UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA
COORDENAÇÃO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

KÁTIA CILENE COSTA FERNANDES

Técnicas de otimização multiobjetivo e otimização estocástica para o roteamento de fluxos em redes

Goiânia
2019

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TESES E DISSERTAÇÕES NA BIBLIOTECA DIGITAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio da Biblioteca Digital de Teses e Dissertações (BDTD/UFG), regulamentada pela Resolução CEPEC nº 832/2007, sem ressarcimento dos direitos autorais, de acordo com a Lei nº 9610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou *download*, a título de divulgação da produção científica brasileira, a partir desta data.

1. Identificação do material bibliográfico: **[ ] Dissertação** **[x] Tese**

2. Identificação da Tese ou Dissertação:

Nome completo do autor: *Kátia Cilene Costa Fernandes*

Título do trabalho: *Técnicas de otimização multiobjetivo e otimização estocástica para o roteamento de fluxos em redes*

3. Informações de acesso ao documento:

Concorda com a liberação total do documento [x] SIM [ ] NÃO[1]

Havendo concordância com a disponibilização eletrônica, torna-se imprescindível o envio do(s) arquivo(s) em formato digital PDF da tese ou dissertação.

Assinatura do(a) autor(a)[2]

Ciente e de acordo:

Assinatura do(a) orientador(a)[2]

Data: 28 / 02 / 2019

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. A extensão deste prazo suscita justificativa junto à coordenação do curso. Os dados do documento não serão disponibilizados durante o período de embargo.
Casos de embargo:
- Solicitação de registro de patente
- Submissão de artigo em revista científica
- Publicