando o conjunto de capacidades assume valores discretos $c_i \in C_i$, o problema de atribuição de capacidades e roteamento de fluxos passa a ser um problema com variáveis mistas, cuja solução é muito difícil. Infelizmente, a presença de variáveis discretas impede a caracterização de ótimos locais partindo de um conceito de vizinhança.

Três heurísticas baseadas na solução do problema de roteamento

Na ausência de um conceito "natural"de vizinhança capaz de definir um critério de parada ou uma estratégia de busca, a solução é definir uma vizinhança. M.Gerla define em [29] uma vizinhança para o caso discreto:

$$(\bar{f},\bar{c}) \textit{ é um mínimo local} \iff \begin{array}{l} (\bar{f},\bar{c}) \textit{ é viável}, \\ \bar{f} \textit{ minimiza } T(f,\bar{c}), \\ \bar{c} \textit{ minimiza } \varphi(\bar{f},c), \textit{ sujeito a } T(\bar{f},c) \leq T_{max}. \end{array}$$

Com base nessa definição, foi proposta a seguinte heurística:

Algoritmo - DisCap

Passo 1- Seja $it = 0$, faça a interpolação dos custos discretos com uma função contínua do tipo "*continuous power law*"$(\theta_i(c_i) = \kappa_i c_i^{\alpha})$.

Passo 2- Defina o comprimento de cada arco i da rede de maneira aleatória (l_i).

Passo 3- Calcule o roteamento (f^0) que segue os caminhos mais curtos de acordo com esta métrica.

Passo 4- Aplique o algoritmo FD no problema contínuo obtido e determine um ótimo local.

Passo 5- Mantendo o roteamento fixo, resolva o problema discreto de atribuição de capacidades.

Passo 6- Mantendo as capacidades fixas, resolva o problema de roteamento.

Passo 7- Armazene o resultado obtido e faça $it = it + 1$.

Passo 8- Enquanto $it \leq it_{max}$ e enquanto houver melhora da solução, retorne ao **Passo 5**.

Passo 9- Selecione o menor mínimo local obtido até o momento.

DisCap converge, a cada iteração (1 - 8), para um ótimo local, que depende da solução inicial f^0. Depois de executar esse algoritmo diversas vezes partindo de diferentes pontos iniciais, interrompa e escolha a melhor solução obtida até o momento como solução final.

A segunda heurística proposta por Gerla trabalha da seguinte forma: inicialmente as menores capacidades disponíveis são atribuídas para os arcos da rede, e então o tráfego na rede é maximizado até que ocorra a saturação de algum arco. Neste momento, a capacidade do arco mais saturado deve ser aumentada. O processo é repetido até que o tráfego da demanda total seja satisfeito. Gerla denomina esta heurística de "*Top Down*".

A terceira heurística proposta "*Down Up*"é uma variação da anterior. Começando com a máxima capacidade disponível atribuída a cada um dos arcos alternativamente, reduza a capacidade de algum arco da rede e maximize a saída da rede. Repita esse processo enquanto o custo diminuir e as restrições forem satisfeitas. Em cada iteração, o arco que perde capacidade é o arco que minimiza um índice de saturação preestabelecido.

Método de Decomposição de Benders

A decomposição de Benders [44] é normalmente utilizada para resolver problemas com variáveis mistas ou que possuam uma estrutura que possa ser decomposta. Boyer [12] resolve o problema de atribuição de capacidades e roteamento de fluxos [$DCFA2$], aplicando a decomposição generalizada de Benders [28][44]. Na decomposição realizada, as variáveis inteiras binárias representam as alternativas de escolha das capacidades de cada arco. A utilização deste método foi motivada pelo fato de que uma vez fixadas as variáveis inteiras(capacidades nos arcos), os subproblemas obtidos são problemas de roteamento multiproduto, com função objetivo convexa, para os quais existem algoritmos de solução eficientes [56]. Contudo, o problema mestre é um problema linear com variáveis inteiras cuja solução não contribui para a eficiência do algoritmo.

Método de Relaxação Lagrangeana

Gavish e Neuman [26] propuseram o problema com capacidades discretas, em que o atraso de sinal é tratado como parte da função objetivo a ser minimizada $DCFA3$. Outros autores também adotaram a mesma formulação e técnicas similares de relaxação Lagrangeana para encontrar soluções. Bons limites inferiores e superiores foram obtidos para o valor da função objetivo [26][24][4][5].

O custo de uma unidade de atraso é assumido como sendo conhecido, e a função objetivo é como sendo a soma dos custos: de instalação das capacidades, custos relacionados ao atraso de congestionamento da rede e custos proporcionais ao uso dos canais de comunicação.

Para cada arco, a capacidade era escolhida entre um conjunto finito de capacidades disponíveis. Os custos associados a cada tipo de capacidade se dividem entre um custo fixo de instalação da capacidade no arco e um custo variável proporcional ao fluxo no arco. O método de solução proposto estabelecia um roteamento sem bifurcação, fato que diminui o leque de alternativas do problema.

O algoritmo de solução proposto é baseado no método de relaxação de Lagrange. A relaxação se efetua nas restrições de capacidade (ou seja, sobre as restrições que são responsáveis pelo acoplamento entre as variáveis de fluxo e as variáveis de capacidade). Assim sendo, a relaxação se efetua sobre a restrição:

$$\sum_{r \in \Upsilon} \delta_{ri} y_r^k f_i \leq c_i, \qquad \forall k = 1, ..., K \;\; \forall i = 1, ..., m$$

O algoritmo trabalha da seguinte forma: primeiro ele relaxa esta restrição, e o Lagrangeano é construído; a seguir, o método do subgradiente é utilizado para melhorar a qualidade do limite inferior. As soluções do problema de Lagrange obtidas a cada uma das iterações do método do subgradiente são usadas como uma base para gerar soluções viáveis do problema original.

O limite superior é obtido gerando uma seqüência de soluções viáveis a partir da solução do problema de Lagrange. Originalmente o trabalho feito por Gavish e Neuman não considerava todas as rotas possíveis entre um par origem-destino. Isto compromete os "limites inferiores"obtidos. Tais limites não correspondem a limites inferiores reais para o problema [$DCFA3$]. Essa falha foi corrigida por Gavish e Altinkemer em [24]. Este foi um dos primeiros a propor limites inferiores consistentes para o problema com capacidades discretas. Alguns outros trabalhos seguiram esta mesma linha [4].

Programação Dinâmica

Ng e Hoang em [55] estudaram um caso especial do problema de atribuição de capacidades e fluxo em que a capacidade de cada arco é igual a um múltiplo de um nível de capacidade pre-especificado que correspondia a algum tipo particular de canal de comunicação. Eles formularam o problema considerando as capacidades variando continuamente e usaram o método de desvios de fluxo (*flow deviation*) para resolver esse problema. Um procedimento de programação dinâmica foi então .usado para obter um solução discreta do problema.

Capítulo 3

Algoritmos com Desempenho Garantido para o Problema de Atribuição de Capacidades e Fluxos

3.1 Introdução

Diversas heurísticas foram desenvolvidas para resolver versões contínuas (côncavas) e discretas do problema CFA [30], [52], [39]. Abordagens heurísticas são comuns, uma vez que o problema é NP-difícil. Os algoritmos propostos nesse capítulo seguem essa mesma linha.

Neste capítulo são propostos dois algoritmos heurísticos com desempenho garantido para a solução de uma formulação proposta em [49]. A separabilidade da função objetivo é usada na obtenção de uma função convexa aproximada com vão de aproximação calculado explicitamente. O roteamento dos fluxos na rede é considerado como sendo bifurcado e os limites para a otimização global do problema são utilizados para garantir a qualidade dos resultados obtidos pelos algoritmos. A atribuição de capacidades é feita implicitamente. A aplicação trata da atribuição de capacidades e roteamento feitos simultaneamente de forma a garantir um nível aceitável de desempenho a um mínimo custo total. O desempenho e a eficiência dos algoritmos propostos são verificados através de experimentos numéricos.

Complexidade adicional ocorre quando as mensagens na rede seguem rotas estáticas, não bifurcadas, como foi estudado por diversos autores [26], [24],[3], [5]. Essa situação será estudada no capítulo 4.

3.2 O problema de atribuição de capacidades e roteamento

Esta seção apresenta uma extensão de uma formulação integrada para o problema de atribuição de capacidades e roteamento [49]. A rede de comunicação é modelada como um grafo $G = (V, A)$, onde V é o conjunto de nós da rede (vértices do grafo, ($|V| = n$), e A é o conjunto de arcos (arestas do grafo, ($|A| = m$)) que podem ser dirigidos ou não-dirigidos. Qualquer tipo de tráfego entre um dado par de nós $(O_k, D_k)_k$ é tratado como um produto separado k com um parâmetro de demanda d_k, onde K é o número de produtos circulando na rede. Seja P_{kh}, $h = (1, ..., N_k)$ um conjunto de N_k caminhos utilizados pelo produto k entre O_k e D_k em G. Esse conjunto pode ser o conjunto de todos os caminhos entre O_k e D_k ou um subconjunto restrito de caminhos viáveis. Seja a variável x_{kh} que informa o fluxo do produto k passando através do caminho h e seja o parâmentro a^{kh} um vetor de incidências arco-caminho $m-$dimensional, com cada componente a_i^{kh} definido por:

$$a_i^{kh} = \begin{cases} 1 & \text{se o arco } i \in P_{kh}, \\ 0 & \text{caso contrário.} \end{cases}$$

O problema de atribuição de capacidades e roteamento é formulado como:

$$\text{minimizar} \quad \sum_{i=1}^{m} \tau_i(f_i) \tag{3.2.1}$$

$$\text{sujeito a} \quad \sum_{k=1}^{K} \sum_{h=1}^{N_k} a_i^{kh} x_{kh} = f_i, \quad \forall i = 1, ..., m \tag{3.2.2}$$

$$\sum_{h=1}^{N_k} x_{kh} = d_k, \quad k = 1, ..., K \tag{3.2.3}$$

$$0 \leq f_i \leq c_i^{n_c}, \quad \forall i = 1, ..., m \tag{3.2.4}$$

$$x_{hk} \in \Re^+, \quad k = 1, ..., K, h = 1, ..., N_k \tag{3.2.5}$$

Cada componente f_i do vetor $m-$dimensional f denota o fluxo total no arco i. $\tau_i(f_i)$ é uma função crescente em $[0, c_i^{n_c})$, onde $c_i^{n_c}$ é o valor da máxima capacidade disponível para instalação no arco i. Nos experimentos realizados, todos os canais das redes são "*full-duplex*", ou seja, podendo admitir fluxo nos dois sentidos . Cada

$\tau_i(f_i)$ é tida como contínua, definida em R com valores nos números reais estendidos $R \cup \{+\infty\}$ e minorada por pelo menos uma função afim.

As restrições (3.2.2) impõem que o fluxo total no arco i seja igual à soma do fluxo de cada produto que utiliza esse arco. As restrições (3.2.3) garantem que a demanda d_k de cada produto k é satisfeita. As restrições (3.2.4) garantem que o fluxo total no arco i é menor que a máxima capacidade disponível para ser atribuída a esse arco.

A função de custos (3.2.1) do modelo proposto é tida como separável em relação aos arcos. Essa é uma função que integra custos operacionais, custos relacionados com a qualidade de serviço e custos de instalação de capacidades. Os custos relacionados com a qualidade de serviço podem ser formulados como funções crescentes que mensuram, por exemplo, o custo devido ao congestionamento em cada arco [43], a probabilidade de estouro de pilha no roteador (*buffer overflow*), ou probabilidade de bloqueio de chamadas [40]. Caso a rede não esteja saturada, o custo incremental de se mandar um pacote adicional é essencialmente zero. Entretanto, se a rede está congestionada, existe um custo social de se enviar novos pacotes, uma vez que o tempo de espera dos demais usuários da rede vai se deteriorar. A escolha da função de custo no arco não pretende representar um modelo específico de tráfego na rede, mas deve preservar as características gerais de uma grande família de medidas de qualidade e deve ser suficientemente simples para ser explicitamente calculada. A função de custos integrados $\tau_i(f_i)$ no arco i é definida como:

$$\tau_i(f_i) = \min\{T_i(f_i, c_i^0) + v_i^0 f_i + \pi_i^0, ...,\ T_i(f_i, c_i^j) + v_i^j f_i + \pi_i^j, \\ ..., T_i(f_i, c_i^{nc}) + v_i^{nc} f_i + \pi_i^{nc}\}. \qquad (3.2.6)$$

onde $[c_i^0, c_i^1, ..., c_i^{n_c}]$ é um conjunto de capacidade disponíveis ($c_i^{n_c} >, c_i^{n_c-1} > ..., > c_i^1 > c_i^0$), que estão relacionadas, em termos de valores monetários por unidade de tempo ($\$/mês$), com as seguintes funções de custo:

$T_i(f_i, c_i^j)$ custo relacionado com a qualidade de serviço no arco i para uma linha de tipo j;

$v_i^j f_i$ custo operacional, onde o custo unitário v_i^j é calculado para cada *kbps* passando pelo arco i;

π_i^j custo da atribuição da capacidade do tipo j ao arco i.

Este novo modelo apresenta uma abordagem unificada com formulação contínua para a atribuição discreta de capacidades e roteamento. O problema de atribuição de capacidades e roteamento admite que a topologia da rede seja conhecida e que em cada arco é instalada uma capacidade capaz de suportar a carga padrão de tráfego.

Como os aspectos topológicos do problema não são considerados, $\pi_i^0 = 0$, para todo arco $i \in A$.

Para cada arco i, é atribuída a capacidade c_i^j, e a função $T_i(f_i, c_i^j)$ deve ser convexa própria e diferenciável. As derivadas $T_i^{'}(f_i, c_i^j)$ são crescentes em f_i. As seguintes hipóteses são assumidas $\forall i = 1, ..., m$, $\forall j = 1, ..., n_c$:

$$T_i(0, c_i^0) = T_i(0, c_i^j) = T_i(0, c_i^{n_c}) = 0, \quad \forall i = 1, ..., m; \tag{3.2.7}$$

$$T_i(f_i, c_i^{n_c}) < T_i(f_i, c_i^j) < T_i(f_i, c_i^0), \forall f_i \in (0, c_i^{n_c}), \forall i = 1, ..., m; \tag{3.2.8}$$

$$T_i^{'}(f_i, c_i^{n_c}) + v_i^{n_c} < T_i^{'}(f_i, c_i^0) + v_i^0, \quad \forall f_i \in (0, c_i^{n_c}), \quad \forall i = 1, ..., m; \tag{3.2.9}$$

$$T_i^{'}(c_i^j, c_i^j) \geq M, \quad \forall j = 0, ..., n_c, \quad \forall i = 1, ..., m. \tag{3.2.10}$$

A relação (3.2.7) trivialmente postula que, para qualquer arco i, um fluxo nulo leva a um custo nulo de qualidade de serviço para qualquer capacidade atribuída c_i^j.

As relações (3.2.8) afirmam que, para um dado fluxo f_i no arco i, o custo de qualidade diminui à medida que a capacidade do arco aumenta.

As relações (3.2.9) mostram que o aumento da derivada induz um aumento nos custos marginais.

As relações (3.2.10) indicam que os custos de qualidade aumentam com o fluxo, forçando que a derivada no ponto de saturação seja maior que um valor M.

A Figura 3.1 ilustra a função integrada de custos no arco, denominada $\tau(f)$, omitindo temporariamente o índice do arco i. Essa função é o ínfimo de uma série de funções convexas. γ_i^j é considerado como sendo a porcentagem da capacidade c_i^j para a qual é mais econômico aumentar a capacidade do arco de c_i^j para c_i^{j+1}. A solução deste modelo é obtida através da busca de caminhos para os fluxos de produtos, do

fluxo do produto no arco e implicitamente das capacidades dos arcos para minimizar o custo total. A estrutura de custos objetiva distribuir a carga entre todos os arcos capacitados reduzindo o custo total de alugar ou comprar as capacidades e os custos de qualidade expressos em valores monetários.

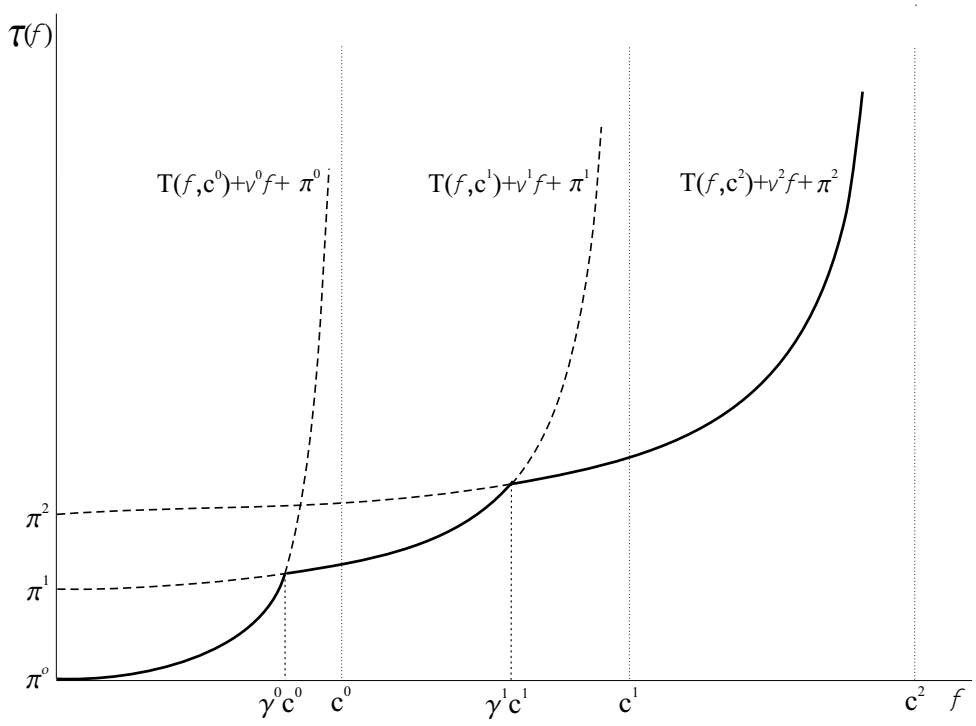


Figura 3.1: Função de custos integrada $\tau(f)$

3.3 Aproximação convexa da formulação proposta

A função objetivo definida em (3.2.1) e (3.2.6) gera um problema de fluxos multiproduto não-convexos. A não-convexidade é uma característica inerente do problema de decisão de capacidades. Algoritmos heurísticos são normalmente propostos na literatura para resolver problemas dessa natureza. Este trabalho segue essa tendência, mas os algoritmos propostos são algoritmos heurísticos com desempenho garantido, no sentido de que um erro máximo com relação ao ótimo global é conhecido.

Seja F um conjunto convexo não vazio de vetores de fluxos nos arcos para os quais fluxos multiproduto viáveis são conhecidos, isto é, existem $f \in F$ e os correspondentes fluxos nos caminhos x_{kh} que satisfazem as restrições de (3.2.2) a (3.2.5). A casca

convexa fechada da função τ_i é denominada $conv(\tau_i)$ e é definida como sendo a maior função convexa fechada majorada por τ_i. A $conv(\tau_i)$ pode ser definida como sendo o supremo de todas as funções afins que minoram τ_i, isto é $conv(\tau_i(f_i)) = \sup\{s.f_i - b; s.y - b \leq \tau_i(y), \forall y \in \mathbf{R}\}$, onde o supremo é calculado sobre todo o intervalo (s, b).

O seguinte resultado sobre o vão de aproximação entre a casca convexa e a função original justifica a metodologia para o cálculo da casca convexa da função τ_i [17], [19], [49].

Proposição 3.3.1. *Suponha que cada função τ_i seja limitada inferiormente e que o problema 3.2.1-3.2.5 possua solução ótima f^* com valor ótimo ϕ^*.*
Então: $\ddot{\phi} = \inf_{f \in F}\{\sum_{i=1}^{m} conv\ \tau_i(f_i)\}$ é um limite inferior do valor ótimo, i.e. $\phi^* \geq \ddot{\phi}$. Mais ainda,

$$\phi^* - \ddot{\phi} \leq \sum_{i=1}^{m} \max_{f_i}[\tau_i(f_i) - conv\ \tau_i(f_i)] = \Delta. \tag{3.3.1}$$

Demonstração. Da definição de $\ddot{\phi}$, tem-se: $\ddot{\phi} \leq \sum_{i=1}^{m} conv\ \tau_i(f_i), \ \ \forall\ f \in F.$

Então, aplicando a definição da casca convexa de cada uma das funções de custo no arco, obtém-se $\ddot{\phi} \leq \sum_{i=1}^{m} \tau_i(f_i), \ \ \forall f \in F$, e assim $\ddot{\phi} \leq \phi^*$.

Seja $\ddot{f} \in F$ o vetor com componentes $\ddot{f}_i$ para cada arco i tal que $\ddot{\phi} = \sum_{i=1}^{m} conv\ \tau_i(\ddot{f}_i)$, então:

$$\begin{aligned} \phi^* - \ddot{\phi} &= \sum_{i=1}^{m} \tau_i(f_i^*) - \sum_{i=1}^{m} conv\ \tau_i(\ddot{f}_i) \leq \sum_{i=1}^{m} [\tau_i(\ddot{f}_i) - conv\ \tau_i(\ddot{f}_i)] \\ &\leq \sum_{i=1}^{m} \max_{f_i}[\tau_i(f_i) - conv\ \tau_i(f_i)] = \sum_{i=1}^{m} \Delta_i = \Delta \end{aligned} \tag{3.3.2}$$

□

O vão máximo Δ associado ao limite inferior acima é, em geral, maior do que zero, uma vez que a casca convexa da soma de um conjunto de funções é, em geral, diferente da soma das cascas convexas de cada função. Tal afirmação é verdadeira mesmo para as funções separáveis porque existem restrições de acoplamento [61]. A motivação para fazer isto vem do fato de que a casca convexa de certas funções unidimensionais é relativamente fácil de ser calculada explicitamente. A Figura 3.2 ilustra a casca convexa $conv(\tau_i)$ da função τ_i, omitindo temporariamente o índice do arco i.

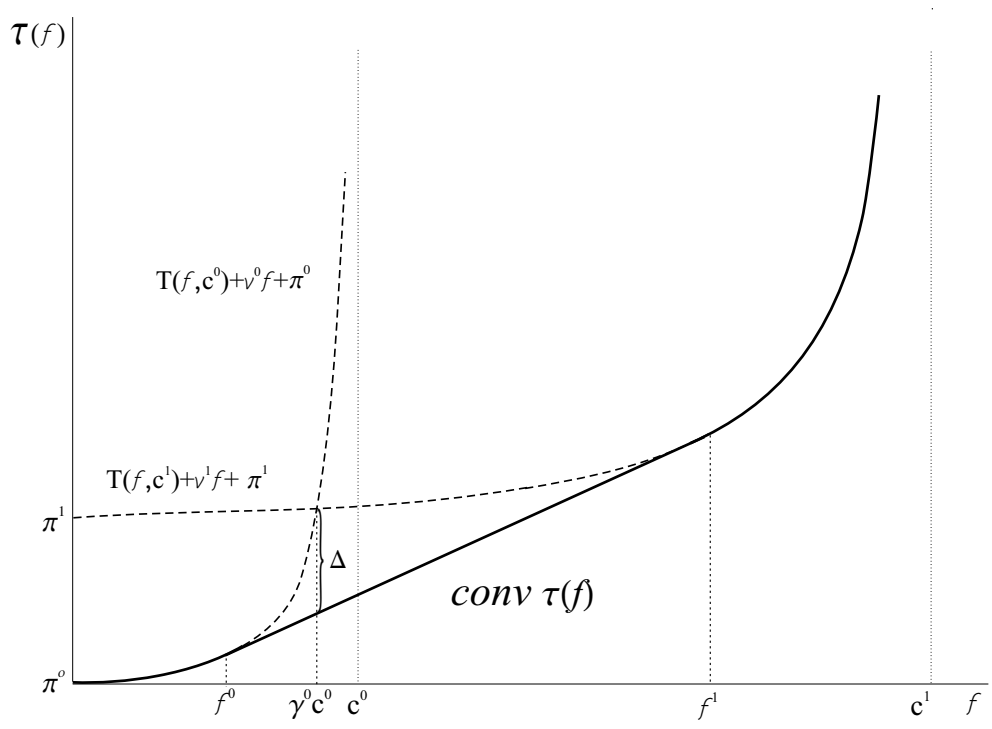


Figura 3.2: Casca convexa da função de custos integrada $conv(\tau)$

O problema de fluxos multiproduto obtido adotando a casca convexa de cada uma das funções de custo no arco pode ser resolvido com qualquer algoritmo apropriado para resolver um problema de fluxos multiproduto convexo [29] [56].

Suponha que não exista qualquer restrição de não-bifurcação dos fluxos ou qualquer tipo de restrição de roteamento. Nesse caso, solucionando o problema convexo, um limite inferior consistente é obtido para o problema original de atribuição de capacidades e roteamento. Por outro lado, um limite superior do problema pode ser obtido usando o Δ calculado em (3.3.2).

3.4 Algoritmo CFA com desempenho garantido

Duas heurísticas com desempenho garantido capazes de encontrar boas soluções do problema (3.2.1-3.2.5) foram desenvolvidas neste trabalho. Esses algoritmos heurísticos determinam inicialmente uma solução viável e gradualmente obtêm soluções menores até ficarem estagnadas. As heurísticas são compostas por duas fases: uma fase inicial comum na qual a aproximação convexa do problema não-convexo original é resolvida

e um limite inferior é obtido; e uma segunda fase, em que o roteamento resultante da primeira fase é adotado como solução inicial para algoritmos cíclicos que ora atribuem capacidades aos arcos, ora aplicam algoritmos de busca local até que a solução não mais apresente melhoria.

Na fase de aproximação convexa (Fase Inicial Comum) um algoritmo de roteamento é aplicado para encontrar o valor ótimo $\ddot{\phi}$ do problema convexo aproximado. O valor obtido $\ddot{\phi}$ é um limite inferior do problema (3.2.1-3.2.5). A função objetivo não-convexa original $\tilde{\phi}$ é avaliada para o roteamento obtido $\ddot{f}$. As proposições a seguir asseguram que as heurísticas propostas possuem desempenho garantido.

Definição 3.4.1. O desempenho garantido de uma heurística para um problema de minimização é α ($\alpha \geq 1$) se necessariamente o algoritmo fornece uma solução ϕ cujo valor é no máximo α vezes maior que o valor ótimo global: $\phi \leq \alpha\phi^*$

Relacionar diretamente ϕ a ϕ^* é difícil porque o valor ótimo global ϕ^* não é conhecido. Entretanto, a seguinte relação indireta pode ser obtida:

$$\phi \leq \alpha\ddot{\phi}$$
$$\ddot{\phi} \leq \phi^*$$
$$\phi \leq \alpha\phi^*$$

Proposição 3.4.1. *Suponha que $\ddot{f}$ seja a solução ótima do problema convexo aproximado. Então $\ddot{\phi} = \sum_{i=1}^{m} conv\ \tau_i(\ddot{f}_i) = \inf_{f \in F} \sum_{i=1}^{m} conv\ \tau_i(f_i)$. Seja $\tilde{\phi}$ o valor da função objetivo considerando o roteamento $\ddot{f}$ e a função objetivo original não-convexa: $\tilde{\phi} = \sum_{i=1}^{m} \tau_i(\ddot{f}_i)$. ϕ é a solução obtida aplicando uma das heurísticas propostas, deste modo resultando $\phi \leq \tilde{\phi}$, $\phi - \ddot{\phi} \leq \tilde{\phi} - \ddot{\phi} \leq \Delta$.*

Demonstração. Da definição de $\tilde{\phi}$:

$$\begin{aligned}
\sum_{i=1}^{m}[\tau_i(\ddot{f}_i) - conv\ \tau_i(\ddot{f}_i)] &\leq \sum_{i=1}^{m} \max_{f_i}[\tau_i(f_i) - conv\ \tau_i(f_i)] = \Delta \\
\sum_{i=1}^{m}[\tau_i(\ddot{f}_i) - conv\ \tau_i(\ddot{f}_i)] &\leq \Delta \\
\sum_{i=1}^{m}[\tau_i(\ddot{f}_i)] - \sum_{i=1}^{m}[conv\ \tau_i(\ddot{f}_i)] &\leq \Delta \\
\tilde{\phi} - \ddot{\phi} &\leq \Delta \\
\phi &\leq \tilde{\phi} \\
\therefore \phi - \ddot{\phi} \leq \tilde{\phi} - \ddot{\phi} &\leq \Delta \qquad (3.4.1)
\end{aligned}$$

□

Agora o parâmetro de desempenho α pode ser calculado como uma função de Δ e $\ddot{\phi}$.

Proposição 3.4.2. *As heurísticas propostas necessariamente retornam uma solução cujo valor é no máximo α vezes o valor do limite inferior $\phi \leq \alpha\ddot{\phi}$.*

Demonstração. De (3.4.1), tem-se:

$$\begin{aligned} \phi - \ddot{\phi} &\leq \Delta \\ \phi &\leq \ddot{\phi} + \Delta \\ \Delta &= \varrho\ddot{\phi} \\ \phi &\leq \ddot{\phi} + \varrho\ddot{\phi} \\ \phi &\leq (1+\varrho)\ddot{\phi} \\ \alpha &= 1 + \varrho = 1 + \Delta/\ddot{\phi} \\ \therefore \quad \phi &\leq \alpha\ddot{\phi} \end{aligned} \tag{3.4.2}$$

□

A diferença entre heurísticas com desempenho garantido e algoritmos de aproximação reside no fato de que esses precisam executar em tempo polinomial. Na programação não-linear, não existem métodos numéricos capazes de determinar soluções ótimas locais em tempo polinomial. Até mesmo verificar a otimalidade local de uma solução viável é um problema NP-Difícil [35].

O primeiro algoritmo é um método cíclico entre a atribuição de capacidades e a aplicação de um algoritmo de roteamento (de busca local). O segundo algoritmo baseia-se em uma estratégia de solução por partes (*piecewise strategy*) [62], com a aplicação direta do método de desvios de fluxos (*Flow Deviation method*) [29] na função objetivo não convexa (3.2.1) e (3.2.6).

Primeiro será apresentada a fase inicial comum às duas heurísticas:

Fase I: Fase de aproximação convexa comum aos dois algoritmos

Passo 1- Determine uma solução inicial viável f^0 para o problema, $t = 0$.

Passo 2- Aplique um algoritmo de roteamento para encontrar a solução ótima $\ddot{f}$ do problema convexo. O valor da solução obtida $\ddot{\phi}$ é um limite inferior do problema (3.2.1-3.2.5).

Passo 3- Calcule o valor da função objetivo do problema original $\tilde{\phi}$, considerando o vetor de fluxos $\ddot{f}$. Se $|\tilde{\phi} - \ddot{\phi}| < \varepsilon$ **pare** com $\ddot{f}$. Ou seja, uma solução global ótima do problema 3.2.1-3.2.5 foi encontrada com precisão ε. Senão, continue na **Fase II**.

Ambos os algoritmos utilizam a mesma fase inicial. Mas cada algoritmo executa sua própria segunda fase.

Primeiro Algoritmo - Método cíclico de melhoria

Fase II: Método cíclico de melhoria.

Passo 4- Partindo do roteamento obtido, para cada arco i atribua capacidade, aplicando as seguintes regras:
se $0 \leq f_i \leq \gamma_i^0 c_i^0$ faça $c_i = c_i^0$;
se $\gamma_i^j c_i^j \leq f_i < \gamma_i^{j+1} c_i^{j+1}$ faça $c_i = c_i^{j+1}$.

Passo 5- Um algoritmo de roteamento é aplicado no problema de fluxos multiproduto convexo resultante, e uma nova solução é obtida para o problema (3.2.1-3.2.5).

Passo 6- Se $|\phi(f^t) - \phi(f^{t+1})| < \varepsilon$ **pare**; senão, faça $t \leftarrow t+1$ e vá para o **Passo 4**.

Segundo Algoritmo - Solução por partes (*Piecewise Strategy*)

Fase II: Solução por partes.

Aplicação do algoritmo para resolver problemas de fluxos multiproduto em problemas parciais.

Passo 4- Partindo do roteamento obtido, para cada arco i, determine uma capacidade, aplicando as seguintes regras:
se $0 \leq f_i \leq \gamma_i^0 c_i^0$ faça $c_i = c_i^0$;
se $\gamma_i^j c_i^j \leq f_i < \gamma_i^{j+1} c_i^{j+1}$ faça $c_i = c_i^{j+1}$.

Passo 5- Adicione as seguintes restrições ao problema (3.2.1-3.2.5):
se $0 \leq f_i < \gamma_i^0 c_i^0$, adicione $f_i \leq \gamma_i^0 c_i^0 + \epsilon$;
se $\gamma_i^j c_i^j \leq f_i < \gamma_i^{j+1} c_i^{j+1}$, adicione $f_i \leq \gamma_i^{j+1} c_i^{j+1} + \epsilon$ e $f_i \geq \gamma_i^j c_i^j - \epsilon$.
Encontre uma solução para o problema convexo obtido.

Passo 6- Se não existem arcos com $|f_i - \gamma_i^j c_i^j| \leq \epsilon$, **pare** com f sendo uma solução aproximada. Senão, determine o conjunto E de arcos i com $|f_i - \gamma_i^j c_i^j| \leq \epsilon$.

Passo 7- Para cada $i \in E$ faça:
se $f_i \geq \gamma_i^j c_i^j$ e $c_i = c_i^j$ mude c_i para c_i^{j+1} e adicione a restrição $f_i \geq \gamma_i^j c_i^j - \epsilon$;
se $f_i \leq \gamma_i^j c_i^j$ e $c_i = c_i^{j+1}$ mude c_i para c_i^j e adicione a restrição $f_i \leq \gamma_i^j c_i^j + \epsilon$.
Encontre uma solução para o problema obtido.

Passo 8- Assuma a atribuição de capacidade do arco i para o qual foi obtida a maior diminuição da função objetivo ϕ no **Passo 7**.
Se $|\phi(f^t) - \phi(f^{t+1})| < \varepsilon$ **pare**; senão, faça $t \leftarrow t + 1$ e vá para o **Passo 7**.

Esses algoritmos geram seqüências de soluções viáveis decrescentes e limitadas inferiormente. Problemas-testes são resolvidos para mostrar que os algoritmos propostos são eficientes na obtenção de boas soluções.

3.5 Experimentos numéricos

Experimentos numéricos foram realizados para o estudo das heurísticas e o exame da influência dos diferentes parâmetros nos resultados obtidos.

Esses algoritmos foram codificados em C e compilados com o GCC 2.95.3 e foram executados em um computador CELERON, 500 MHz, 64 MegaBytes RAM, com o sistema operacional LINUX.

O método de desvios de fluxos foi adotado como o algoritmo de roteamento [56]. O método de desvios de fluxos é um método primal que foi desenvolvido para o problema de fluxos multiproduto por Fratta et al. [21] e tem sido utilizado por diversos autores [45]. Trata-se de um caso especial do método de Frank-Wolfe para resolver problemas

Tabela 3.1: Parâmetros e propriedades topológicas das diferentes redes

Rede ID	nós	arcos	pares-OD	grau médio do nó	profundidade média	diâmetro
	n	m	K	$2m/n$		
N5	5	6	10	2.40	1.4	2
N25	25	54	600	4.32	3.4	4
N50	50	217	2450	8.68	2.96	3
N100	100	962	2000	19.0	2.7	3
NTS100	100	187	2000	3.74	8.96	11

de otimização não-lineares com restrições lineares [29]. Uma de suas características é que a cada iteração um limite inferior do valor ótimo é obtido. O procedimento termina quando o valor do limite inferior corrente está suficientemente próximo do valor da solução. Nos experimentos realizados foi estabelecida uma precisão de 1% para o algoritmo de roteamento.

Cinco topologias foram utilizadas como casos experimentais. Elas estão representadas nas Figuras 3.3, 3.4, 3.5, 3.6 e 3.7. A topologia (N5) é muito simples, e o problema correspondente é resolvido até a otimalidade global por enumeração explícita. As demais topologias foram criadas utilizando um programa gerador de topologias de rede [69]. O código fonte utilizado está disponível gratuitamente em [69][15] (http://www.cc.gatech.edu/fac/Ellen.Zegura/graphs.h). Três grafos aleatórios planos (N25, N50, N100) e um grafo hierárquico (NTS100)(*transit-stub*) foram criados. O usuário especifica um conjunto de parâmetros como o número de nós e sua conectividade e o programa produz uma lista de nós e de arcos. Tais redes aleatórias pretendem representar redes reais de comunicação de dados mais precisamente que os modelos usados até então. A Tabela 3.1 apresenta os parâmetros topológicos dessas redes.

A distância entre dois nós é definida como sendo o menor caminho entre estes nós. O profundidade é a profundidade da árvore de caminhos mínimos de um nó raiz até todos os outros nós da rede. Os caminhos mínimos são calculados considerando que cada arco (aresta) possui comprimento unitário. O diâmetro da rede (do grafo) é definido como sendo a maior profundidade da rede.

Com relação ao número de produtos circulando pela rede, nas três primeiras topologias, cada par de nós se comunica. Para as topologias N100 e NTS100, $K =$ 2000 pares origem-destino foram criados aleatoriamente. Uma seção de comunicação

Tabela 3.2: Capacidades disponíveis e seus respectivos custos

Capacidade c_i^j $[kbps]$	Custo de instalação π_i^j $[\$/mês]$
64	150
128	250
256	390
384	480
512	570

está ativa para cada par origem-destino, e, cada seção gera uma taxa média de tráfego de $\Gamma_k = 1\ mensagem/s$. As diferentes capacidades disponíveis para instalação e seus correspondentes custos são apresentadas na Tabela 3.2.

Os custos de qualidade foram associados ao atraso sofrido pelos pacotes na rede. O modelo clássico de filas de espera do tipo $M/M/1$ é adotado (equações em 2.1.1, 2.1.2 e 2.1.3).

Nesse modelo, o atraso médio a que os pacotes são submetidos ao atravessarem a rede é dado pela equação 2.1.3. Essa função é separável nos arcos. Adotando uma constante de proporcionalidade denominada de custo de congestionamento ρ $[\$/mês/mensagem]$ o atraso médio passa a ser medido em unidades monetárias. A parcela de cada arco i no atraso médio é dada por:

$$T_i(f_i, c_i) = \rho \frac{f_i}{c_i - f_i} \tag{3.5.1}$$

A constante de proporcionalidade ou custo de congestionamento ρ é definida pelos requisitos dos usuários. Nos experimentos, foram assumidos os valores $\rho \in \{1, 5, 10, 50, 100, 500, 1000\}[\$/mês/mensagem]$. Um conjunto de possibilidades para a constante ρ deve ser avaliado para permitir a escolha do valor mais apropriado em uma aplicação real [26].

Além do modelo $M/M/1$, é possível usar modelos mais precisos permitindo que o tempo de serviço, que é uma função do tamanho das mensagens e da capacidade do arco, siga outra distribuição. O tamanho do "*buffer*"também pode ser considerado. A consideração destes efeitos aumenta a complexidade da formulação, mas a estrutura geral é preservada, e uma função de custos separável e convexa para cada capacidade ainda pode ser obtida [8], [40].

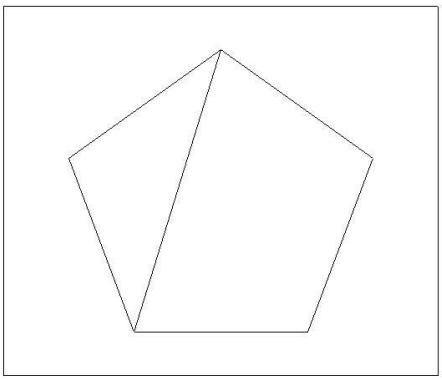


Figura 3.3: Topologia da rede N5

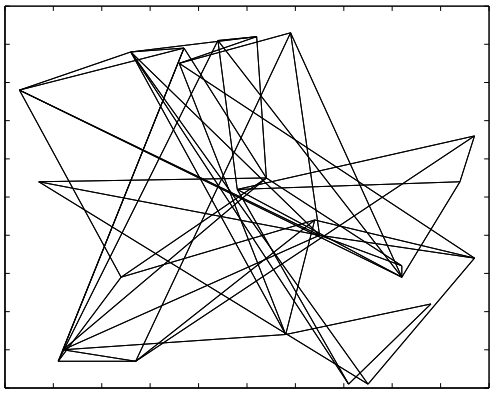


Figura 3.4: Topologia da rede N25

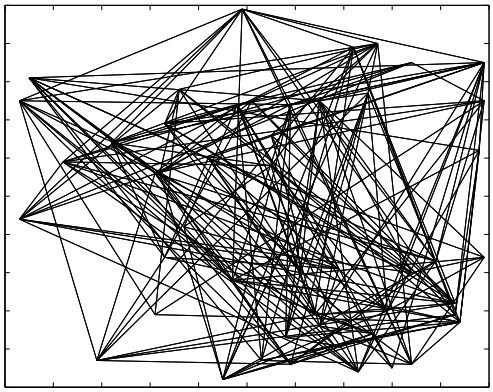


Figura 3.5: Topologia da rede N50

Figura 3.6: Topologia da rede N100

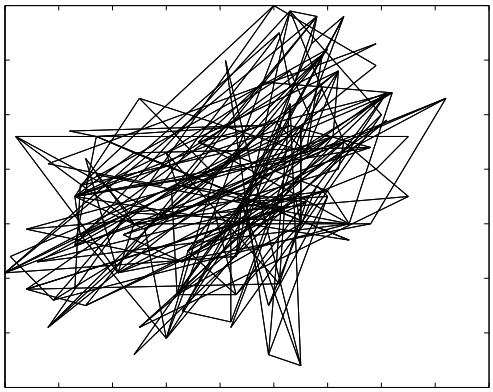


Figura 3.7: Topologia da rede NTS100

Os testes foram feitos para avaliar o desempenho dos algoritmos para diferentes parâmetros. Os melhores resultados obtidos podem ser observados nas Tabelas 3.3 e 3.4. O custo de congestionamento $\rho = 100\,\$/mês/mensagem$ e o tamanho de mensagem 10 $kbits$ foram adotados como caso base para os problemas N25, N50, N100 e NTS100. Para o problema N5, foi adotado um tamanho de mensagem de 40 $kbits$.

A relação de desempenho entre as soluções obtidas aplicando as heurísticas propostas e os limites inferiores obtidos a partir da convexificação dos problemas é definida como $\alpha_{sw} = \frac{\phi_{sw}}{\ddot{\phi}}$ e $\alpha_{pc} = \frac{\phi_{pc}}{\ddot{\phi}}$ para, respectivamente, o método cíclico de melhoria e o método de solução por partes. A relação de desempenho entre a solução ótima e os limites inferiores obtidos com a convexificação é definida por $\alpha^* = \frac{\phi^*}{\ddot{\phi}}$. O parâmetro de medida do pior caso teórico α, associado com a equação 3.4.2), é calculado como $\alpha = 1 + \Delta/\ddot{\phi}$.

As Figuras 3.8 e 3.9 comparam os desempenhos dos algoritmos heurísticos (α_{sw} e α_{pc}) com os desempenhos das soluções ótimas globais (α^*) do problema N5. Para

esse problema pode ser observado que as soluções heurísticas obtidas são próximas das soluções ótimas globais. Nos experimentos realizados, as piores soluções encontradas ocorrem com a rede N5, $\rho = 100\,\$/mês/mensagem$, comprimento de mensagem $= 40\ kbits$, $\alpha_{sw} = 1.37$ e $\alpha_{pc} = 1.37$, e com um atraso médio de mensagem de $0.904s$ em ambos os casos.

Note que na Figura 3.8, $\alpha^* \leq \alpha_{sw} \leq \alpha$ e $\alpha^* \leq \alpha_{pc} \leq \alpha$ e que as medidas de desempenho observadas (α_{sw} e α_{pc}) são significativamente menores do que os valores dos piores casos teóricos α. Desde que os valores dos ótimos globais para algumas instâncias do problema N5 são conhecidos, é possível verificar a qualidade das soluções heurísticas com relação às soluções ótimas globais para estes casos. Por exemplo, no caso N5, $\rho = 100\,/mês/mensagem$, para um tamanho de mensagem $= 1kbit$ os valores das heurísticas coincidem com os valores ótimos globais. Para um tamanho de mensagem $= 40\,kbits$ um erro de 6% foi observado entre a solução heurística e a solução ótima global. Essa instância também foi a que apresentou o maior erro, o que talvez se explique pelo menor número de alternativas de soluções viáveis.

A Figura 3.10 apresenta comparações entre os resultados obtidos com as duas heurísticas para diferentes tamanhos de mensagens e considerando o custo de congestionamento $\rho = 100\$/mês/mensagem$. Inicialmente, à medida que o tamanho da mensagem aumenta correspondendo a um aumento da carga na rede, a qualidade das soluções obtidas diminui ($\alpha_{sw} > 1$ e $\alpha_{pc} > 1$). Contudo, quando a carga da rede começa a se aproximar da capacidade máxima disponível, a qualidade das soluções obtidas melhora ($\alpha_{sw} \simeq 1$ e $\alpha_{pc} \simeq 1$). Esse efeito pode ser explicado observando a Figura 3.2. Quando o fluxo em um arco é menor que f_0 ou maior que f_1, a função de aproximação coincide com a função original. A Figura 3.10 também mostra que o parâmetro de pior caso teórico α diminui quando a carga da rede aumenta. Este efeito pode ser explicado observando que ambos os custos fixos e de congestionamento aumentam com a carga da rede (veja Figura 3.13) e que, com o aumento geral dos custos, é reduzida a importância relativa do vão máximo Δ, um parâmetro fixo que é calculado a priori.

A Figura 3.11 mostra comparações entre os resultados obtidos para diferentes custos de congestionamento e considera o tamanho das mensagens como $10\,kbits$ para as redes N25, N50, N100 e $1\,kbit$ para a rede NTS100. A qualidade das soluções obtidas foi afetada pela constante de proporcionalidade ρ. Nas redes N25, N50,

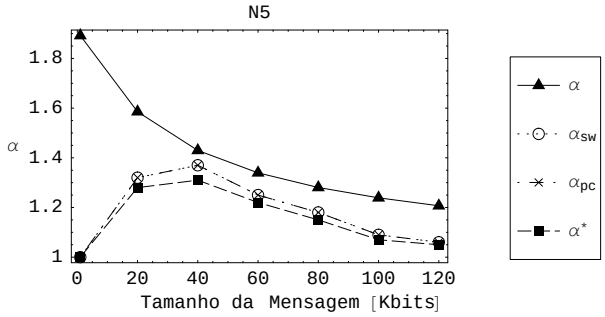


Figura 3.8: Comparação entre os desempenhos do ótimo global α^* (■), dos resultados heurísticos α_{pc} (×), α_{sw} (∘) e do pior caso teórico α (▲) para diferentes tamanhos de mensagens, e para $\rho = 100\,\$/mês/mensagem$ na rede(N5)

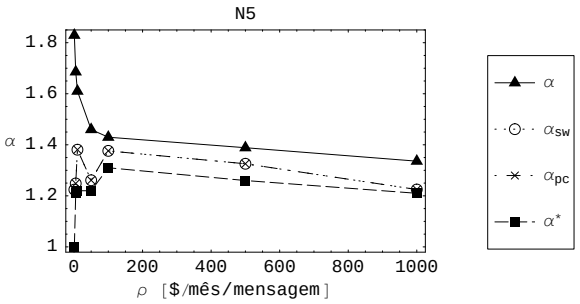
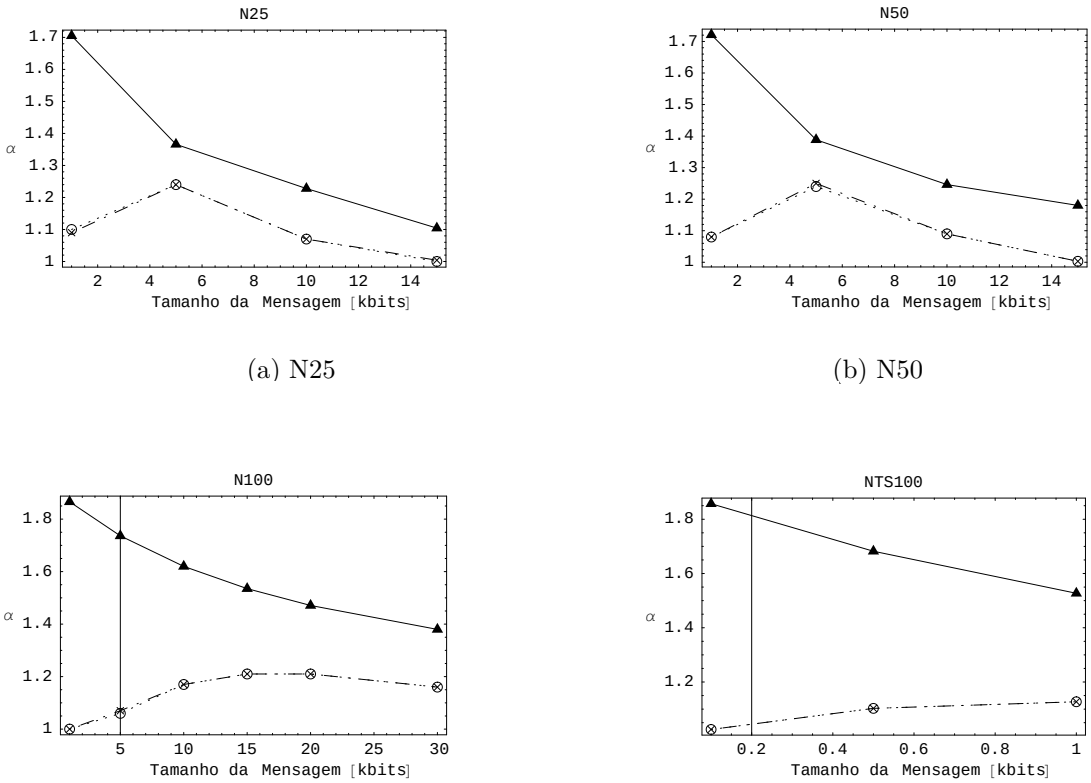


Figura 3.9: Comparação entre os desempenhos do ótimo global α^* (■) dos resultados heurísticos α_{pc} (×), e α_{sw} (∘) e do pior caso teórico α (▲) para diferentes constantes de proporcionalidade, e para mensagens de tamanho $40kbits$ na rede (N5)



(a) N25

(b) N50

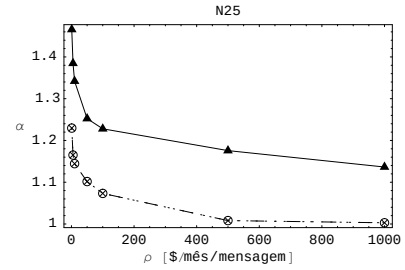
(c) N100

(d) NTS100

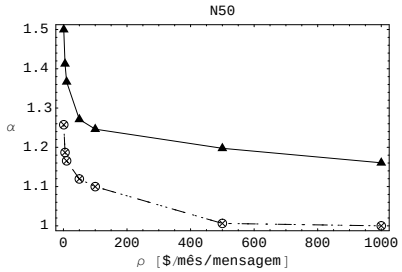
Figura 3.10: Comparação entre os resultados α (▲), α_{pc} (×), α_{sw} (∘) para diferentes tamanhos de mensagens

quando ρ aumenta, α_{sw} e α_{pc} diminuem. Para as redes N100, NTS100 quando $\rho \leq 500\,\$/mês/mensagem$ α_{sw} e α_{pc} aumentam e quando $\rho \geq 500\,\$/mês/mensagem$ α_{sw} e α_{pc} diminuem. Esses resultados podem ser explicados observando que, quando a carga da rede é pequena ou quando ela é próxima da capacidade máxima da rede, a aproximação convexa coincide com a função original. A Figura 3.11 também mostra que o parâmetro de pior caso teórico α tende a diminuir quando o custo de congestionamento da rede aumenta. Esse resultado pode ser explicado observando a relação entre o vão máximo Δ, o custo de congestionamento ρ e o custo total $\ddot{\phi}$. Por exemplo, na rede N50 para $\rho = 1\,\$/mês/mensagem$; $\Delta = 35\,317\$$ e $\ddot{\phi} = 41\,251\$$ então $\alpha = 1 + \frac{\Delta}{\ddot{\phi}} = 1.85$ e quando $\rho = 1000\,\$/mês/mensagem$; $\Delta = 46\,848\$$ e $\ddot{\phi} = 292\,803\$$ então $\alpha = 1 + \frac{\Delta}{\ddot{\phi}} = 1.16$.

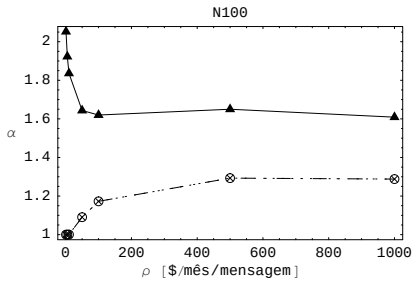
As Figuras 3.12 e 3.13 e as Tabelas 3.3 e 3.4 resumem os melhores resultados



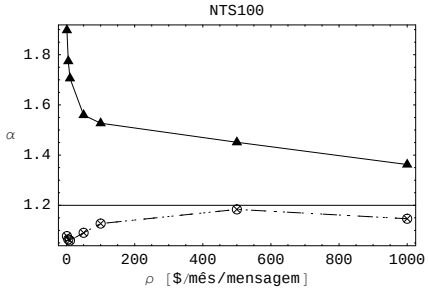
(a) N25



(b) N50



(c) N100



(d) NTS100

Figura 3.11: Comparação entre resultados α (▲), α_{pc} (×), α_{sw} (○) para diferentes parâmetros de custos de congestionamento

obtidos. Elas mostram o impacto de diferentes comprimentos de mensagens e de diferentes parâmetros de custo. A Figura 3.12 e a Tabela 3.3 mostram os resultados obtidos para diferentes tamanhos de mensagens. É importante ressaltar que os custos fixos são dominantes no custo total e que normalmente a taxa de crescimento do custo fixo é maior do que a taxa de crescimento dos custos de congestionamento. Uma exceção ocorre no caso N5, devido ao fato de a rede estar congestionada. A Figura 3.13 e a Tabela 3.4 mostram os resultados para diferentes custos de congestionamento. Elas mostram que os custos de congestionamento são fortemente influenciados pelo parâmetro ρ.

Como esperado, quando o custo de congestionamento ρ aumenta, o atraso médio na rede tende a diminuir conforme pode ser observado na Figura 3.15. Por outro

lado, os custos de congestionamento aumentam bastante como pode ser observado na Figura 3.13. Quando o parâmetro de custo de congestionamento aumenta, os custos de congestionamento são mais e mais importantes para os custos totais. Por exemplo, no caso N50, $\rho = 1\,\$/mês/mensagem$, comprimento de mensagem $10kbits$, e atraso médio de mensagem $0.885s$ o custo de congestionamento representa apenas 2.4% do custo total. Por outro lado,no caso N50, $\rho = 1000\,\$/mês/mensagem$, comprimento de mensagem $= 10kbits$, e atraso médio $= 0.0069s$, os custos de congestionamento representam 57% dos custos totais.

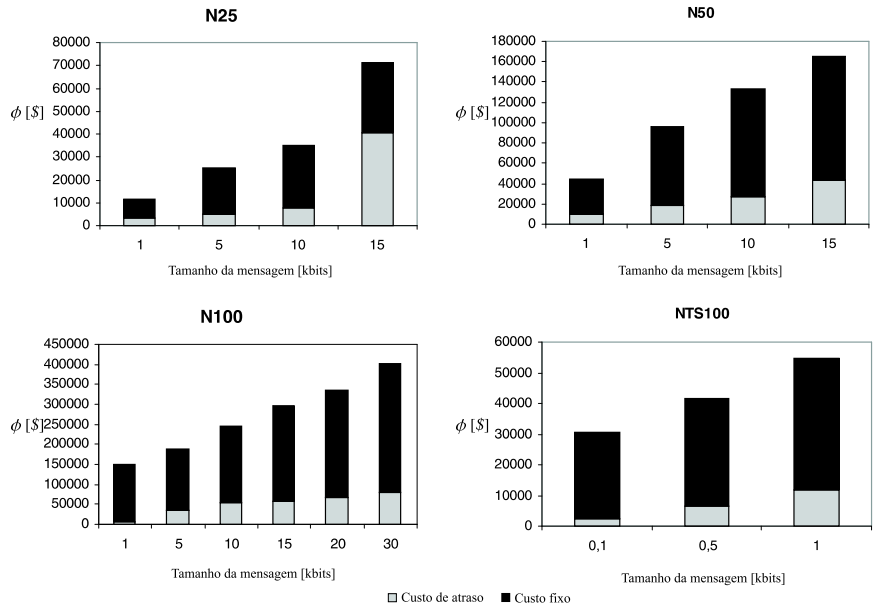


Figura 3.12: Custos de congestionamento e custos fixos para diferentes tamanhos de mensagens. O parâmetro de proporcionalidade foi fixado como sendo $\rho = 100\$/mês/mensagem$

As heurísticas propostas apresentaram desempenhos similares, mas a heurística de solução por parte (*Piecewise Strategy*) é em geral um método mais rápido. O tempo de execução foi fortemente influenciado pelo tamanho das mensagens em ambos os algoritmos (veja Tabela 3.3). À medida que o tamanho das mensagens aumentava, o tempo de execução também aumentava. Esse efeito pode ser explicado pelo fato de que resolver um algoritmo de roteamento em uma rede congestionada é uma tarefa difícil. Na prática, o gargalo computacional dos algoritmos propostos é o método

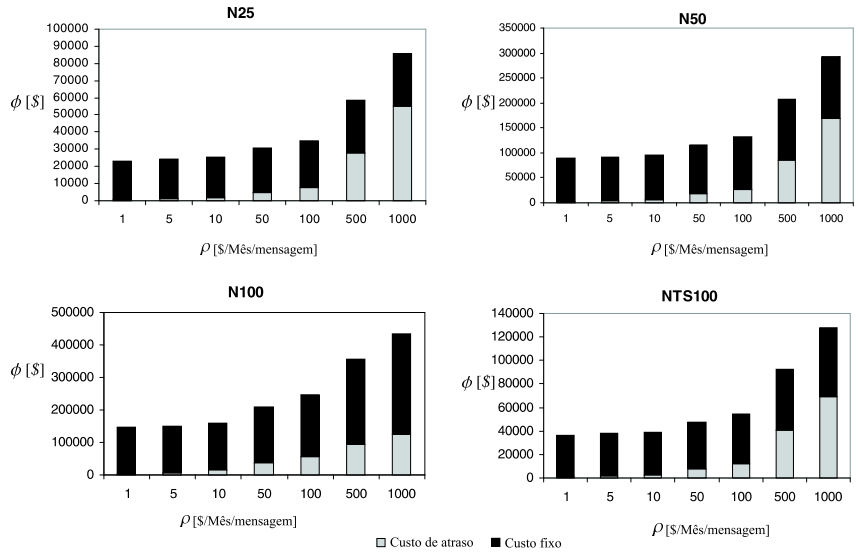


Figura 3.13: Custos de congestionamento e custos fixos para diferentes parâmetros de proporcionalidade ρ. O tamanho das mensagens para cada conjunto de testes foi fixado como sendo: $40[kbits]$ para N25, $10[kbits]$ para N50 e N100, e $1[kbit]$ para NTS100

de resolução do problema de roteamento de fluxos multiproduto que é gerado pela heurística. O método de desvios de fluxos foi escolhido como algoritmo de roteamento por causa da precisão requerida e porque, se considerações sobre os recursos de memória requisitados, este método continua competitivo para resolver o problema de fluxos multiproduto convexo.

3.6 Comentários

Um novo conjunto de modelos e algoritmos foi implementado para integrar o projeto e a operação de redes de computadores. A abordagem integrada adotada associa ao atraso do pacote uma função de custos de congestionamento, de tal maneira que o problema todo pode ser visto em termos de um único critério de custos. O resultado é que tanto o modelo contínuo adotado quanto as heurísticas de otimização lidam simultaneamente com dois critérios antagônicos do problema. Por um lado, as capacidades dos arcos são atribuídas ao mínimo custo, e por outro lado, os fluxos de comunicação devem ser roteados de forma a obter a maior qualidade de serviço.

Embora a medida de qualidade de serviço adotada seja a interpretação clássica do atraso sofrido pelas mensagens na rede, uma extensão natural poderia levar em conta muitos outros critérios de qualidade. A abordagem pode incorporar qualquer caso em

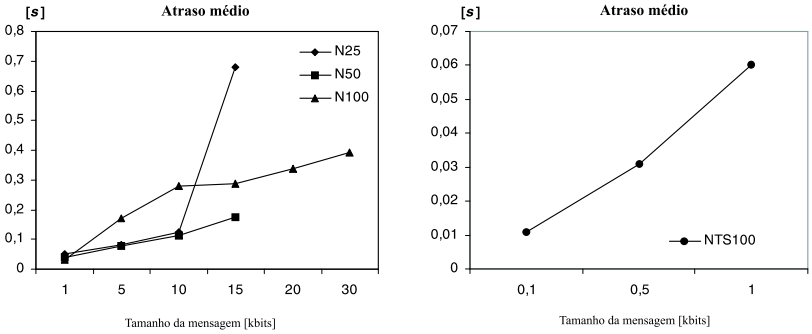


Figura 3.14: Atraso médio para diferentes tamanhos de mensagens, com a constante de proporcionalidade fixada em $\rho = 100\$/mês/mensagem$

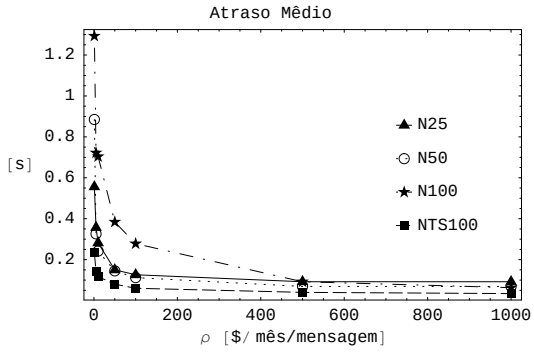


Figura 3.15: Atraso médio das mensagens para diferentes parâmetros de proporcionalidade ρ. O tamanho das mensagens para cada conjunto de testes foi fixado como sendo 40$[kbits]$ para N25, 10$[kbits]$ para N50 e N100, e 1$[kbits]$ para NTS100.

que uma função de custos de qualidade convexa é uma representação adequada de um critério de qualidade alternativo, por exemplo: probabilidade de estouro de pilha, probabilidade de bloqueio de chamada ou taxa de perdas de pacotes. Em geral, uma combinação linear de muitos critérios diferentes de qualidade pode ser convertida em custo no contexto da metodologia.

Uma boa aproximação convexa da função no arco é uma dos principais resultados explorados nesta abordagem integrada. Foi mostrado como obter limites inferiores aplicando algoritmos eficientes para a solução do problema de fluxos multiproduto convexo resultante. Economias de escala na atribuição de capacidades induzem concavidade nos custos fixos, mas os custos convexos do congestionamento fazem da função objetivo integrada o ínfimo de um conjunto de funções convexas. A convexificação da função objetivo foi a chave para demonstrar que as heurísticas propostas têm desempenho garantido. Os pequenos tempos de execução obtidos para resolver os problemas-testes indicam que a metodologia pode ser aplicada para resolver problema de grande porte. De fato, os experimentos computacionais sugerem que o procedimento é eficiente e efetivo na identificação de boas soluções de problemas práticos. Melhores algoritmos de roteamento poderiam ser usados eventualmente para melhorar o tempo de execução.

Uma estratégia bifurcada de roteamento foi adotada nos testes estudados neste trabalho. Motivado pelos requisitos de qualidade de serviço nas tecnologias emergentes, o próximo passo é adotar estratégias de roteamento não-bifurcadas. Os limites inferiores obtidos com o roteamento bifurcado continuam sendo limites inferiores do problema não-bifurcado. Essa importante propriedade será utilizada nos métodos de solução do difícil problema não-bifurcado.

Tabela 3.3: Melhores resultados computacionais obtidos para diferentes tamanhos de mensagens.

Rede ID	Parâmetro custo de congestionamento ρ	Tamanho da mensagem $[kbit]$	Limite inferior $\ddot{\phi}$	Limite superior ϕ	Custo do atraso [$]	Custo fixo [$]	Atraso médio $[s]$	Tempo de execução $[s]$
N5	100	1	922	922	22	900	0.022	0
N5	100	20	1 405	1 863	363	1 500	0.363	0
N5	100	40	1 914	2 634	904	1 730	0.904	0
N5	100	60	2 423	3 035	605	2 430	0.605	0
N5	100	80	2 932	3 484	514	2 970	0.514	0
N5	100	100	3 443	3 778	718	3 060	0.718	0
N5	100	120	3 975	4 219	1 159	3 060	1.159	0
N25	100	1	10 499	11 528	3 028	8 500	0.0504	0
N25	100	5	20 228	25 179	4 849	20 330	0.0808	0
N25	100	10	32 483	34 861	7 561	27 300	0.126	27
N25	100	15	70 985	71 312	40 802	30 510	0.680	206
N50	100	1	41 251	44 717	9 567	35 150	0.039	3
N50	100	5	76 592	95 872	18 882	76 990	0.077	3
N50	100	10	120 828	132 892	27 422	105 470	0.112	197
N50	100	15	165 144	165 770	42 890	122 880	0.175	1280
N100	100	1	150 381	150 387	6 081	144 300	0.030	11
N100	100	5	176 891	187 787	34 046	153 740	0.170	40
N100	100	10	210 078	246 355	55 685	190 670	0.278	24
N100	100	15	243 361	295 223	57 443	237 780	0.287	222
N100	100	20	276 457	336 021	67 521	268 500	0.337	929
N100	100	30	342 879	400 878	78 448	322 430	0.392	1024
NTS100	100	0,1	29 904	30 651	2 301	28 350	0.011	2
NTS100	100	0.5	37 582	41 444	6 334	35 110	0.031	156
NTS100	100	1	48 638	54 822	12 012	42 810	0.060	359

Tabela 3.4: Melhores resultados obtidos para diferentes custos de congestionamento de proporcionalidade ρ.

Rede ID	Parâmetro custo de congestionamento ρ	Tamanho da mensagem $[kbit]$	Limite inferior $\ddot{\phi}$	Limite superior ϕ	Custo do atraso [$]	Custo fixo [$]	Atraso médio $[s]$	Tempo de execução $[s]$
N5	1	40	1 177	1 441	41	1 400	4.100	0
N5	5	40	1 288	1 608	208	1 400	4.160	0
N5	10	40	1 371	1 893	73	1 820	0.730	0
N5	50	40	1 697	2 140	500	1 640	1.000	0
N5	100	40	1 914	2 634	904	1 730	0.904	0
N5	500	40	2 908	3 855	885	2 970	0.177	0
N5	1 000	40	3 870	4 740	1770	2 970	0.177	0
N25	1	10	18 863	23 194	334	22 860	0.556	24
N25	5	10	20 663	24 073	1 073	23 000	0.357	93
N25	10	10	22 014	25 193	1 693	23 500	0.282	22
N25	50	10	27 825	30 652	4 522	26 130	0.150	25
N25	100	10	32 483	34 861	7 561	27 300	0.126	27
N25	500	10	57 877	58 348	27 748	30 600	0.092	36
N25	1 000	10	85 764	86 009	55 229	30 780	0.092	54
N50	1	10	70 634	88 819	2 169	86 650	0.885	908
N50	5	10	77 423	91 890	4 000	87 890	0.326	1393
N50	10	10	82 489	96 143	5 953	90 190	0.242	1207
N50	50	10	104 043	116 465	17 805	98 660	0.145	593
N50	100	10	120 828	132 892	27 422	105 470	0.112	197
N50	500	10	206 951	208 233	84 543	123 690	0.069	808
N50	1 000	10	292 803	292 801	169 111	123 690	0.069	524
N100	1	10	146 866	146 887	2 587	144 300	1.293	115
N100	5	10	151 524	151 522	7 222	144 300	0.722	178
N100	10	10	158 424	158 491	14 091	144 400	0.704	361
N100	50	10	192 009	209 370	38 430	170 940	0.384	203
N100	100	10	210 078	246 355	55 685	190 670	0.278	40
N100	500	10	275 259	355 902	92 412	263 490	0.092	22
N100	1 000	10	338 032	435 493	126 553	308 940	0.063	710
NTS100	1	1	33 883	36 498	468	36 030	0.234	202
NTS100	5	1	35 552	37 835	1 425	36 410	0.142	148
NTS100	10	1	36 931	39 141	2 351	36 790	0.117	138
NTS100	50	1	43 457	47 387	8 087	39 300	0.080	618
NTS100	100	1	48 638	54 822	12 012	42 810	0.060	359
NTS100	500	1	78 058	92 391	40 331	52 060	0.040	3590
NTS100	1 000	1	111 769	128 139	68 729	59 410	0.034	4409

Capítulo 4

Algoritmo para o Problema de Atribuição de Capacidades e Roteamento de Fluxos Não-Bifurcados

4.1 Introdução

Problemas de fluxos em redes multiproduto, tais como problemas de atribuição de capacidades e roteamento de fluxos não-bifurcados, são bem conhecidos por sua relevância no planejamento de redes [4, 5, 24, 29, 30, 43, 54]. Especificamente, esse problema assume uma única rota a ser utilizada por cada tipo de tráfego (dados, imagem e voz) entre cada par de nós que se comunicam na rede. Tal problema é desafiador dado que cada variável de projeto depende das demais, além da natureza inerentemente combinatória dos problemas de alocação de capacidade e roteamento não bifurcado, resultando em um problema de otimização combinatória NP-difícil.

Sugere-se um algoritmo heurístico baseado em limites inferiores para a solução de uma formulação integrada [49]. A separabilidade da função objetivo é utilizada para obter uma função convexa aproximada de vão explícito. Usam-se alguns limites para a otimização global de um problema de fluxos em redes com designação implícita de capacidades nos arcos. As rede são expandidas tão logo os custos de qualidade de serviço induzam uma expansão. A aplicação refere-se ao uso simultâneo de roteamento e à alocação de banda de modo a assegurar um nível aceitável de desempenho a custo mínimo. Algoritmos de roteamento contínuos como o método de desvios de fluxos não é normalmente adequado para tal problema, e métodos combinatórios exatos

demandam excessivo esforço computacional. Entretanto, quando a rede tem demanda balanceada e um bom número de nós ($|n| \geq 25$), uma heurística gulosa inspirada no método de desvio de fluxos apresenta bons resultados [29].

A eficiência de uma heurística proposta aqui para resolver a formulação integrada será avaliada através de experimentos numéricos.

4.2 O Problema de atribuição de capacidades discretas e roteamento estático não-bifurcado

Apresenta-se nesta seção uma extensão não-bifurcada da formulação integrada para atribuição discreta de capacidades e fluxos bifurcados proposta no capítulo 3 em 3.2.1-3.2.5. Seja P_{kh}, $(h = 1, ..., N_k)$ um conjunto de N_k caminhos diretos que podem ser utilizados para o transporte do produto k conectando O_k e D_k em G. Esse conjunto pode ser um conjunto de todos os caminhos diretos entre O_k e D_k ou um sub-conjunto de caminhos viáveis. Seja x_{kh} o volume de fluxo do produto k através do caminho estático direto h. Seja y_{kh} uma variável de decisão, onde $y_{kh} = 1$ se o caminho h é selecionado para transportar o fluxo x_{kh}, e $y_{kh} = 0$ zero caso contrário. Seja a^{kh} o vetor de incidência arco-caminho m-dimensional.

O problema de atribuição de capacidades e roteamento não-bifurcado é definido como:

$$\text{minimize} \quad \sum_{i=1}^{m} \tau_i(f_i) \tag{4.2.1}$$

$$\text{sujeito a} \quad \sum_{k=1}^{K} \sum_{h=1}^{N_k} a_i^{kh} x_{kh} = f_i, \quad \forall i = 1, ..., m \tag{4.2.2}$$

$$\sum_{h=1}^{N_k} x_{kh} = d_k, \quad k = 1, ..., K \tag{4.2.3}$$

$$0 \leq f_i \leq c_i^{n_c}, \quad \forall i = 1, ..., m \tag{4.2.4}$$

$$x_{kh} \in \Re^{+}, \quad k = 1, ..., K, h = 1, ..., N_k \tag{4.2.5}$$

$$x_{kh} \leq d_k y_{kh}, \quad k = 1, ..., K, h = 1, ..., N_k \tag{4.2.6}$$

$$\sum_{h=1}^{N_k} y_{kh} = 1, \quad k = 1, ..., K \tag{4.2.7}$$

$$y_{kh} \in \{0, 1\}, \quad k = 1, ..., K, h = 1, ..., N_k \tag{4.2.8}$$

As restrições (4.2.6)-(4.2.8) garantem que para cada produto um único caminho seja selecionado.

A função objetivo definida em (4.2.1) gera um problema multiproduto não-convexo. Essa característica de não convexidade é inerente ao problema de decisão associado à escolha da capacidade de cada arco. Uma extensão do método cíclico de melhoria apresentado no capítulo anterior será adotada. A adaptação do método cíclico de melhoria é justificada pelo fato de que os limites inferiores obtidos para o problema bifurcado são limites também válidos para o problema não-bifurcado correspondente.

A heurística encontra uma solução viável e, então, gradualmente reduz o custo da solução obtida até que nenhuma solução melhor possa ser encontrada para (4.2.1) a (4.2.8). Ela é baseada em três fases.

Na primeira fase, o problema convexificado é solucionado sem as restrições (4.2.6), (4.2.7) e (4.2.8). A solução obtida $\ddot{\phi}$ é um limite inferior para o problema original.

$$\ddot{\phi} = \inf_{f \in F} \left\{ \sum_{i}^{m} conv\ \tau_i(f_i) \right\} \tag{4.2.9}$$

onde F é definido por (4.2.2) a (4.2.5).

Na segunda fase, o roteamento obtido é usado como ponto de partida para encontrar uma solução heurística $\breve{\phi}$ do problema não bifurcado dado por (4.2.1)-(4.2.8) novamente com a função objetivo convexificada.

$$\breve{\phi} = \inf_{\breve{f} \in F} \left\{ \sum_{i}^{m} conv\ \tau_i(\breve{f}_i) \right\} \tag{4.2.10}$$

onde F é definido por (4.2.2) a (4.2.8).

Na terceira fase, o roteamento obtido é usado como solução inicial para o método cíclico entre atribuição de capacidades e roteamento de fluxos não bifurcado até que não ocorram mais melhorias significativas no valor da função objetivo.

$$\hat{\phi} = \inf_{\hat{f} \in F} \left\{ \sum_{i}^{m} \tau_i(\hat{f}_i) \right\} \tag{4.2.11}$$

onde F é definido por (4.2.2) a (4.2.8).

Em termos práticos, o gargalo computacional do algoritmo proposto está no método de solução de problemas não-lineares multiproduto não-bifurcados. Se fosse

possível resolver o problema de roteamento não-bifurcado de maneira eficiente o algoritmo heurístico proposto teria desempenho garantido, tal como no caso bifurcado.

Método cíclico de melhoria

Fase I: Fase de aproximação convexa com roteamento bifurcado:

Passo 1- Encontre uma solução inicial viável f^0 para o problema bifurcado, $t = 0$.

Passo 2- Aplique um algoritmo de roteamento para encontrar o ótimo $\ddot{f}$ do problema convexo aproximado. O valor da função objetivo obtido $\ddot{\phi}$ é um limite inferior do problema (4.2.1)-(4.2.5).

Fase II: Fase de aproximação convexa com roteamento não-bifurcado:

Passo 3- Partindo do roteamento bifurcado obtido determine uma solução inicial não-bifurcada f^0_{nb}, definida como o conjunto de rotas mais curtas sob a métrica $\left(l_i = \frac{\partial conv\ \tau_i(f_i)}{\partial f_i}\right)$.

Passo 4- Aplique um algoritmo de roteamento não-bifurcado para encontrar uma solução $\breve{f}_{nb}$ do problema convexificado. Se $|\breve{\phi} - \ddot{\phi}| < \varepsilon$, **pare** com $\breve{f}_{nb}$, sendo, com erro ε, um ótimo global do problema não-bifurcado convexificado. Se isso ocorre é possível determinar um limite superior teórico para o problema (4.2.1)-(4.2.8).

Passo 5- Avalie a função objetivo não-convexa original $\tilde{\phi}$, para o vetor de fluxos $\breve{f}_{nb}$. Se $|\tilde{\phi} - \ddot{\phi}| < \varepsilon$, **pare**. Se $\breve{f}_{nb}$ for, com erro ε, um ótimo global do problema não-bifurcado convexificado, então $\breve{f}_{nb}$ é também com erro ε, um ótimo global de (4.2.1)-(4.2.8). Caso contrário, vá para a **Fase III**.

Fase III: Método cíclico de melhoria:

Passo 6- $\hat{f}_{i_{nb}} = \breve{f}_{i_{nb}} \ \ \forall i = 1, ..., m.$

Passo 7- Usando a rota obtida, para cada arco i, assinale uma capacidade aplicando as seguinte regras:
se $0 \leq \hat{f}_{i_{nb}} \leq \gamma_i^0 c_i^0$ então $c_i = c_i^0$;

se $\gamma_i^j c_i^j \leq \hat{f}_{i_{nb}} < \gamma_i^{j+1} c_i^{j+1}$ então $c_i = c_i^{j+1}$.

Passo 8- Aplique um algoritmo de roteamento não-bifurcado para o problema convexo multiproduto resultante e obtenha uma nova solução para (4.2.1)-(4.2.8).

Passo 9- Se $|\hat{\phi}(\hat{f}_{nb}^t) - \hat{\phi}(\hat{f}_{nb}^{t+1})| < \epsilon$, **pare**; senão $t \leftarrow t+1$ e vá para o **Passo 6**.

Esse algorimo produz uma seqüência decrescente e limitada inferiormente de soluções viáveis. Uma série de testes foram executados com o objetivo de mostrar que esse algoritmo é eficaz na obtenção de boas soluções.

4.2.1 Roteamento não-bifurcado

A introdução das restrições de não-bifurcação transforma o conjunto de roteamentos de fluxos multiprodutos viável em um conjunto discreto. O número de elementos desse conjunto é igual ao número de todas as possíveis combinações de caminhos entre todos os pares origem-destino e este problema é NP-difícil. O problema de roteamento não-bifurcado de fluxos com função de custos linear e atribuição de capacidades é denominado na literatura de problema de projeto de redes capacitadas [11]. O estado da arte dos algoritmos de resolução exata desse problema tem resolvido instâncias com, por exemplo: 14 nós, 44 arcos e 210 produtos [10] ou 29 nós, 61 arcos e 70 produtos [6]. Cabe ressaltar que nas formulações adotadas não existe um limite superior para a capacidade a ser instalada que é tratada como tendo um valor múltiplo de uma capacidade pré-definida. Não são feitas considerações sobre economia de escala nos custos de instalação de capacidades.

Não existe muita pesquisa publicada sobre o problema de roteamento não-bifurcado e não-linear. As abordagens de solução encontradas se concentram em métodos de Relaxação Lagrangeana [14] [25][67].

Métodos contínuos como o método de desvio de fluxos não podem ser usados na solução de problemas de roteamento de fluxos não-bifurcados. Entretanto, Gerla [29] mostrou que quando a rede é grande e balanceada uma heurística baseada no método de desvios de fluxos pode ser aplicada apresentando bons resultados.

Uma rede é denominada grande e balanceada se:

$$\eta \triangleq \frac{\omega\sigma}{(K-1)\bar{K}} \ll 1 \tag{4.2.12}$$

onde:

$$\sigma \triangleq \max_{ij} \left[\frac{d_{ij}}{d}\right]$$

a razão entre a maior demanda e a demanda média, e

$$d \triangleq \frac{1}{(K-1)K} \sum_{ij} d_{ij}$$

é a demanda média por par de nós (i, j), e

$$\omega \triangleq \frac{m}{n}$$

a densidade média de arcos por nós na rede, e

$$\bar{K} \triangleq \frac{\sum_{ij} d_{ij}\bar{l}_{ij}}{\sum_{ij} f_{ij}}$$

onde $\bar{l}_{ij}$ é o comprimento em número de arcos do caminho mais curto entre (i, j), e f_{ij} é o fluxo total no caminho (i, j).

O fato é que, como foi demonstrado por Kleinrock em [41], em uma rede grande e balanceada, na média, o fluxo de um único produto em qualquer arco pode ser considerado como sendo infinitesimal, quando comparado ao fluxo total naquele arco. Uma conseqüência desse resultado é que, se a rede for grande e balanceada, a solução do problema de roteamento com uma formulação bifurcada é uma boa aproximação para o problema não-bifurcado. Tal fato também permitiu a Gerla desenvolver uma heurística baseada no método de desvios de fluxos e em uma estratégia de coordenadas descendentes para resolver o problema de roteamento não-bifurcado. Esta heurística foi adotada e os resultados obtidos e apresentados nas próximas seções atestam que os resultados não-bifurcados se aproximam dos bifurcados.

4.3 Experimentos Computacionais

Dois conjuntos de experimentos foram conduzidos com o objetivo de estudar o algoritmo apresentado.

Tabela 4.1: Capacidades e seus custos correspondentes

Capacidade c $[kbps]$	Custo de instalação π $[\$/mês]$
64	150
128	250
256	390
384	480
512	570

O método de solução foi implementado em linguagem C via GCC 3.0, sendo a plataforma de implementação um computador com processador AMD DURON 950 MHz, 128 Mb de RAM e sistema operacional LINUX.

Cinco topologias de redes foram usadas nos testes. As três primeiras topologias estão representadas nas Figuras 3.5, 3.6 e 3.7 e foram estudadas para o problema com roteamento bifurcado apresentado no capítulo 3. As outras duas topologias (Figuras 4.7 e 4.8) são tradicionais na literatura e usam parâmetros de tráfego e estrutura de custos similares às usadas em [29], [26], [24], [4], e [5].

O mesmo modelo adotado no Capítulo 3 para a função $T_i(f_i, c_i)$ de custos de qualidade de serviço na rede foi empregado. O custo fixo para a instalação de uma capacidade c em um arco i de comprimento d_i é calculado por $\pi^c = S^c + D^c d_i$, e S^c incorpora o custo de equipamentos terminais no nível de capacidade c e D^c inclui o custo unitário (por quilômetro) de toda infra-estrutura de cabeamento para se instalar o nível de capacidade c no arco i.

4.3.1 Primeiro conjunto de experimentos

Para as topologias N50, N100 e NTS100, dois mil pares origem-destino foram criados aleatoriamente. Uma sessão é suposta ativa para cada par origem-destino, gerando tráfego de uma mensagem por segundo. As capacidades e suas correspondentes componentes de custo são apresentadas na Tabela 4.1.

A Tabela 4.2 e as Figuras 4.1, 4.2 e 4.3 exibem os resultados para diferentes tamanhos de mensagens, para um custo de congestionamento de $\rho = 100[\$/mês/mensagem]$. O tempo de execução é fortemente influenciado por esse parâmetro. Com o aumento do tamanho de mensagem, correspondente a uma maior carga na rede, cai a qualidade das soluções obtidas. Todavia, quando tal carga se aproxima da máxima saída da rede,

Tabela 4.2: Resultados computacionais para diferentes tamanhos de mensagens

Rede ID		Tamanho da mensagem	Razão	Limite inferior	Limite superior	Custo de qualidade	Custo fixo	Atraso médio	Tempo de execução
	ρ	$[kbits]$	$\frac{\hat{\phi}}{\ddot{\phi}}$	$\ddot{\phi}$	$\hat{\phi}$	[$]	[$]	$[s]$	$[s]$
N50	100	1	1.08	41251	44739	9689	35050	0.03	2.9
N50	100	5	1.24	76592	95662	19172	76490	0.078	3.4
N50	100	10	1.08	120828	131398	29178	102220	0.11	6.9
N50	100	15	1.00	165144	166130	44250	121880	0.18	4.4
N100	100	1	1.00	150387	150374	6074	144300	0.03	7.4
N100	100	5	1.07	176891	190117	32017	158100	0.16	5.0
N100	100	10	1.21	210078	254974	54676	200290	0.27	8.2
N100	100	15	1.21	243361	295332	68102	227230	0.34	13.2
N100	100	20	1.20	276457	333096	67538	265510	0.33	11.4
NTS100	100	0.5	1.10	37582	41455	6485	34970	0.032	18.1
NTS100	100	1.0	1.13	48638	54933	12323	42610	0.062	14.2

a qualidade das soluções volta a melhorar ($\frac{\bar{\phi}}{\phi} \simeq 1$). A chave para esse fenômeno está na Figura 3.2 pois, para fluxos menores que f_0 ou maiores que f_1, a função convexa aproximada coincide com a função original. O custo fixo é dominante no custo total na maioria dos testes realizados. A exceção ocorre para as redes N50 e N100 quando o tamanho das mensagens é de $10[kbits]$ e $\rho = 1000[\$/mês/mensagem]$.

A Tabela 4.3 e as Figuras 4.4, 4.5 e 4.6 apresentam os resultados obtidos para diferentes custos de congestionamento, para um tamanho de mensagem de $10[kbits]$ para N50 e N100 e $1[kbit]$ para NTS100. A qualidade dos resultados sofre influência desse custo, uma vez que o vão máximo Δ é uma função do custo de congestionamento. Outro efeito observado evidencia que o aumento de ρ diminui o atraso médio das mensagens na rede. Por outro lado, os custos de congestionamento aumentam substancialmente.

As Figuras de 4.1 a 4.6 comparam os resultados obtidos com roteamento bifurcado e não-bifurcado. Elas evidenciam que o algoritmo de roteamento não-bifurcado é eficiente e, os resultados obtidos são muito próximos dos resultados obtidos com roteamento bifurcado.

4.3.2 Segundo conjunto de experimentos

O problema teste 1, Figura 4.7, é uma rede de 21 nós, 26 arcos (*full duplex*). Nessa rede cada nó pode se comunicar com todos os demais, e é assumido um tráfego de oito mensagens por segundo em cada direção, para cada par de nós. Totalizam-se,

Tabela 4.3: Resultados computacionais para diferentes custos de congestionamento

Rede ID		Tamanho da mensagem	Razão	Limite inferior	Limite superior	Custo de qualidade	Custo fixo	Atraso médio	Tempo de execução
	ρ	$[kbits]$	$\frac{\hat{\phi}}{\ddot{\phi}}$	$\ddot{\phi}$	$\hat{\phi}$	[$]	[$]	$[s]$	$[s]$
N50	1	10	1.13	76408	86586	886	85700	0.36	9.4
N50	5	10	1.18	77423	91840	3710	88130	0.30	8.1
N50	10	10	1.16	82489	95654	6594	89060	0.27	1.5
N50	50	10	1.09	104043	114445	17766	96678	0.14	1.3
N50	100	10	1.08	120828	131398	29178	102220	0.11	6.9
N50	500	10	1.00	206951	208565	87095	121470	0.07	6.9
N50	1000	10	1.00	292803	292858	169168	123690	0.068	4.0
N100	1	10	1.08	150186	162673	973	161700	0.48	6.0
N100	5	10	1.09	151524	165780	4978	160800	0.48	5.0
N100	10	10	1.13	158424	179377	8177	171200	0.41	4.2
N100	50	10	1.13	192009	217592	33292	184300	0.33	6.0
N100	100	10	1.21	210078	254974	54676	200290	0.27	8.2
N100	500	10	1.18	275259	327265	75535	251730	0.07	6.6
N100	1000	10	1.14	338032	386571	101201	285370	0.05	5.7
NTS100	1	1	1.04	35232	36874	344	36530	0.17	20.7
NTS100	5	1	1.07	35552	38160	1250	36910	0.12	21.2
NTS100	10	1	1.06	36931	39357	2167	37190	0.108	21.4
NTS100	50	1	1.08	43457	47313	7883	39430	0.079	18.7
NTS100	100	1	1.12	48638	54933	12323	42610	0.062	14.2
NTS100	500	1	1.18	78058	92077	40507	51570	0.041	10.7
NTS100	1000	1	1.15	111769	128716	69436	59280	0.034	18.0

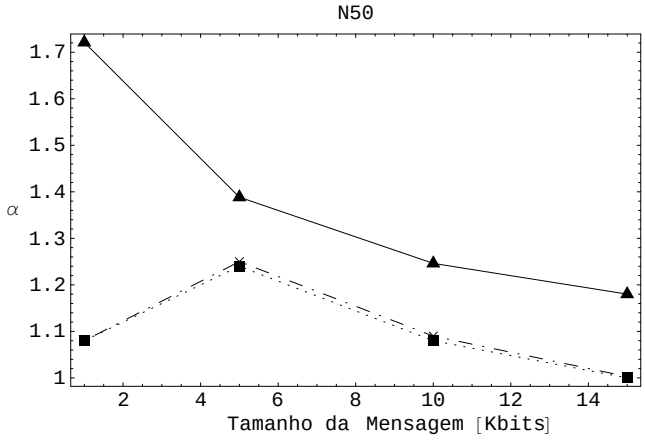


Figura 4.1: Comparação entre os resultados da rede N50: α^* (■) resultado não-bifurcado, α_{pc} (×) bifurcado e α (▲) limite superior do bifurcado para diferentes tamanhos de mensagens

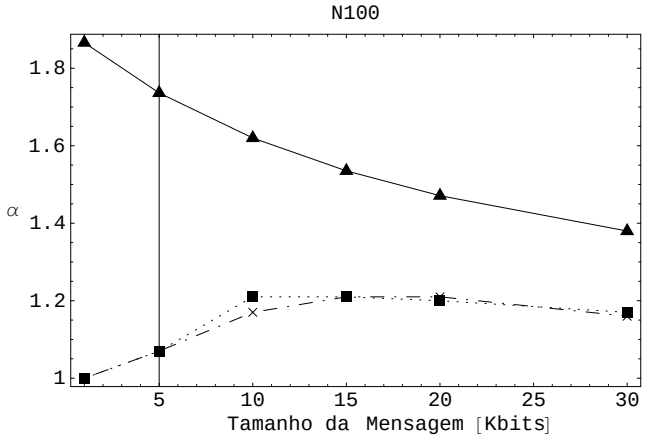


Figura 4.2: Comparação entre os resultados da rede N100: α^* (■) resultado não-bifurcado, α_{pc} (×) bifurcado e α (▲) limite superior do bifurcado para diferentes tamanhos de mensagens

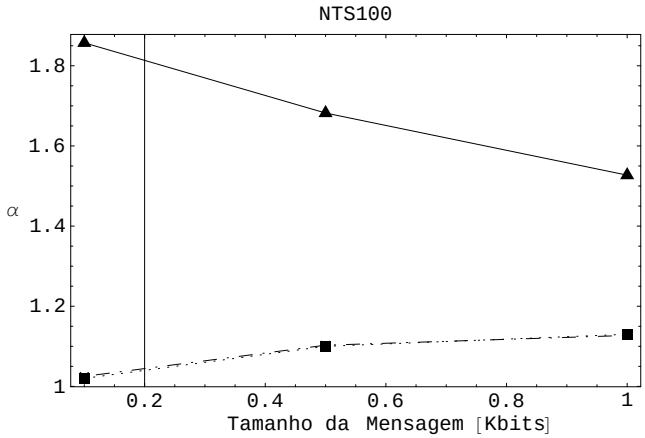


Figura 4.3: Comparação entre os resultados da rede NTS100: α^* (■) resultado não-bifurcado, α_{pc} (×) bifurcado e α (▲) limite superior do bifurcado para diferentes tamanhos de mensagens

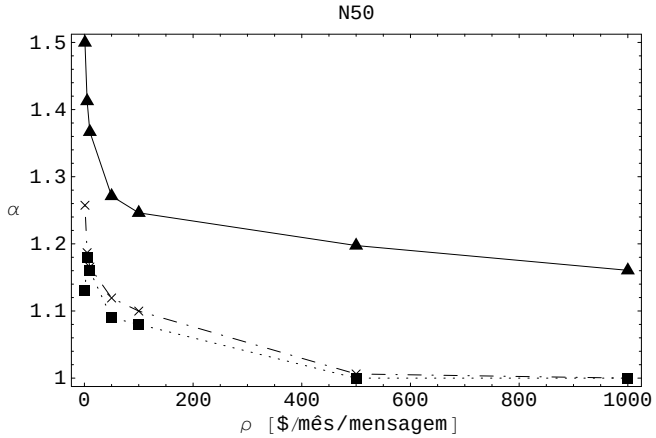


Figura 4.4: Comparação entre os resultados da rede N50 α^* (■) resultado não-bifurcado, α_{pc} (×) bifurcado e α (▲) para diferentes custos de congestionamento

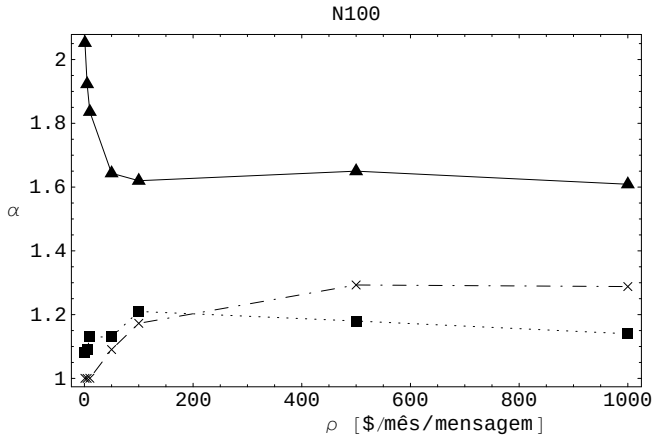


Figura 4.5: Comparação entre os resultados da rede N100 α^* (■) resultado não-bifurcado, α_{pc} (×) bifurcado e α (▲) para diferentes custos de congestionamento

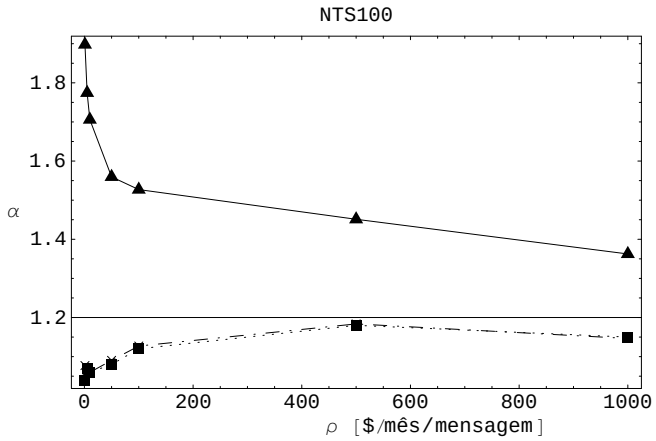


Figura 4.6: Comparação entre os resultados da rede NTS100 α^* (■) resultado não-bifurcado, α_{pc} (×) bifurcado e α (▲) para diferentes custos de congestionamento

então, 210 pares origem-destino.

O problema teste 2, Figura 4.8, é uma rede de 32 nós, e 50 arcos (*full duplex*). Nessa rede cada nó pode se comunicar com todos os demais e é assumido um tráfego de oito mensagens por segundo em cada direção, para cada par de nós. Totaliza-se então, 496 pares origem-destino.

Para essas duas topologias, as capacidades disponíveis para instalação e seus respectivos custos são apresentados na Tabela 4.4. Estes custos foram adotados em diferentes artigos que tratam do problema CFA [26], [24], [4], e [5]. Entretanto,

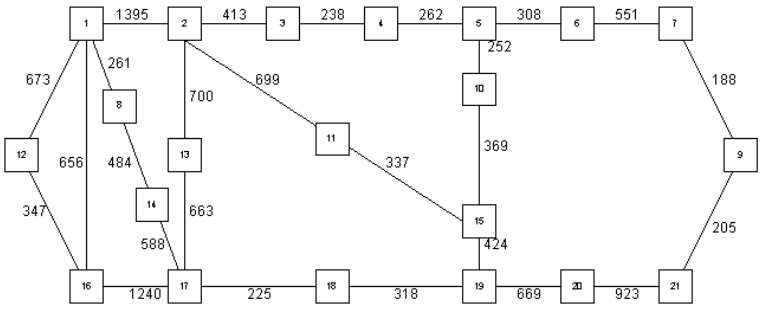


Figura 4.7: Topologia de rede ARPA com 21 nós e 26 arcos

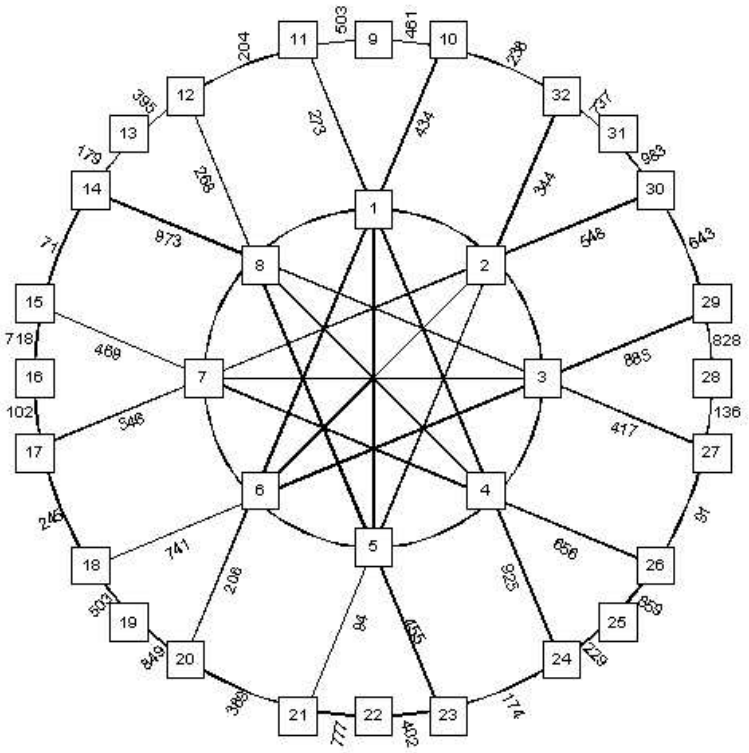


Figura 4.8: Topologia de rede RING com 32 nós e 60 arcos

eles não apresentam economia de escala em todas as situações, sendo esse um comportamento incomum. Uma conseqüência dessa anomalia é que a combinação de canais com capacidades menores pode ser economicamente mais interessante do que instalar um canal de grande capacidade no arco. Por exemplo, o custo fixo de instalação da capacidade de 460000 $[bps]$ em um arco com $d_{ij} = 50$ $[milhas]$ é de $1300 + 60 * 500 = 31300$ $[\$/mês]$ o custo variável máximo neste canal é de $460000 * 0.017 = 7820$ $[\$/mês]$. Assim teria-se um custo fixo mais um custo variável de 39120 $[\$/mês]$. Por outro lado, se tivessem sido instalados dois canais de 230000 $[bps]$, o custo fixo, de instalação de capacidade seria de $2 * 1300 + 21 * 500 = 13100$, e o custo variável máximo neste canal seria de $2 * 230000 * 0.020 = 9200$ $[\$/mês]$. Assim o custo variável mais o custo fixo atingiriam 22300 $[\$/mês]$. Nesse caso, fica evidente que instalar dois canais de 230000 $[bps]$ é mais barato do que um de 460000 $[bps]$.

Tendo observado que a combinação de capacidades pode fornecer resultados mais consistentes do que a instalação somente das capacidades apresentadas na Tabela 4.4, no primeiro conjunto de testes, duas classes de problemas foram estudadas, a saber: os problemas denominados de N7, tratados considerando que a capacidade de cada arco necessariamente é uma das capacidades apresentadas na Tabela 4.4, e os problemas denominados de NC7, tratados considerando que a capacidade de cada

Tabela 4.4: Capacidades disponíveis e seus respectivos custos

Capacidade c $[bps]$	Custo de instalação S_c $[\$/mês]$	Custo de distância D_c $[\$/mês/milhas]$	Custo variável v_c $[\$/mês/bps]$
4800	650	0.4	0.360
9600	750	0.5	0.252
19200	850	2.1	0.126
50000	850	4.2	0.030
108000	2400	4.2	0.024
230000	1300	21	0.020
460000	1300	60	0.017

$\pi^c = S^c + D^c d_{ij}$

arco é uma combinação com repetição de uma, duas ou três capacidades escolhidas da Tabela 4.4.

Para a classe NC7 há 119 alternativas de capacidade disponíveis
$(C(8+3-1,3)-1 = C(10,3)-1 = \frac{10!}{3!7!} - 1 = 119)$.

Os componentes de custo das capacidades geradas $\bar{c}_i$ através das equações:

$$c_i^0 = 0 \tag{4.3.1}$$

$$\sum_{k=1}^{7} \delta_k = 3, \quad \delta_k \in [0,1,2,3] \tag{4.3.2}$$

$$\bar{c}_i = \sum_{k=1}^{7} \delta_k c_i^j, \quad j \in [0,1,...,7], \ \forall i = 1,...,m; \tag{4.3.3}$$

$$f_i = \sum_{k=1}^{7} \delta_k f^j, \quad j \in [0,1,...,7], \ \forall i = 1,...,m; \tag{4.3.4}$$

$$\pi_{\bar{c}_i} = \sum_{k=1}^{7} \delta_k \pi^{c_i^j}, \quad j \in [0,1,...,7], \ \forall i = 1,...,m; \tag{4.3.5}$$

$$v_{\bar{c}_i} = \frac{\sum_{k=1}^{7} \delta_k v_{c_i^j} c_i^j}{\bar{c}_i} \quad j \in [0,1,...,7], \ \forall i = 1,...,m; \tag{4.3.6}$$

$$T_i(f_i, \bar{c}_i) = \rho \frac{f_i}{\bar{c}_i - f_i}, \quad \forall i = 1,...,m. \tag{4.3.7}$$

Na Figura (4.9) pode-se ver a função de custo com $d_{ij} = 300$ e $\rho = 1000$ considerando os casos N7 e NC7 com 7 e 119 capacidades disponíveis para instalação.

Os custos de congestionamento adotados foram $\rho \in \{1000, 2000, 3000\}[\$/mês/mensagem]$.

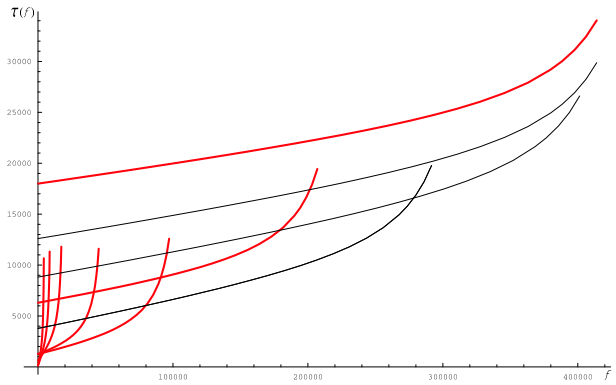


Figura 4.9: Função de custo no arco com $d_{ij} = 500$ e $\rho = 1000$ considerando os casos com 7 e 119 capacidades. As curvas em negrito representam as 7 capacidades originais.

As Tabelas 4.5 e 4.6 apresentam os resultados obtidos variando o tamanho de mensagem. Os resultados para diferentes custos de atraso são apresentados nas Tabelas 4.7 e 4.8.

Comparações entre soluções deste trabalho e as de Gavish e Neuman [26], Gavish e Altinkemer [24] e Amiri e Pirkul [4] são mostradas nas Tabelas 4.9 e 4.11 e nas Figuras 4.10 e 4.11. As Tabelas 4.9 e 4.11 e as Figuras 4.10 e 4.11 comparam os resultados obtidos e os resultados obtidos por Gavish e Altinkemer [24], Gavish e Neuman [26] e Amiri e Pirkul [4]. Os resultados para os problemas N7 são equivalentes aos apresentados na literatura, mas para NC7 observa-se a obtenção de resultados melhores. Isto pode ser explicado pelo fato de que o uso de capacidades combinadas permite encontrar um capacidade mais próxima das necessidades de um arco. O aumento do número de capacidades disponíveis não afeta o tempo de solução. Os resultados dos experimentos indicam que o algoritmo baseado em aproximação convexa para o modelo integrado de custos de instalação de capacidade e congestionamento aqui proposto é eficiente na solução de problemas de alocação de fluxos e capacidades. Quando comparados aos resultados de [26], [24] e [4] nota-se, que foi possível obter resultados similares com menores números de iterações e tempos de CPU. Uma comparação preliminar é apresentada na Tabela 4.11 que é meramente indicativa. É importante ressaltar que o método de solução descrito em [24] foi escrito em FORTRAN, o método de solução proposto em [4] em PASCAL e o método apresentado neste trabalho em C.

Tabela 4.5: Resultados computacionais obtidos para diferentes tamanhos de mensagens, 7 capacidades

Rede ID	Tamanho da mensagem [$bits$]	Limite inferior $\ddot{\phi}$	Limite superior $\hat{\phi}$	Custo do atraso [$]	Custo fixo [$]	Custo variável [$]	Atraso médio [s]	Tempo de execução [s]
ARPA	200	152152	190335	45103	28531	116701	0.0134	0.00
ARPA	300	220324	248472	41274	39470	167727	0.0134	0.01
ARPA	400	291584	309169	81466	52697	175005	0.0242	0.01
ARPA	500	370368	385745	118219	64476	205050	0.0351	0.02
RING	200	286242	363263	78187	57713	227362	0.0098	1.00
RING	300	421173	473485	96688	79447	297350	0.0121	1.02
RING	400	527285	578646	116749	101481	360416	0.0147	1.00
RING	500	674214	705983	197162	125620	383201	0.0248	2.03

Tabela 4.6: Resultados computacionais obtidos para diferentes tamanhos de mensagens, 119 capacidades

Rede ID	Tamanho da mensagem [$bits$]	Limite inferior $\ddot{\phi}$	Limite superior $\hat{\phi}$	Custo do atraso [$]	Custo fixo [$]	Custo variável [$]	Atraso médio [s]	Tempo de execução [s]
ARPA	300	179241	245625	47921	41593	156111	0.0149	0.00
ARPA	400	240559	302154	63673	55285	183233	0.0190	0.01
ARPA	500	301223	351710	69368	68523	213882	0.0206	0.00
RING	200	208303	362003	80488	59189	225722	0.0101	7.04
RING	300	313150	464924	94623	83137	290718	0.0119	4.21
RING	400	417301	562028	114263	107846	342147	0.0144	3.50
RING	500	522847	657086	143146	134132	382273	0.0180	3.17

Tabela 4.7: Resultados computacionais obtidos para diferentes custos de congestionamento, 7 capacidades

Rede ID	Custo de congetionamento ρ	Limite inferior $\ddot{\phi}$	Limite superior $\hat{\phi}$	Custo do atraso [$]	Custo fixo [$]	Custo variável [$]	Atraso médio [s]	Tempo de execução [s]
ARPA	1000	215602	260835	39784	53228	167822	0.0236	0.00
ARPA	2000	291584	309169	81466	52697	175005	0.0242	0.01
ARPA	3000	318404	346694	98580	50989	200203	0.0196	0.00
RING	1000	476978	525569	78320	104288	342961	0.0197	1.00
RING	2000	524603	575903	127484	102657	345761	0.0160	1.00
RING	3000	579456	642082	129174	100073	412834	0.0125	1.21

Tabela 4.8: Resultados computacionais obtidos para diferentes custos de congestionamento, 119 capacidades

Rede ID	Custo de congetionamento	Limite inferior	Limite superior	Custo do atraso	Custo fixo	Custo variável	Atraso médio	Tempo de execução
	ρ	$\ddot{\phi}$	$\hat{\phi}$	[$]	[$]	[$]	$[s]$	$[s]$
ARPA	1000	212625	268970	40237	55500	173233	0.0240	0.00
ARPA	2000	240559	302154	63673	55285	183233	0.0190	0.01
ARPA	3000	262479	332101	67962	54988	209232	0.0135	0.00
RING	1000	366068	494107	82273	109992	305409	0.0207	3.50
RING	2000	417301	562028	114263	107846	342147	0.0144	3.50
RING	3000	458974	620254	153039	108053	359162	0.0130	2.93

Tabela 4.9: Comparação de resultados obtidos por diferentes autores para diferentes tamanhos de mensagens e custo de congestionamento $\rho = 2000\$/mês/mensagem$

Rede ID	Tam. das mens.	N7 l.i.	N7 l.s.	G&A l.i	G&A l.s.	G&N l.i.	G&N l.s.	A&P l.i.	A&P l.s.	NC7 l.i.	NC7 l.s.
	$[bits]$	[$]	[$]	[$]	[$]	[$]	[$]	[$]	[$]	[$]	[$]
ARPA	200	152152	190335	159307	**185009**	165543	186457	*176513*	185565	150231	185000
ARPA	300	220324	247625	219270	**243927**	224499	245798	*235370*	245740	179241	247625
ARPA	400	291584	309169	285440	309137	288567	311079	*298361*	**308637**	240559	302154
ARPA	500	*370368*	385745	351546	379516	355536	377538	362662	**377433**	301223	351710
RING	300	421173	473485	400723	**463343**	*453291*	487288	436100	464817	313150	464924
RING	400	527285	578646	516780	**569141**	*571368*	595285	550219	571185	417301	562028
RING	500	674214	705983	634065	699065	*686015*	714269	663173	**697968**	522847	657086

G&A - Gavish and Altinkemer [24], G&N - Gavish and Neuman [26], A&P - Amiri and Pirkul [4]
l.i. limite inferior - maior limite em itálico
l.s. limite superior - menor limite em negrito

Tabela 4.10: Comparação de resultados obtidos por diferentes autores para diferentes custos de congestionamento e tamanho de mensagem fixo em 400[bits]

Rede ID		N7 l.i.	N7 l.s.	G&A l.i	G&A l.s.	G&N l.i.	G&N l.s.	A&P l.i.	A&P l.s.	NC7 l.i.	NC7 l.s.
	ρ	[$]	[$]	[$]	[$]	[$]	[$]	[$]	[$]	[$]	[$]
ARPA	1000	215602	**260835**	244592	265155	246569	265822	*254867*	262519	212625	268970
ARPA	2000	291584	309169	285440	309137	287428	311079	*298361*	**308637**	240559	302154
ARPA	3000	318404	346694	315010	344589	313269	343919	*331002*	**343719**	262479	332101
RING	1000	*476978*	525569	446966	495660	494690	518119	473529	**493792**	366068	494107
RING	2000	524603	578646	516780	**569141**	*571368*	595285	550219	571185	417301	562028
RING	3000	579456	642082	570650	635437	*829823*	664235	611231	**633607**	458974	620254

G&A - Gavish and Altinkemer [24], G&N - Gavish and Neuman [26], A&P - Amiri and Pirkul [4]
l.i. limite inferior - maior limite em itálico
l.s. limite superior - menor limite em negrito

Tabela 4.11: Comparação de tempo de execução e número de iterações para um custo de congestionamento $\rho = 2000$ e tamanho de mensagem 400[bits]

Rede ID	Nosso método Número de iterações	Nosso método Total CPU $[s]$	A&P Número de iterações	A&P Total CPU $[s]$	G&A Número de iterações ρ	G&A Total CPU $[s]$
ARPA	6	0	300	590	300	45
RING	7	1	300	3120	300	230

Esta tabela é meramente indicativa e baseada nas referências
G&A - Gavish e Altinkemer [24]
G&N - Gavish e Neuman [26]
A&P - Amiri e Pirkul [4]

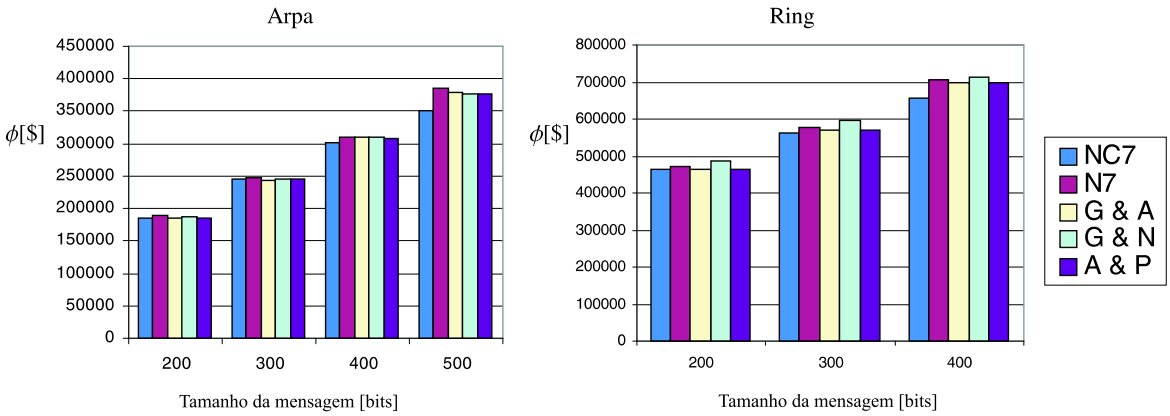


Figura 4.10: Comparação de resultados obtidos por diferentes autores para diferentes tamanhos de mensagens e custo de congestionamento $\rho = 2000$, onde: N7 e NC7 são obtidos nesse trabalho e G&A - Gavish e Altinkemer [24], G&N - Gavish e Neuman [26], A&P - Amiri e Pirkul [4]

4.4 Comentários

Um dos algoritmos propostos no capítulo 3 para resolver o problema de atribuição de capacidades e roteamento foi adaptado para resolver o mesmo problema, mas com restrições para permitir apenas roteamento não-bifurcado. A aproximação integrada usada aqui associa ao congestionamento uma função de custo, possibilitando que todo o problema seja visto em termos de um único critério de custo. Resulta que ambos, o modelo contínuo adotado e a heurística proposta, lidam simultanemamente com os dois critérios (conflitantes) do problema. Capacidades são atribuídas aos arcos, e fluxos de dados são roteados sem bifurcação visando a uma melhor qualidade de

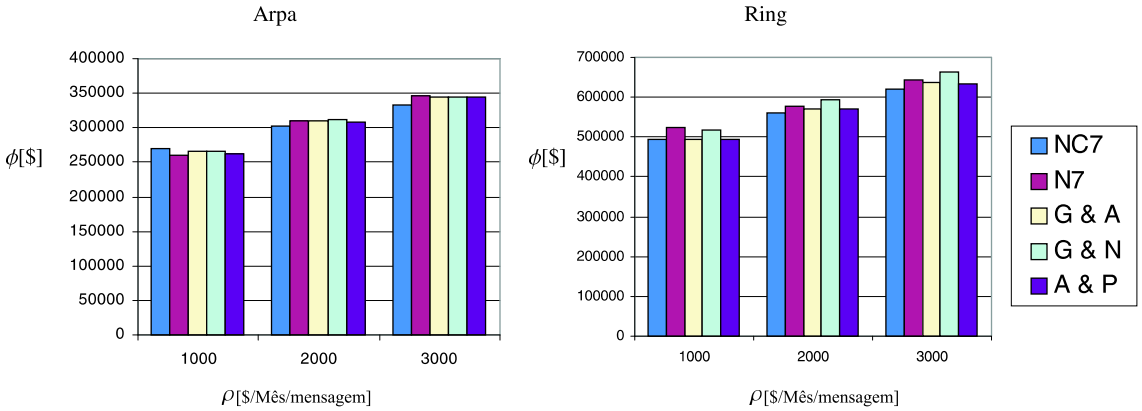


Figura 4.11: Comparação de resultados obtidos por diferentes autores para diferentes custos de congestionamento e tamanho de mensagem fixo em 400[bits], onde: N7 e NC7 são obtidos nesse trabalho e G&A - Gavish e Altinkemer [24], G&N - Gavish e Neuman [26], A&P - Amiri e Pirkul [4]

serviço.

Economias de escala na alocação de capacidades induzem concavidade em custos de assinatura, mas custos de congestionamento convexos fazem da função integrada um ínfimo de uma série de funções convexas. Pequenos tempos de execução indicam que o método proposto pode ser usado para resolver problemas de grande escala. De fato, os experimentos computacionais deste trabalho sugerem que o procedimento proposto é eficaz na identificação de boas soluções.

Os limites inferiores foram obtidos utilizando-se roteamento bifurcado. Tais soluções foram usadas como ponto de partida para a solução de problemas de roteamento não-bifurcado multiproduto, através de um método proposto por [29]. Um método simples porém eficaz em determinar boas soluções de ($\tilde{\phi}/\phi \simeq 1.05$) quando a rede é grande e balanceada. Outros algoritmos de roteamento poderiam ser utilizados para melhorar a qualidade das soluções. Do ponto de vista teórico, se fosse aplicado um algoritmo de roteamento exato, o algoritmo desenvolvido poderia ser considerado como de aproximação.

Na abordagem proposta, o atraso máximo em cada arco é controlado, o que é um diferencial em relação a outras formulações em que somente o atraso médio está sob controle, possibilitando a ocorrência de saturação em alguns arcos, mesmo com o atraso médio da rede sendo baixo.

A abordagem de solução proposta não é sensível ao número de capacidades disponíveis para instalação em um arco. Isso pode representar um vantagem na solução de

problemas.

Embora o modelo de custos de congestionamento utilizado nos experimentos tenha sido um modelo simples, ele estabelece que o custo aumenta à medida que o fluxo em um arco se aproxima de sua largura de banda. Esta é uma propriedade de outros modelos não-lineares tais como a probabilidade de estouro de *buffer*, a probabilidade de falha e o tempo de resposta médio. Atualmente a maioria das abordagens de medição do desempenho de uma rede tem se concentrado em modelos de simulação. Não tem sido empreendido muito esforço para se encontrar modelos analíticos melhores para medir o desempenho de uma rede. Mesmo assim, os modelos clássicos [43] continuam sendo uma ferramenta valiosa quando se desejam obter limites e ganhar "insight"em problemas de tamanho real.

Capítulo 5

Otimização global do problema de expansão de capacidades e roteamento de fluxos em redes de comunicação

5.1 Introdução

Este capítulo aborda o problema de expansão de redes. Tal problema considera que já existe uma rede instalada e que, por causa do aumento de tráfego ou de uma expectativa de aumento, deve-se reavaliar as capacidades dos canais de comunicação. Trata-se de um caso particular do problema de atribuição de capacidades e roteamento. A formulação proposta no capítulo 3 para o problema de atribuição de capacidades e roteamento bifurcado apresenta uma função objetivo contínua não-convexa. Essa formulação origina um problema de otimização global. Aproveitando propriedades da casca convexa da função objetivo, são propostos dois algoritmos com desempenho garantido. Entretanto, nenhum deles garante a otimalidade das soluções obtidas. Sem a certeza de que o ótimo global foi encontrado permanece, a questão se é possível melhorar um resultado obtido.

Uma das dificuldades em se formular métodos de otimização global está na caracterização das condições necessárias e suficientes para que uma solução viável seja mínimo global.

Pode-se definir um problema de otimização global da seguinte maneira [34]: dada uma função contínua $\Phi : f \rightarrow R$ e um conjunto de restrições D, as seguintes questões são apresentadas:

- determinar um limitante inferior de Φ em D;
- determinar um mínimo global de Φ em D, i.e. $\overline{x} \in D$ tal que $\Phi(\overline{f}) \leq \Phi(f)\ \forall\ f \in D$.

A otimização global possui a mesma natureza de objetivos que a otimização local, encontrar, aproximar soluções $\overline{f}$ de D satisfazendo $\Phi(\overline{f}) \leq \Phi(f), \quad \forall f \in D \bigcap N$, onde N é alguma vizinhança de $\overline{f}$. Desta forma em otimização global, a vizinhança é definida como sendo todo o domínio de solução. Dado um conjunto não vazio e fechado $D \subset R^n$ e uma função $\Phi : A \rightarrow R$, onde $A \subset R^n$ é um conjunto que contem D, encontre pelo menos um ponto $f^* \in D$ que satisfaça $\Phi(f^*) \leq \Phi(f)$ para todo $\Phi \in D$ ou mostre que este ponto não existe.

Os métodos de programação matemática são próprios para resolver problemas de otimização convexos em domínios convexos. Esses métodos são apropriados para encontrar o ótimo local "mais próximo"do ponto de partida. Como não existe um critério local capaz de garantir quando uma solução local é global, métodos convencionais de otimização que se utilizam de ferramentas como gradientes, subgradientes, derivadas não são capazes de localizar ou identificar um ótimo global. Muita pesquisa foi e está sendo realizada para resolver classes específicas de problemas de otimização global [37],[36]. Esses problemas são muito difíceis de serem resolvidos e normalmente apenas instâncias pequenas com poucas variáveis são solucionadas.

A abordagem de solução proposta recupera a natureza discreta do problema original de atribuição de capacidades e roteamento e usa os resultados obtidos com a formulação contínua para formular um algoritmo capaz de determinar soluções ótimas globais. Uma abordagem híbrida é adotada aproveitando a equivalência entre a formulação contínua (3.2.1)-(3.2.5) e a discreta (5.1.1)-(5.1.7). Um modelo discreto para o problema de atribuição de capacidades e roteamento pode ser formulado como sendo:

$$\text{minimizar}: \quad \sum_{i=1}^{m} \overline{\tau}_i(f_i, y_i^0, y_i^1, ..., y_i^{n_c}) \tag{5.1.1}$$

$$\text{sujeito a}: \quad \sum_{k=1}^{K}\sum_{h=1}^{N_k} a_i^{kh} x_{kh} = \quad f_i, \quad \forall i = 1, ..., m \tag{5.1.2}$$

$$\sum_{h=1}^{N_k} x_{kh} = \quad d_k, \quad k = 1, ..., K \tag{5.1.3}$$

$$y_i^0 + y_i^1 + ... + y_i^{n_c} = \quad 1, \quad \forall i = 1, ..., m \tag{5.1.4}$$

$$0 \leq f_i \leq \quad c_i^j y_i^j, \quad \forall i = 1, ..., m, \ j \in (0, ..., n_c) \tag{5.1.5}$$

$$x_{hk} \in \quad \Re^+, \quad k = 1, ..., K, \ h = 1, ..., N_k \tag{5.1.6}$$

$$y_i^j \in \quad 0, 1, \quad \forall i = 1, ..., m, \ j \in (0, ..., n_c) \tag{5.1.7}$$

onde:

$$\overline{\tau}_i(f_i, y_i^0, y_i^1, ..., y_i^{n_c}) = \sum_{j=0}^{n_c} (T_i(f_i, c_i^j) + v_i^j f_i + \pi_i^j) y_i^j. \tag{5.1.8}$$

Para cada arco i da rede, foram atribuídas variáveis binárias de decisão ($y_i^0, y_i^1, ..., y_i^{n_c}$) que correspondem à escolha ou não das capacidades $(0, 1, ..., n_c)$ respectivamente. Uma relação coerente é estabelecida considerando que no modelo discreto as funções do custo de congestionamento e do custo de investimento também satisfazem às hipóteses de (3.2.7)-(3.2.10).

Trata-se de um problema de otimização combinatória NP-difícil. Um método de enumeração implícita é desenvolvido para obter a solução ótima global do problema de expansão de capacidades e roteamento de fluxos. Esse método, a princípio, é apropriado para resolver problemas com duas capacidades disponíveis em cada arco. Assim, o problema de expansão de capacidades e roteamento de fluxos fica sendo um problema de programação inteira [0,1].

As proposições apresentadas a seguir consideram que estão disponíveis duas capacidades c_i^0 e c_i^1 para cada arco i.

5.2 Relação entre o modelo contínuo proposto e o modelo discreto

A relação entre a solução ótima do modelo contínuo e a solução ótima do modelo discreto foi formalizada e demonstrada por Souza em [63] [64]. O conjunto de soluções viáveis do modelo contínuo corresponde a um subconjunto de soluções viáveis do modelo discreto, e as soluções ótimas dos dois modelos possuem o mesmo valor.

Todas as soluções viáveis do problema contínuo podem ser transformadas em soluções viáveis do problema discreto. O inverso não é possível. As soluções viáveis do problema discreto em que pelo menos um arco possui $y_i^0 = 0$ e $f_i > \gamma_i^0 c_i^0$ ou $y_i^1 = 1$ e $f_i < \gamma_i^0 c_i^0$ não possuem soluções correspondentes no problema contínuo. Por exemplo, quando para o caso contínuo o fluxo que passa por um arco i é maior que $\gamma_i^0 c_i^0$, o mínimo da função de custos no arco $\tau_i(f_i)$ no problema contínuo acontece em $T_i(f_i, c_i^1) + v_i^1 f_i + \pi_i^1$; esse fato tem como conseqüência a decisão de expandir a capacidade no arco i em contradição com $y_i^0 = 1$.

Proposição 5.2.1. *Suponha que o problema discreto admita solução ótima. Considere que, em uma solução ótima do problema discreto, temos* $y_i^0 = 0$, $y_i^1 = 1$ *e* $f_i = \gamma_i^0 c_i^0$ *para um arco i, então existe uma outra solução ótima para este mesmo arco* i, *com* $y_i^0 = 1$, $f_i = \gamma_i^0 c_i^0$, $y_i^1 = 0$ *e vice-e-versa; ou seja,é indiferente fazer ou não a expansão(redução) da capacidade do arco. A expansão da capacidade de um arco i com capacidade atual* c_i^0 *é feita se, e somente se, o fluxo total que passa por* i, f_i *é maior que* $\gamma_i^0 c_i^0$.

Proposição 5.2.2. *Considerando as formulações discreta e contínua, exatamente uma das duas alternativas a seguir é possível:*

- *se um dos problemas discreto ou contínuo admite solução ótima, o outro admite também; e mais, elas possuem o mesmo valor;*
- *se um dos problemas não admite solução viável, o outro também não admite.*

5.3 Abordagem de solução global

O método de solução proposto combina limites inferiores e superiores obtidos adotando-se a formulação contínua com um método da otimização combinatória apropriado para

resolver problemas com variáveis binárias [27]. O algoritmo de enumeração implícita é um algoritmo "elegante"que gasta pouca memória e é naturalmente apropriado para ser paralelizado. Ele foi proposto por Balas em [7] originalmente para resolver o problema de programação linear inteira com variáveis binárias [27].

5.3.1 Método de enumeração implícita

O método de enumeração implícita consiste na avaliação sistemática de todas as soluções sem a necessidade de se calcular explicitamente cada uma delas. Esse método pode ser obrigado a enumerar todas as soluções no pior caso.

Uma idéia da busca em profundidade e um procedimento de retorno de trilha (*back-track*) são a base do método de enumeração implícita. Esse método é dito ser equivalente a um método *branch and bound*; entretanto, as primeiras versões do *branch-and-bound* eram do tipo a melhor-primeiro (*best-first*) fazendo uma busca em largura na árvore de busca. Por outro lado, o método de enumeração implícita faz busca em profundidade no grafo de soluções.

Para explicar o funcionamento do algoritmo, será dado um exemplo. Considere um problema com cinco arcos $m = 5$ que podem ter sua capacidade expandida. Esse caso possui $2^m = 32$ soluções possíveis. As soluções são enumeradas implicitamente pela consideração de grupos de soluções conjuntamente.

O conjunto de soluções pode ser dividido em $m+1$ subconjuntos, de maneira que o k-ésimo subconjunto ($q = 0, 1, ..., m$) contém todas as soluções com exatamente q componentes de Y sendo iguais a 1.

As relações entre os elementos do conjunto de soluções pode ser descrita com a ajuda de um grafo G. Um nó r de G é associado a cada solução Y^r, e um arco (r, s) é associado a cada par de soluções Y^r Y^s.

O grafo de soluções G, representado na Figura 5.1, apresenta todas as soluções possíveis e todas as transições entre soluções à medida que o valor 1 é atribuído a alguma das variáveis binárias. Por exemplo: o nó 124 representa a solução $Y^{124} = (1, 1, 0, 1, 0)$, ou seja $y^1 = 1$, $y^2 = 1$, $y^3 = 0$, $y^4 = 1$, $y^5 = 0$. Partindo de uma solução como esta, podemos obter as soluções: $(1, 2, 3, 4)$, $(1, 2, 4, 5)$ e $(1, 2, 3, 4, 5)$.

O princípio de funcionamento da enumeração é a construção de uma árvore no grafo de soluções G. Começando do nó 0, um arco $(0, k)$ é escolhido. O valor 1 é

atribuído a uma variável de acordo com regras pré-determinadas, assim que o algoritmo alcança um novo nó r. Toda vez que um novo nó r é alcançado, a solução correspondente é submetida a testes que determinam se pode existir solução melhor do que a já encontrada até o momento. Se a solução passa nos testes, o procedimento continua com a escolha de um novo arco, (r, s). Caso não, então r é abandonado com todos os seus descendentes, e retorna-se ao último nó visitado antes de r, q. O procedimento recomeça do nó q, mas as regras de construção da árvore são tais que nenhum descendente de um nó que foi abandonado pode ser revisitado. A Figura 5.1 ilustra o grafo de solução de uma problema com cinco variáveis. Por exemplo: se o nó 5 na Figura 5.2 for abandonado, então junto com ele metade de todas as soluções é excluída de qualquer investigação. O grafo resultante do abandono de 5 é apresentado na Figura 5.2. A eficiência do algoritmo obviamente dependerá do tamanho da árvore de solução a ser construída depois que uma solução ótima é encontrada ou quando a ausência de uma solução viável é estabelecida.

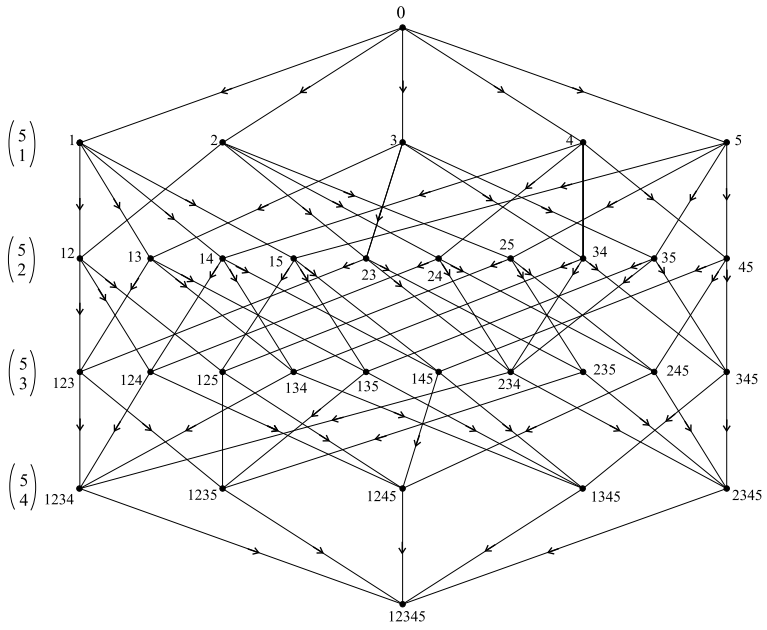


Figura 5.1: Grafo de decisão

Para explicar como grupos de solução são definidos, é necessária a definição de solução parcial. Uma solução parcial S é definida como uma atribuição de valores binários a um subconjunto de $\overline{m}$ variáveis. O conjunto S contém os sub-problemas (nós) que ainda podem ser decompostos. Qualquer variável à qual não está assinalado

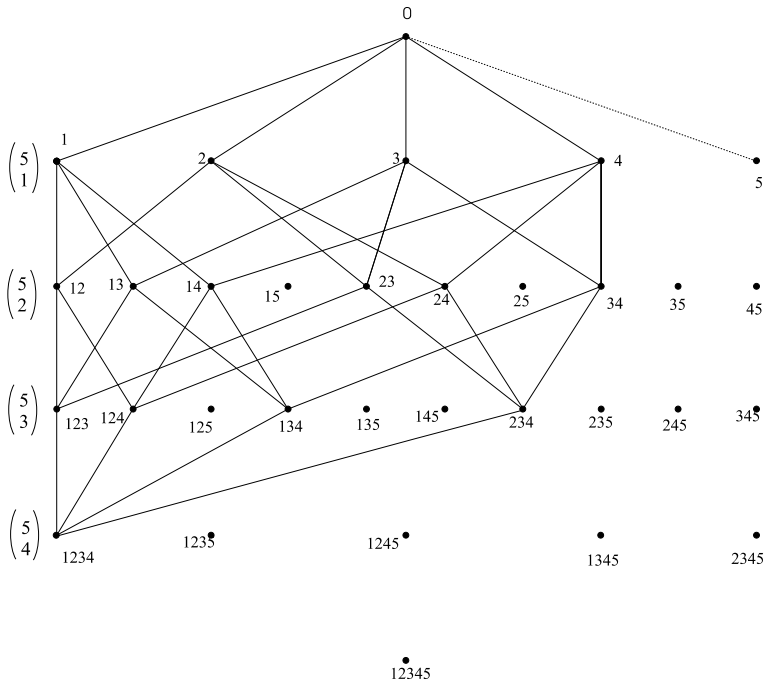


Figura 5.2: Grafo de decisão depois que se verificou que $y^5 = 0$

um valor é denominada livre. Uma completação de uma solução parcial S é definida como uma solução determinada por S com uma especificação de valores de S para as variáveis livres. A convenção de que o símbolo j define $y_j = 1$ e o símbolo $-j$ define $y_j = 0$ é adotada. Assim, se $\overline{m} = 5$ e $S = 3, 5, -2$, então $y_3 = 1$, $y_5 = 1$, $y_2 = 0$, e y_1 e y_4 estão livres. A ordem na qual os elementos de S aparecem corresponde à ordem pela qual eles foram gerados. Uma solução determinada por S juntamente com a especificação dos valores das variáveis livres define uma completação da solução parcial S. No exemplo, as quatro completações possíveis para S são: $\{0, 0, 1, 0, 1\}$, $\{0, 0, 1, 1, 1\}$, $\{1, 0, 1, 0, 1\}$, $\{1, 0, 1, 1, 1\}$.

Assim, uma solução parcial S com s elementos determina um conjunto de $2^{\acute{m}-s}$ diferentes completações. Quando não existem variáveis livres, há somente uma completação de S: a solução S propriamente dita.

Outro conceito básico do algoritmo é que quando uma solução parcial S é dada, podemos:

- determinar a melhor completação viável de S, mesmo se não for possível determinar a melhor completação. Verificar se alguma das soluções obtidas com a completação é melhor do que qualquer outra solução já obtida. Se for, ela passa

a ocupar o cargo de incumbente da melhor solução até então encontrada para o problema;

- determinar que S não tem nenhuma completação melhor que a incumbente corrente.

Caso ocorra qualquer uma dessas situações podemos, cortar (*fathom*) a solução S. O corte é feito da seguinte forma: localize o elemento mais à direita do conjunto S que não esteja sublinhado. Se não houver nenhum, termine; de outra forma, substitua este elemento pelo seu complemento sublinhado $j \rightarrow \underline{-j}$ e elimine todos os elementos à direita. Ou seja, quando um elemento está sublinhado todas as completações viáveis de seu complemento já foram implicitamente enumeradas.

Todas as completações viáveis de um S que foi cortado foram enumeradas implicitamente no sentido de que elas podem ser excluídas de qualquer outra consideração, a menos, é claro, se a melhor completação viável obtida for melhor do que a solução incumbente.

Caso nenhuma das duas situações tenha ocorrido, deve-se aumentar S à direita de j ou $-j$.

O procedimento básico que gera uma seqüência $\langle S \rangle$ de soluções parciais não redundantes e que termina somente depois que todas as 2^m soluções do problema foram (implicitamente) enumeradas é apresentado [7]:

Algoritmo de enumeração implícita clássico

Passo 1 - Comece com $S^0 = \varnothing$, onde $\varnothing$ indica um conjunto vazio.

Passo 2 - Tente cortar S. O corte pode ser feito de duas maneiras: encontre a melhor completação viável de S. Essa melhor completação substitui a solução incumbente. Determine que S não possui nenhuma completação viável melhor do que a solução incumbente.

Passo 3 - Foi possível cortar S? Se sim, vá para o Passo 5. Se não, vá para o **Passo 4**.

Passo 4 - Aumente S à direita de j ou $-j$, onde j significa atribuir o valor 1 à variável y_j e $-j$ significa atribuir o valor 0 à variável y_j. Retorne ao **Passo 2**.

Passo 5 - Se a melhor completação viável de S foi encontrada e se ela é melhor que a solução incumbente, armazene-a como nova incumbente.

Passo 6 - Localize o elemento mais à direita de S que não esteja sublinhado. Se nenhum existe, termine; de outra forma, substitua o elemento pelo seu complemento sublinhado $k \rightarrow \underline{-k}$ e elimine todos os elementos à direita. Retorne ao **Passo 2**.

Esse procedimento gera uma seqüência não-redundante de soluções parciais, tentativas que terminam somente quando todas as 2^m (onde m é o número de arcos) soluções já tiverem sido enumeradas implicitamente [27].

Observações:

- m é o número de arcos e a enumeração de cada uma das 2^m soluções possíveis pode ser implícita ou explícita;
- S inicial pode ser qualquer solução parcial sem variáveis sublinhadas ou um conjunto vazio;
- No **Passo 2** a solução de um problema relaxado é uma maneira de determinar que S não possui nenhuma completação viável melhor do que a solução incumbente;
- S pode ser aumentada no **Passo 4** por uma coleção de elementos em vez de um só elemento;
- qualquer seqüência consecutiva de elementos não sublinhados em S pode ser permutada arbitrariamente a qualquer momento;
- S pode ser aumentado no **Passo 4** por uma coleção de elementos sublinhados quando isso não exclui alguma completação viável de S que seja melhor que a incumbente.

Partindo desse procedimento clássico, foi desenvolvido um algoritmo de enumeração implícita para resolver o problema de expansão de capacidades e roteamento. O algoritmo, mesmo sendo simples, apresenta bons resultados, conforme será discutido através de experiências numéricas.

5.3.2 Algoritmo de enumeração implícita para o problema de expansão de capacidades e roteamento de fluxos

Algumas perguntas precisam de resposta durante o desenvolvimento de um algoritmo de enumeração implícita específico para o problema de atribuição de capacidades e roteamento.

Como definir uma solução parcial para esse problema?

Uma solução parcial S é definida como a atribuição de uma capacidade a cada um dos arcos de um subconjunto $\overline{m}$ de arcos da rede. Os demais arcos permanecem sem uma capacidade definida. Para esses arcos, a função de custos é a aproximação convexa da função objetivo contínua.

Como encontrar a melhor completação viável de uma solução parcial S?

Determinar qual é a melhor completação viável de uma solução parcial S do problema de atribuição de capacidades e roteamento de fluxos é tão difícil quanto resolver o problema original. No algoritmo proposto, não será determinada a melhor completação viável de S. Entretanto, usando os procedimentos propostos no capítulo 3, um bom limite superior para as completações viáveis de uma solução parcial S pode ser determinado.

Como determinar que S não possui nenhuma completação melhor que a incumbente?

É possível calcular o limite inferior de todas as completações viáveis de uma dada solução parcial fixando as capacidades dos arcos para os quais foram atribuídos valores na solução parcial e para os demais arcos adotando a aproximação convexa da função objetivo. Resolvendo o problema convexo resultante obtém-se um limite inferior para o valor de todas as completações viáveis desta solução parcial. Se o valor desse limite inferior for superior ao valor da solução incumbente, pode-se garantir que a solução parcial S não possui nenhuma completação melhor que a incumbente (coração do algoritmo).

Qual solução parcial inicial S^0 adotar?

O conjunto vazio é adotado. Testes com a solução parcial inicial sendo a solução obtida através da adoção de um dos algoritmos com desempenho garantido não indicaram melhoria.

Que regra adotar para acrescentar uma variável a uma solução parcial?

Esta regra oferece a possibilidade de se agregar conhecimento sobre o problema para auxiliar na solução do mesmo. Como esta regra é essencialmente empírica diversas estratégias foram experimentadas. Entre elas:

- escolha aleatória;
- a variável candidata para entrar na solução parcial como sendo a variável livre com o maior ou o menor custo marginal;
- a variável candidata para entrar na solução parcial como sendo aquela que, se for fixada, aumenta mais o limite inferior da solução parcial correspondente.

A regra que se mostrou mais promissora foi escolher a variável livre candidata como sendo a que representa o arco de maior fluxo.

Foi feita uma modificação na convenção que especifica os valores de uma solução parcial S, isto é, no lugar de adotar a convenção de que o símbolo j define $y_j = 1$, e o símbolo $-j$ define $y_j = 0$. Primeiro obtém-se um bom limite superior aplicando um dos algoritmos propostos no capítulo 3. O valor de cada uma das variáveis da solução correspondente é que define a convenção. Assim, o símbolo j define $y_j = 1$ se $\ddot{y}_j = 1$ na solução obtida, ou define $y_j = 0$ se $\ddot{y}_j = 0$, e o símbolo $-j$ define $y_j = 0$ se $\ddot{y}_j = 0$, ou $y_j = 1$ se $\ddot{y}_j = 1$.

O seguinte algoritmo é proposto adotando o conjunto de regras descritas acima:

Algoritmo de enumeração implícita adaptado para o problema CFA

Passo 1 - Comece com $S^0 = \varnothing$, onde $\varnothing$ indica um conjunto vazio.

Passo 2 - Determine o limite superior e o inferior para o problema aplicando um dos algoritmos propostos no capítulo 3 considerando como fixas as capacidades dos arcos que estão no conjunto S. Armazene a primeira solução obtida $\hat{S}$.

Passo 3 - É possível cortar S? Se sim, vá para o Passo 5. Se não, vá para o **Passo 4**.

Passo 4 - Aumente S à direita de j ou $-j$, de acordo com a convenção adotada. Retorne ao **Passo 2**.

Passo 5 - Se o limite superior de S obtido for melhor que o valor da solução incumbente, armazene a solução correspondente a esse limite como sendo a nova solução incumbente.

Passo 6 - Localize o elemento mais à direita de S que não esteja sublinhado. Se nenhum existe, termine; de outra forma, substitua o elemento pelo seu complemento sublinhado $j \rightarrow \underline{-j}$ e elimine todos os elementos à direita. Retorne ao **Passo 2**.

5.4 Experimentos numéricos

Experimentos numéricos foram feitos para estudar o algoritmo proposto e a influência dos diferentes parâmetros nas soluções obtidas bem como para verificar sua eficiência na obtenção da solução ótima global.

O método de solução foi codificado em C e compilado com GCC 2.95.3. Os experimentos foram realizados usando uma estação SUN Blabe Spark 500 MHz com 1 GigaByte de RAM rodando o sistema operacional Solaris.

Uma variação do método de tangentes paralelas "PARTAN"[20] foi adotada para resolver os subproblemas de roteamento que ocorrem no método proposto. Este é uma adaptação do método de Frank e Wolfe que foi adaptado por Fratta et al [21] para resolver problemas de roteamento multiproduto com função de custo não-linear. A opção de adotar tal método é motivada pelo fato de que ele converge mais rapidamente que o de desvios de fluxos, além de ser um método de segunda ordem. O PARTAN foi desenvolvido para resolver problemas diferenciáveis e sem restrições, entretanto ele pode ser adaptado para receber restrições [48]. Nos experimentos foi fixada uma precisão de 1%.

Três topologias diferentes foram usadas nos experimentos computacionais: a rede CNET, representada na Figura 5.3; a rede RING, apresentada no capítulo 4, e a rede NTS100, no capítulo 3. A Tabela 5.1 apresenta as características dessas redes.

Dois conjuntos de testes foram realizados com essas topologias. O primeiro conjunto de testes foi realizado com a rede CNET [56]. O objetivo desses testes foi verificar a influência do aumento de demanda e do salto de capacidade entre a capacidade já instalada e a capacidade disponível para expansão (c_i^1/c_i^0) no número de iterações do algoritmo exato. O segundo conjunto de testes foi realizado sobre as

Tabela 5.1: Parâmetros e propriedades topológicas das redes adotadas

Rede ID	nós	arcos	Pares-OD	grau médio do nó	diâmetro
	n	m	K	$2m/n$	
CNET	19	34	38	3.36	4
RING	32	60	496	3.75	6
NTS100	100	187	2000	3.74	11

demais topologias com o objetivo de verificar a adequação da metodologia proposta ao problema de expansão de capacidades em redes de comunicação.

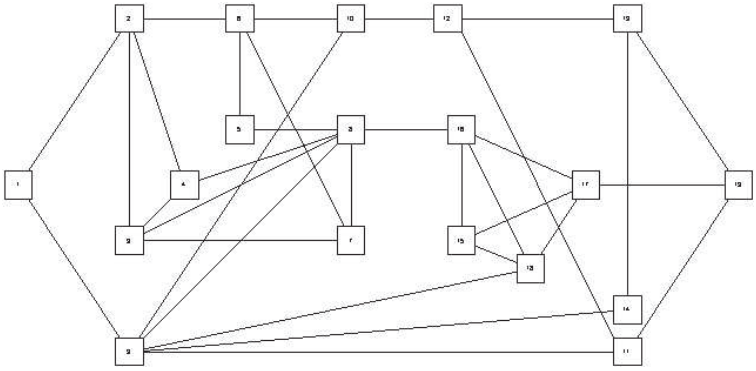


Figura 5.3: Topologia de rede CNET com 19 nós e 34 arcos

As seguintes definições foram adotadas:

- ϕ^* valor ótimo global;
- $\ddot{\phi}$ valor ótimo da aproximação convexa;
- $\phi_{[pc,sw]}$ resultado obtido apliaando um dos algoritmos apresentados no capítulo 4;
- N_{max} número de nós necessariamente pesquisados para garantir a otimalidade global;
- N_{min} número mínimo de nós para encontrar o ponto ótimo;
- $\alpha = \frac{\phi^*}{\ddot{\phi}}$;
- $\alpha_{[pc,sw]} = \frac{\phi_{[pc,sw]}}{\ddot{\phi}}$.

Tabela 5.2: Problema teste CNET, caso D38510.70

$\ddot{\phi}$	ϕ^*	$\phi_{[pc,sw]}$	α	$\alpha_{[}pc, sw]$	N_{max}	N_{min}	Demanda
13.64	13.84	13.86	1.01	1.01	26	24	0.50
37.68	39.21	39.43	1.04	1.05	206	178	1.00
68.78	70.59	71.54	1.02	1.04	152	66	1.50
181.94	184.19	184.99	1.01	1.02	146	110	2.00

Número de produtos 38, $c_0 = 5$,$c_1 = 10$, $\gamma = 0.70$

Tabela 5.3: Problema teste CNET, caso D38510.90

$\ddot{\phi}$	ϕ^*	$\phi_{[pc,sw]}$	α	$\alpha_{[}pc, sw]$	N_{max}	N_{min}	Demanda
13.64	13.83	13.86	1.01	1.02	0	0	0.50
56.45	56.97	59.67	1.01	1.06	18	16	1.00
135.06	140.32	144.10	1.04	1.07	92	27	1.50
291.38	301.38	304.86	1.03	1.05	208	112	2.00

Número de produtos 38, $c_0 = 5$,$c_1 = 10$, $\gamma = 0.90$

5.4.1 Primeiro conjunto de testes

Um conjunto de experimentos foi realizado sobre a rede CNET apresentada na Figura 5.3. Nesses experimentos não se especificaram as grandezas físicas envolvidas, e o custo de congestionamento foi fixado como sendo $\rho = 1$. Todos os arcos da rede possuem a mesma capacidade instalada. Com o custo de congestionamento prédefinido, para cada valor de γ igual a 0.7 e 0.9, o custo de expansão da capacidade no arco i, π_i^1 será obtido por 5.4.1:

$$\pi_i^1 = \rho\gamma c_i^0 \frac{(c_i^1 - c_i^0)}{(c_i^0 - \gamma c_i^0)(c_i^1 - \gamma c_i^0)} \tag{5.4.1}$$

Como se trata de um problema de expansão de capacidades, a capacidade inicial do arco já está instalada $\pi_i^0 = 0$.

Nas Tabelas 5.2 a 5.5 são apresentados os resultados obtidos para a rede CNET variando as demandas, o γ e a capacidade disponível para expansão.

Tabela 5.4: Problema teste CNET, caso D38520.70

$\ddot{\phi}$	ϕ^*	$\phi_{[pc,sw]}$	α	$\alpha_{[}pc, sw]$	N_{max}	N_{min}	Demanda
13.64	13.84	13.86	1.01	1.02	168	161	0.50
24.77	31.66	34.99	1.27	1.41	2893	272	1.00
37.38	45.51	49.35	1.21	1.32	6194	968	1.50
50.82	59.78	63.02	1.17	1.24	8026	2048	2.00

Número de produtos 38, $c_0 = 5$,$c_1 = 20$, $\gamma = 0.70$

Tabela 5.5: Problema teste CNET, caso D38520.90

$\ddot{\phi}$	ϕ^*	$\phi_{[pc,sw]}$	α	$\alpha_{[}pc, sw]$	N_{max}	N_{min}	Demanda
13.64	13.84	13.86	1.01	1.02	4	2	0.50
42.05	49.97	54.41	1.19	1.29	286	268	1.00
75.52	95.14	112.54	1.26	1.49	2672	2248	1.50
110.08	134.26	151.42	1.22	1.37	2944	317	2.00

Número de produtos 38, $c_0 = 5$,$c_1 = 20$, $\gamma = 0.90$

Observando os resultados obtidos verifica-se que o algoritmo é sensível ao degrau de expansão de capacidade. Quando $c_i^1 \geq 4c_i^0$ o número de iterações aumenta significativamente. O aumento de γ também afeta a eficiência do algoritmo. Ambos efeitos podem ser explicados observando α e $\alpha_{[pc,sw]}$. Como o algoritmo exato depende da solução do problema convexo para determinar limites inferiores e podar a árvore de soluções, quanto pior a aproximação convexa, maior o número de soluções a serem testadas. Observa-se também que o ótimo global sempre foi obtido antes do término da execução, em algumas instâncias bem antes $N_{max} \gg N_{min}$.

5.4.2 Segundo conjunto de testes

O segundo conjunto de testes aplica o algoritmo proposto em problemas com redes heterogêneas que representam melhor os problemas reais de expansão de capacidade. Essa seqüência de experimentos foi feita da seguinte forma: partindo de topologias escolhidas e aplicando os algoritmos propostos no Capítulo 3, uma solução viável para a atribuição de capacidades e roteamento foi obtida. A demanda inicial entre cada par origem-destino de cada rede foi pré-estabelecida $10[Mbits]$ para a rede Ring e $3[Mbits]$ para a rede NTS100. As capacidades disponíveis para instalação são apresentadas na Tabela 5.6.

Partindo da atribuição de capacidades e das demandas iniciais, a rede é submetida a um aumento médio de demanda. A expansão das capacidades nos arcos ocorre, uma vez que é economicamente mais interessante expandir as capacidades do que suportar o aumento de congestionamento.

O aumento de demanda foi feito da seguinte forma: em um primeiro conjunto de experimentos com a rede RING, as demandas sofrem um aumento uniforme de 20%,40%,60%,80% e 100%. Os resultados destes testes são apresentados nas Tabelas 5.7. Em um segundo conjunto de experimentos o aumento de demanda não é mais

Tabela 5.6: Capacidades disponíveis e seus respectivos custos

Capacidade c $[Mbps]$	Custo de instalação S_c $[\$/ano]$	Custo de distância D_c $[\$/ano/km]$
2	1750	40
10	2800	50
34	4800	55
155	10000	80
300	14000	90
622	21000	120
922	35000	210

$\pi^c = S^c + D^c d_{ij}$

uniforme entre todos os pares origem-destino das redes RING e NTS100. Nesse caso, o aumento obedece à seguinte regra: 25% das demandas sofrem um aumento 50% maior do que o aumento médio, 25% das demandas sofrem um aumento 50% inferior ao aumento médio, e as 50% restantes aumentam pelo aumento médio. Por exemplo, um aumento médio na demanda base de 25% é distribuído da seguinte maneira: 25% dos pares origem-destino têm sua demanda aumentada de 37.5%, 50% dos pares origem-destino têm sua demanda aumentada de 25%, e os 25% pares origem-destino restantes têm sua demanda aumentada de 12.5%. Os resultados desses testes são apresentados nas Tabelas 5.8 e 5.9. Uma distribuição inicial do aumento de demanda entre os pares origem-destino foi definida de forma aleatória e fixada para cada problema. Os custos de congestionamento adotados foram $\rho \in 500, 1000, 5000, 10000[\$/mês/mensagem]$.

5.4.3 Comentários

Um número de iterações muito modesto comparado ao espaço de busca foi necessário para resolver as instâncias propostas. Considerando o número de soluções realmente avaliadas e o número total de soluções possíveis de cada problema estudado (CNET $2^{34} = 17179869184$ soluções, RING 2^{60} soluções, NTS100 2^{187} soluções), pode-se avaliar a eficácia do método proposto. Os problemas do segundo conjunto de testes se mostraram mais "fáceis"; isso pode ser explicado observando que, para esses problemas, a qualidade da aproximação convexa do problema original é muito boa. Na maioria dos casos estudados, o algoritmo encontrou o ótimo antes de terminar sua execução.

Cada iteração do algoritmo proposto consiste na solução de um problema de roteamento de fluxos multiproduto não-linear. Esse problema sozinho pode ser muito

Tabela 5.7: Resultados obtidos com a rede RING e aumento de demanda uniforme

ρ	$\ddot{\phi}$ [\$] $\times 10^6$	ϕ^* [\$] $\times 10^6$	$\phi_{[pc,sw]}$ [\$] $\times 10^6$	α	$\alpha_{[pc,sw]}$	Aumento médio de demanda %
1000	3.69	3.71	3.71	1.00	1.00	10
1000	3.84	3.91	3.92	1.01	1.02	20
1000	4.05	4.11	4.15	1.01	1.02	40
1000	4.35	4.39	4.49	1.01	1.01	60
1000	4.70	4.76	4.76	1.01	1.01	80
1000	5.13	5.13	5.13	1.00	1.00	100
5000	4.22	4.26	4.26	1.01	1.01	10
5000	4.47	4.50	4.50	1.00	1.00	20
5000	4.79	4.84	4.84	1.01	1.01	40
5000	5.17	5.23	5.23	1.01	1.01	60
5000	5.62	5.68	5.69	1.01	1.01	80
5000	6.31	6.35	6.37	1.01	1.01	100
10000	4.61	4.62	4.62	1.00	1.00	10
10000	4.95	5.00	5.00	1.01	1.01	20
10000	5.36	5.42	5.42	1.01	1.01	40
10000	5.84	5.87	5.87	1.00	1.00	60
10000	6.38	6.42	6.42	1.00	1.00	80
10000	7.03	7.08	7.10	1.01	1.01	100

Tabela 5.8: Resultados obtidos com a rede RING e aumento de demanda heterogênio

ρ	$\ddot{\phi}$ [\$] $\times 10^6$	ϕ^* [\$] $\times 10^6$	$\phi_{[pc,sw]}$ [\$] $\times 10^6$	α	$\alpha_{[pc,sw]}$	N_{min}	N_{max}	Aumento médio de demanda %
500	3.70	3.82	3.85	1.01	1.02	32	138	25
500	4.06	4.14	4.27	1.025	1.05	84	98	50
500	4.88	4.93	4.93	1.01	1.00	2	4	100
1000	3.89	3.97	3.97	1.01	1.02	4	265	25
1000	4.21	4.33	4.33	1.01	1.02	1	2	50
1000	5.08	5.12	5.13	1.00	1.01	1	2	100
5000	4.01	4.02	4.03	1.01	1.00	5	555	25
5000	4.97	5.02	5.02	1.01	1.01	1	2	50
5000	6.30	6.37	6.37	1.01	1.01	1	2	100
10000	4.87	4.91	4.92	1.01	1.01	6	365	25
10000	5.59	5.64	5.64	1.01	1.01	3	1	50
10000	7.06	7.07	7.07	1.00	1.01	5	1	100

Tabela 5.9: Resultados obtidos com a rede NTS100 e aumento de demanda heterogêneo

ρ	$\ddot{\phi}$ [\$] $\times 10^6$	ϕ^* [\$] $\times 10^6$	$\phi_{[pc,sw]}$ [\$] $\times 10^6$	α	$\alpha_{[pc,sw]}$	N_{min}	N_{max}	Aumento médio de demanda %
500	2.31	2.41	2.41	1.04	1.05	163	333	25
500	2.43	2.55	2.57	1.05	1.06	37	259	50
500	2.73	2.81	2.81	1.03	1.03	1	22	100
1000	2.48	2.58	2.58	1.04	1.05	21	374	25
1000	2.62	2.71	2.72	1.03	1.04	25	313	50
1000	2.95	3.03	3.04	1.02	1.03	32	100	100
5000	3.05	3.06	3.06	1.01	1.01	678	2450	25
5000	3.35	3.45	3.45	1.02	1.03	485	1760	50
5000	3.80	3.95	3.95	1.03	1.04	180	810	100
10000	3.40	3.41	3.41	1.01	1.01	789	2348	25
10000	3.91	4.05	4.05	1.03	1.04	119	2219	50
10000	4.49	4.60	4.61	1.02	1.02	237	519	100

difícil, principalmente se a carga na rede estiver próxima da capacidade máxima quando começam a aparecer problemas de viabilidade. A opção por aplicar o PARTAN (método de segunda ordem) se deveu principalmente ao ganho na velocidade na resolução dos problemas para uma precisão pré-fixada em 1%. Cada problema multiproduto gastou de algumas frações de segundo a alguns minutos.

Foi verificado que o problema possui um número elevado de ótimos locais com valores da função objetivo muito próximos ou iguais. Esse fato dificulta a aplicação de metaheurísticas na solução de tal problema. Além disso, os algoritmos com desempenho garantido apresentados no capítulo 3 se mostraram capazes de fornecer soluções consistentes com poucas iterações.

A busca em profundidade possui necessidades muito modestas de memória. Ela precisa armazenar somente o caminho do nó raiz ao nó folha. Como o problema estudado tem muitas soluções, a busca em profundidade é mais rápida do que a busca em largura, porque ela tem boa chance de encontrar uma solução depois de explorar uma pequena porção de todo o espaço de busca. O algoritmo proposto é apropriado para ser paralelizado e acredita-se que seja possível obter uma boa escalabilidade.

O algoritmo proposto pode ser especializado ainda mais se informações adicionais sobre os problemas forem consideradas. Por exemplo, em problemas reais existem limitações orçamentárias e operacionais que podem restringir bastante o conjunto dos arcos candidatos à expansão.

O procedimento pode ser utilizado para melhorar soluções obtidas para problemas

com mais de duas capacidades disponíveis por arco. Para tanto, pode-se resolver um algoritmo de aproximação e então utilizar para cada arco a capacidade instalada e a capacidade imediatamente inferior. Solucionando o problema de otimização global obtido, o resultado é um ótimo local do problema original. O procedimento proposto também pode ser usado no contexto de redução de capacidades.

A literatura é escassa em abordagens exatas de solução do problema de atribuição de capacidades discretas e roteamento de fluxos. Basicamente duas linhas de pensamento têm sido adotadas: o método de decomposição de Benders [22] [12] e métodos para a solução de problemas DC [50]. Os resultados obtidos demonstram que algoritmo de enumeração implícita proposto pode ser considerado como uma boa alternativa à disposição de quem deseja resolver o problema CFA discreto.

Capítulo 6

Conclusão

6.1 Comentários finais

No capítulo 2 foram apresentados modelos e formulações contínuas para o problema de atribuição de capacidades e roteamento de fluxos contínuo. Nessas formulações a atribuição de capacidades e o roteamento de fluxos é feito com o objetivo de minimizar os custos de instalação de capacidades e o atraso médio, enquanto satisfaz as demandas de comunicação. Essas abordagens tratam da minimização de dois critérios e o objetivo é obter soluções Pareto-eficientes [47]. A abordagem de solução de problemas multi-critérios é minimizar um dos critérios sujeito a um limite superior do outro. No problema em questão, a abordagem consiste em minimizar o atraso médio sujeito a um limite de gastos de instalação de capacidade, ou em minimizar custos de instalação de capacidade sujeito a um atraso médio máximo. O novo modelo proposto no Capítulo 3 trata o atraso médio na rede como sendo custo de congestionamento. Isso é feito com o auxílio de uma constante de proporcionalidade ρ que quantifica o atraso da rede. O novo modelo integra custos de congestionamento e custos de instalação de capacidade em uma única função objetivo contínua.

Se a rede estiver saturada, existe um custo social de enviar um novo pacote à medida que aumenta o tempo de resposta, a probabilidade de estouro de pilha, a taxa de perdas de pacotes, a probabilidade de ocorrência de clientes não atendidos e insatisfeitos, a probabilidade de falhas e o consumo de energia. Estas complicações podem ser modeladas como um custo de congestionamento não linear.

Embora o custo de congestionamento não seja pago pelo provedor de serviço de rede, ele é suportado pelos usuários do serviço. O tempo gasto pelo usuário esperando

por uma transferência de arquivo é um custo social, e deve ser reconhecido como tal em qualquer contabilidade econômica. Não foi encontrado na literatura um modelo matemático que possa inferir o custo do congestionamento em redes de comunicação. Dessa forma nos experimentos realizados foi necessário testar diversos valores para obter uma qualidade de serviço pré-estabelecida. Diversos autores que foram referenciados adotam esta estratégia [2], [5].

Em problemas de expansão de capacidade o tempo é um fator importante. No sentido de que perguntas como: existe uma contínua necessidade por facilidades? as facilidades ou equipamentos adicionados são duráveis? Assim, uma questão importante é decidir qual deve ser o critério de decisão a ser adotado. Neste trabalho foi considerada uma abordagem estática onde o planejamento de capacidades é feito considerando que as expansões de capacidades de todos os arcos são feitas de uma só vez.

Outra hipótese adotada é que a demanda por capacidade é determinada independentemente das decisões de expansão de capacidade. Foi considerado também que o aumento de demanda é conhecido.

6.2 Conclusões

O objetivo central deste trabalho foi estudar a formulação proposta por Luna e Mahey [49] para o problema de expansão de capacidades e roteamento de fluxos para múltiplas expansões de capacidade. A aproximação convexa da nova função de custos integrada proposta é um resultado chave. Essa aproximação permite a obtenção de bons limites inferiores que permitiram o desenvolvimento de heurísticas e de um método de otimização global para a solução do problema. Os resultados experimentais obtidos demonstram que as metodologias adotadas são competitivas quando comparadas com outros resultados encontrados na literatura, tanto em tempo de execução, considerando que é possível resolver instâncias maiores, quanto na qualidade das soluções obtidas.

A formulação proposta caminha em uma direção original ao propor uma fomulação contínua para um problema discreto. O preço pago foi que o modelo não é diferenciável e nem sub-diferenciável, apresentando quinas. Como resultado disto foi

necessário o desenvolvimento de algoritmos específicos para lidar com a nova formulação.

Os resultados obtidos indicam que as heurísticas propostas se beneficiam do aumento do número de capacidades disponíveis para instalação. Se por um lado este fato favorece a aplicação das heurísticas propostas neste trabalho, por outro lado o mesmo fato dificulta a aplicação das heurísticas presentes na literatura. Pode-se explicar este efeito considerando que para as demais abordagens há um aumento de complexidade do problema quando há um aumento do número de capacidades. O mesmo não acontece com a nova metodologia proposta, que trabalha essencialmente com uma única função convexificada, independentemente do número de capacidades instaladas.

A principal conclusão do trabalho é de que a metodologia proposta é eficiente e eficaz na solução do problema de atribuição de capacidades e roteamento de fluxos.

6.3 Perspectivas

Com relação às perspectivas de estudos futuros relacionados com o presente trabalho pode-se citar:

1. Modelos e formulações analíticas de qualidade de serviço em redes são escassos e empíricos e as abordagens de simulação são numericamente muito dispendiosas para a aplicação mesmo em problemas com poucas dezenas de nós. Desta maneira, o estudo de modelos matemáticos para quantificar as métricas de qualidade de serviço em redes de comunicação permanece sendo um campo aberto para investigação.

2. A estimativa dos custos de congestionamento em redes viárias não é um assunto novo. Estes custos constituem a base para regras de restrição de circulação em grandes cidades e para taxas de pedágio. Por outro lado, o estudo da estimativa de custos de congestionamento em redes de comunicação é um problema em aberto.

3. Não existe muita pesquisa publicada sobre o problema de roteamento multiproduto não-bifurcado, não-linear e capacitado sendo um campo de pesquisa em aberto.

4. O algoritmo de enumeração implícita proposto no capítulo 5 pode ser paralelizado.

5. Nos problemas formulados e resolvidos a topologia da rede é suposta conhecida. Cabe estudar extensões da abordagem adotada que possam incorporar o importante problema de topologia da rede.

6. O problema de roteamento em redes com custos côncavos continua sendo um um dos desafios da programação matemática.

7. Estudar redes onde os produtos possuam diferentes requisitos de qualidade de serviço, redes integradas de serviço.

Apêndice A

Modelo e algoritmo para o problema contínuo de atribuição de capacidades e roteamento de fluxos

A.1 Novo modelo contínuo

Esta seção apresenta uma formulação contínua da formulação integrada para o problema de atribuição de capacidades e roteamento proposta por Luna e Mahey [49] para o problema com capacidades discretas.

Quarto modelo contínuo $[CFA4]$:

Conhecendo: a topologia da rede,
os custo de instalação de capacidade em cada arco $\theta_i(c_i)$,
a parcela do atraso médio total de cada arco $T_i(f_i, c_i)$,
o vetor de demandas máximas,
custo de uma unidade de atraso ρ (unidade de custo/tempo),

Determinar: as capacidades que minimizem $\phi(c, f) = \sum_{i=1}^{n} \{\theta_i(c_i) + \rho T_i(f_i, c_i)\}$,

Variáveis: os fluxos f_i e as capacidades c_i nos arcos.

Sujeito a: o fluxo nos arcos não pode ultrapassar a capacidade do arco,
satisfazer todas as demandas,
as restrições de fluxo através da rede.

Podemos formular esse modelo da seguinte forma:

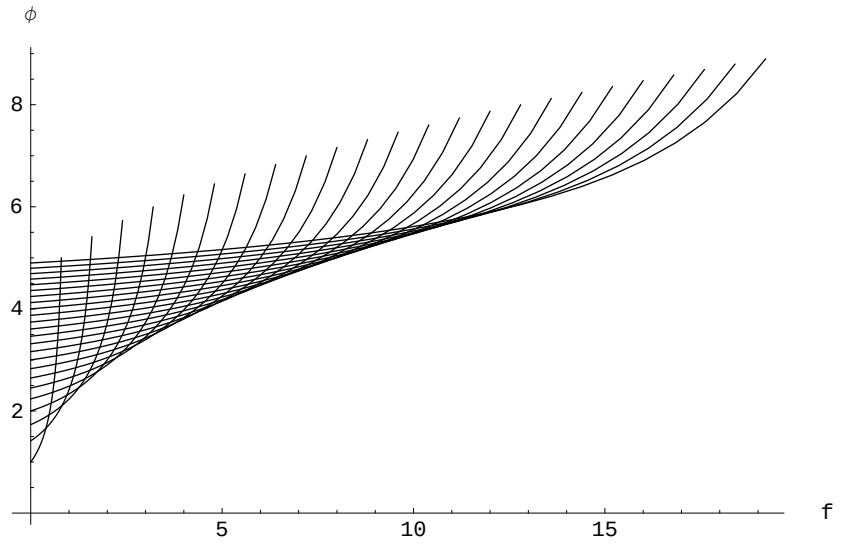


Figura A.1: Função de custos integrada

$$
[CFA4] \begin{cases}
minimizar: & \phi(c,f) = \sum_{i=1}^{n} \{\theta_i(c_i) + \rho T_i(f_i, c_i)\} & \\
sujeito\ a: & T(f,c) \leq T_{max} & \\
 & f_i = \sum_{i=1}^{K} x_i^k & \forall i = 1, ..., m \\
 & f_i \leq c_i, & \forall i = 1, ..., m \\
 & Ax^k = d^k, & \forall k = 1, ..., K \\
 & x \in R^{Km+} & \\
 & f \in R^{m+} & \\
 & c_i \in R^{m+} &
\end{cases}
$$

O custo de instalação de capacidade no arco pode ser um custo qualquer (não-linear) porém côncavo, o que caracteriza economia de escala. A função objetivo obtida ϕ é côncava.

Considerando que a função que descreve o custo de instalação de capacidades obedece a uma função de potência "*power law cost*" $\theta_i(c_i) = \kappa_i(c_i)^{\xi}$ onde $0 \leq \xi \leq 1$ e que a parcela do atraso médio total devido ao arco i é dada por $T_i(f_i, c_i) = \frac{f_i}{c_i - f_i}$ obtem-se:

$$\phi_i(c_i, f_i) = \rho \frac{f_i}{c_i - f_i} + \kappa_i(c_i)^{\xi} \tag{A.1.1}$$

O mínimo de $\phi_i(c_i, f_i)$ em c_i é dado por:

$$\kappa\xi c_i^{\xi-1}(c_i - f_i)^2 - \rho f_i = 0 \tag{A.1.2}$$

A equação A.1.2 é transcendente para valores $0 < \xi < 1$. Uma solução analítica para essa equação não é facilmente obtida. Na Figura A.1 podemos observar como seria esta função. Ela é a casca côncava de todas as funções A.1.1. Entretanto, para cada valor de f_i, $c_i(f_i)$ pode ser determinado numericamente aplicando o método da bisseção, ou o método da falsa posição, ou métodos de ponto fixo. Derivando implicitamente a equação A.1.2 com relação a f_i a derivada de c_i com relação a f_i é obtida:

$$\frac{dc_i}{df_i} = \frac{2\kappa\xi c_i^{\xi} - 2\kappa\xi c_i^{\xi-1} f_i}{2\xi(\xi-1)c_i^{\xi} - 2\kappa\xi^2 c_i^{\xi-1} f_i - \kappa\xi(\xi-1)c^{\xi-1} f_i^2} \tag{A.1.3}$$

Se por um lado, não foi possível obter uma expressão analítica explícita para $c_i(f_i)$ o que permitiria obter uma expressão analítica explícita para $\phi_i(f_i)$, por outro lado $\frac{d\phi_i(f_i)}{df_i}$ pode ser calculado explicitamente com $c_i(f_i)$.

Aplicando A.1.3 a derivada da equação A.1.1 com relação a f_i é dada por:

$$\frac{d\phi_i}{df_i} = \frac{(\rho + 2\kappa\xi c^{\xi-1}(c_i - f_i))}{2\kappa\xi c^{\xi-1}(c_i - f_i) + \kappa(\xi-1) * \xi c^{\xi-2}(c_i - f_i)^2}; \tag{A.1.4}$$

Mesmo não conhecendo uma expressão analítica explícita para $\phi_i(f_i)$ tudo que é necessário saber sobre ela ($\phi_i(f_i)$, $\frac{d\phi_i}{df_i}$, $\frac{d^2\phi_i}{d^2 f_i}$) é facilmente calculado com a precisão desejada. Este conhecimento é o diferencial sobre as demais formulações propostas na literatura e permite a aplicação direta de métodos de otimização.

O problema de roteamento em uma rede com função objetivo côncava é um problema de otimização global [33]. Por causa da concavidade da função objetivo a otimalidade local não implica em otimalidade global [37]. Abordagens clássicas da programação convexa são capazes apenas de encontrar ótimos locais [34]. A maioria dos problemas de otimização não-convexa são problemas NP-difícies. O problema de rede com função objetivo côncava não foge a esta regra e é um problema NP-difícil [32]. A conseqüência disso é que somente pequenas instâncias do problema podem ser resolvidas de forma exata e a maioria dos algoritmos de solução propostos na literatura são algoritmos de otimização local [66],[29],[52].

A.2 Algoritmo de busca local

Métodos de otimização convexa podem ser adotados na obtenção de soluções ótimas locais do problema A.1. Nesse trabalho o método de desvio de fluxos foi adotado. Esse método na realidade se reduz ao método do ponto fixo proposto por Yaged [66] quando a função objetivo é côncava.

Algoritmo de busca local

Passo 1. Determine uma solução inicial viável f^0 para o problema, $it = 0$.

Passo 2. Resolva a equação A.1.2 e determine c_i. Determine $\frac{d\phi_i}{df_i}$.

Passo 3. Determine o roteamento fluxos de caminho mínimo f^{it+1} correspondente à métrica $l^{it} = [\frac{d\phi^{it}}{df^{it}}]_{f^{it}}$.

Passo 4. Se $(\phi^{it} - \phi^{it+1}) < \epsilon$, onde $\epsilon > 0$ é um erro permitido, interrompa: f^{it+1} é um ótimo local. Senão faça $it = it + 1$ e retorne ao Passo 2.

Diversos algoritmos heurísticos são propostos na literatura para resolver o problema de roteamento em redes multiproduto com custo côncavo entre elas podemos ressaltar o método de separação e avaliação proposto por Minoux [52] e uma heurística proposta por Queiroz e Humes em [60]. As heurísticas propostas por estes autores apresentam resultados melhores do que os algoritmos baseados no método de desvio de fluxos apresentado na próxima seção.

A.3 Experimentos numéricos

Um conjunto de experimentos foi conduzido com o objetivo de estudar a formulação e o algoritmo apresentados. Três topologias apresentadas nos capítulos 3 e 4 foram usadas nos testes: N50 - Figura 3.5 , ARPA - Figura 4.7, RING - Figura 4.8. Para comparar a qualidade dos ótimos locais obtidos com a nova formulação o algoritmo FD apresentado no Capítulo 2 (seção 2.2.1) foi implementado para resolver o problema $CFA1$. Os seguintes parâmetros foram adotados: uma demanda de $1Mbit/s$ foi

Tabela A.1: Comparação dos resultados obtidos com o algoritmo FD aplicado no problema $CFA1$ e com a aplicação do método de desvio de fluxos no problema $CFA4$

Rede	φ^* [$]	$\widetilde{\varphi}$ [$]	σ [$]	ϕ^* [$]	$\widetilde{\phi}$ [$]	σ [$]	$\frac{\widetilde{\varphi}-\widetilde{\phi}}{\widetilde{\phi}}100$ %
ARPA	147.52	151.53	2.67	148.19	149.85	1.55	1.1
RING	282.46	291.20	6.86	282.37	290.15	5.18	0.3
N50	705.29	728.43	17.52	649.45	662.24	9.07	9.0

φ^* melhor resultado obtido em 10 execuções com o algoritmo FD
$\widetilde{\varphi}$ média dos resultados obtidos em 10 execuções com o algoritmo FD
σ desvio padrão
ϕ^* melhor resultado obtido em 10 soluções do problema $CFA4$
$\widetilde{\phi}$ média dos resultados obtidos em 10 soluções do problema $CFA4$
σ desvio padrão

assumida entre cada par de nós, o atraso médio máximo permitido é de $T_{max} = 0.5s$, $\theta_i(c_i) = (c_i)^{0.5}$ ou seja, $\kappa = 1$ e $\xi = 0.5$. Os algoritmos foram implementados em liguagem C via GCC 3.0, sendo a plataforma de implementação um computador com processador AMD DURON 950 MHz, 128 Mb de RAM e sistema operacional LINUX.

Na formulação proposta não é possível estabelecer diretamente o atraso médio máximo da rede. Entretanto, este parâmetro pode ser controlado através do custo de congestionamento ρ. Este parâmetro foi calibrado em cada instância resolvida de forma a se obter $T_{max} = 0.5s$.

Como a função objetivo $\phi(f)$ é côncava e possui um conjunto denso de ótimos locais para cada instância testada foram gerados 10 soluções iniciais aleatórias. Cada solução inicial faz com que os métodos convirjam para ótimos locais não necessariamente distintos. A Tabela A.1 apresenta os resultados obtidos aplicando o algoritmo FD no problema $CFA1$ e o método de desvio de fluxos no problema $CFA4$.

Os resultados obtidos indicam que a formulação integrada favorece a obtenção de melhores resultados. Tanto valores médios menores quanto desvios padrão menores foram obtidos com a nova formulação. Uma especulação que pode ser feita é que a nova formulação integrada proposta $CFA4$ possui uma quantidade de ótimos locais menor do que o problema $CFA1$.

Uma extensão natural desse trabalho é implementar outras heurísticas para confirmar a qualidade da nova formulação.

Referências Bibliográficas

[1] Highway capacity manual. Tech. Rep. 87, Highway Research Board, 1965.

[2] Altinkemer, K., and Gavish, B. Heuristics with constant error guarantees for the design of tree networks. *Management Science 34* (1988), 331–341.

[3] Amiri, A. A system for the design of packet-switched communication networks with economics tradeoffs. *Computer Communications 21* (1998), 1670–1680.

[4] Amiri, A., and Pirkul, H. Routing and capacity assignment in backbone communication. *Computers and Operations Research 24*, 3 (1997), 275–287.

[5] Amiri, A., and Pirkul, H. Routing and capacity assignment in backbone communication networks under time varying traffic conditions. *European Journal of Operational Research 117* (1999), 15–29.

[6] Atamtürk, A., and Rajan, D. On splittable and unsplittable flow capacitated network design arc-set polyedra. *Mathematical Programming 92*, 2 (April 2002), 315–334.

[7] Balas, E. Discrete programming by the filter method. *Operations Research 13*, 3 (1966), 915–955.

[8] Beaubrun, R., and Pierre, S. Routing and delay analysis for high-speed networks. *Computers and Eletrical Engineering*, 27 (2001), 37–53.

[9] Bertsekas, D., and Gallager, R. *Data Networks*. Prentice-Hall, 1987.

[10] Bienstock, D., Chopra, S., Günlük, O., and Tsai, C. Y. Minimum cos capacity intallation for multicommodity network flows. *CORE Discussion Paper* (1995).

[11] Bienstock, D., and Günlük, O. Computacional experience with a difficult mixed-interger multicommodity flow problem. *Mathematical Programming*, 68 (1995), 213–238.

[12] Boyer, F. *Conception et routage dans le réseaux de télécommunication; application de la Méthode de Benders Généralisée*. PhD thesis, Université Blaise Pascal - Clermont II, Clermont-Ferrand, França, 1997.

[13] Chan, H., and Yan, T. Combined channel allocation, routing and flow control algorithm for packet-switched networks. Tech. rep., Jet Propulsion Laboratory, 1987.

[14] Courtois, P. J., and Semal, P. An algorithm for the optimization of nonbifurcated flows in computer communication networks. *Performance Evaluation 1* (1981), 139–152.

[15] Doar, M. B. A better model for generating test networks. *Proceedings of Globecom '96* (November 1996).

[16] Dutta, A., and Lim, J.-I. A multiperiod capacity planning model for backbone computer communications networks. *Operations Research 40*, 4 (July 1992), 689–705.

[17] Falk, J. E. Lagrange multipliers and nonconvex programs. *SIAM J. Control 7* (1969), 534–545.

[18] Falk, J. E., and Hoffman, K. L. A sucessive underestimation method for concave minimization problems. *Mathematics of Operations Research 1* (1976), 251–259.

[19] Falk, J. E., and Soland, R. M. An algorithm for separable nonconvex programming problems. *Management Science 15* (1969), 550–569.

[20] Florian, M., Guélat, J., and Spiess, H. An efficient implementation of the PARTAN variant of the linear approximation method for the network equilibrium problem. *NETWORKS 17* (1987), 319–339.

[21] Fratta, M., Gerla, M., and Kleinrock, L. The flow deviation method: An approach to store-and-forward communication network design. *Networks 3* (1973), 97–133.

[22] Gabrel, V., Knippel, A., and Minoux, M. Exact solution of multicommodity network optimization problems with general step cost functions. *Operations Research Letters 25* (January 1999), 15–23.

[23] Garey, M., Graham, R., and D.S.Johnson. Some NP-complete geometric problems. *Proceedings of the 8'th Annual ACM Symposium on Theory of Computing* (1976), 10–22.

[24] Gavish, B., and Altinkemer, K. Backbone network design tools with economic tradeoffs. *ORSA Journal on Computing 2*, 3 (1990), 236–252.

[25] Gavish, B., and Hantler, S. L. An algorithm for optimal route selection in SNA networks. *IEEE Transactions on Communications 31*, 10 (1983), 1154–1161.

[26] Gavish, B., and Neuman, I. System for routing and capacity assignment in computer communication networks. *IEEE Transactions on Communications 37*, 4 (1989), 360–366.

[27] Geoffrion, A. Integer programming by implicit enumeration and Balas'method. *SIAM Review 9*, 2 (April 1967), 178–190.

[28] Geoffrion, A. Generalized Benders decomposition. *Journal of Optimization Theory and Applications 10*, 4 (1972), 237–260.

[29] Gerla, M. *The Design of Store-and-Forward (S/F) Networks for Computer Communications*. PhD thesis, University of California, Los Angeles, EUA, 1973.

[30] Gerla, M., and Kleinrock, L. On the topological design of distributed computer networks. *IEEE Transactions on Communications COM-25* (1977), 48–60.

[31] Ghannadan, S. *Feasibility and global optimality in networks flows*. PhD thesis, Linköping University, Linköping, Sweden, 1995.

[32] Guisewite, G. *Handbook of Global Optimization*, vol. 2 of *Nonconvex optimization and its applications*. R. Horst P.M. Pardalos, 1995, ch. Network Problems, pp. 609–642.

[33] Guisewite, G., and Pardalos, P. Minimum concave-cost network flow problems: applications, complexity and algorithms. *Annals os Operations Research 25* (1990), 75–100.

[34] Hiriart-Urruty, J.-B. *Handbook of Global Optimization*, vol. 2 of *Nonconvex optimization and its applications*. R. Horst P.M. Pardalos, 1995, ch. Conditions for Global Optimality, pp. 1–26.

[35] Horst, R., Pardalos, P., and Thoai, N. *Introduction to Global Optimization*. Kluwer Academic Publisher, 1995.

[36] Horst, R., and P.M. Pardalos, e. *Handbook of Global Optimization*, vol. 2 of *Nonconvex optimization and its applications*. Kluwer Academic, Dordrecht, The Netherlands, 1995.

[37] Horst, R., and Tuy, H. *Global Optimization Deterministic Approaches*, third ed. Springer, 1995.

[38] Jr, C. H. Tópicos de otimização e redes de computadores. Tese (livre-docência), IME - Universidade de São Paulo, 1988.

[39] Jr, C. H. A projection-feasible directions method for the continuos capacity and flow assignment. *Matemática Aplicada e Computacional 11*, 2 (1992).

[40] Kang, C., and Tan, H. Combined channel allocation and routing algorithms in packet switched networks. *Computer Communications 20* (1997), 1175–1190.

[41] Kleinrock, L. *Communication Nets: Stochastic Message Flow and Delay.* McGraw-Hill, New York, 1964.

[42] Kleinrock, L. *Queueing Systems, Vol.II: Computer Applications*, vol. II. Wiley, New York, 1975.

[43] Kleinrock, L. On the modeling and analysis of computer networks. *Proceedings of the IEEE 81*, 8 (august 1993), 1179–1191.

[44] Lasdon, L. *Optimization Theory for Large Systems.* Macmillan, London, 1970.

[45] LeBlanc, L. Design and operation of packet-switched networks with uncertain message requirements. *IEEE Transactions on Communications 38*, 8 (1990), 1223–1227.

[46] LeBlanc, L., and Simmons, R. V. Continuous models for capacity design of large packet-switched telecommunication networks. *ORSA Journal on Computing 1*, 4 (1989), 271–286.

[47] Lin, J. G. Maximal vectors and multiobjective optimization. *Journal of Optimization Theory and Applications 18*, 1 (1976), 41–64.

[48] Luenberger, D. *Linear and Nonlinear Programming*, second ed. Addison-Wesley, 1984.

[49] Luna, H., and Mahey, P. Bounds for global optimization of capacity expansion and flow assignment problems. *Operations Research Letters 26* (2000), 211–216.

[50] Mahey, P., Phong, T., and Luna, H. Separable convexification and DCA techniques for capacity and flow assignment problems. *RAIRO Operations Research 35* (2001), 269–281.

[51] Minoux, M. Optimization et planification des réseaux de telecommunications. *In Optimization Techniques(Goos,G. Hartmanis,J., Cea,J. Eds.),Lecture Notes in Computer Science* (1976), 419–430.

[52] Minoux, M. Network synthesis and optimum network design problems: models, solution methods and applications. *Network 19* (1989), 313–360.

[53] Monteiro, J. A. S., and Gerla, M. Topological reconfiguration of ATM networks. In *Proceedings of INFOCOM 90* (1990).

[54] Monteiro, J. A. S., Gerla, M., and Pazos, R. Topology design and bandwidth allocation in ATM nets. *IEEE Journal on Selected Areas in Communications 7*, 8 (1989), 1253–1262.

[55] Ng, T. M. J., and Hoang, D. B. Joint optimization of capacity and flow assignment in a packet switched communications network. *IEEE Transactions on Communications 35*, 2 (1987), 202–209.

[56] Ouorou, A., Mahey, P., and Vial, J. P. A survey of algorithms for convex multicommodity flow problems. *Management Science 46*, 1 (2000), 126–147.

[57] Pardalos, P., and Rosen, J. Methods for global concave minimization: a bibliografy survey. *SIAM Review 28*, 3 (1986).

[58] Queiroz, M. Otimização global e o problema de designação de fluxos e capacidades. Master's thesis, IME-Universidade de São Paulo, São Paulo, 1997.

[59] Queiroz, M. G., and Humes-JR, C. The projected pairwise multicommodity flow polyhedron. *Applied Mathematics Letters*, 14 (2001), 443–448.

[60] Queiroz, M. G., and Humes-JR, C. A heuristic for the continuous capacity and flow assignment. *European Journal of Operational Research in press* (2002).

[61] Rockafellar, R. T. *Convex Analisys*. Princeton University Press, 1970.

[62] Rosen, J. B., and Ornea, J. C. Solution of nonlinear programming problems by partitioning. *Management Science*, 10 (1963), 160–173.

[63] Souza, M. C. Un modèle continu et non convexe pour l'expansion de capacités d'un réseau de communications. Tech. rep., LIMOS - Universidade Blaise Pascal, Clermont-Ferrand, França, mars 2001. Quinta Jornada Científica da Escola de Doutorado De Ciências para Engenheiro.

[64] Souza, M. C. *Conception et expansion de réseaux de télécommunication*. PhD thesis, Université Blaise Pascal - Clermont II, Clermont-Ferrand, França, 2002.

[65] Tuy, H. Concave programming under linear constraints. *Soviet Mathematics*, 5 (1964), 1437–1440.

[66] Yaged, B. Minimum cost routing for static network. *Networks 1* (1971), 139–172.

[67] Yen, H. H., and Lin, F. Y. S. Near-optimal delay constrained routing in virtual circuit networks. In *Proceedings of IEEE INFOCOM 2001* (2001).

[68] Zangwill, W. Minimum concave cost flows in certain networks. *Management Science 14*, 7 (1968), 429–450.

[69] Zegura, E., Calvert, K., and Bhattacharjee, S. How to model an internetwork. In *Proceedings of IEEE INFOCOM 96* (1996).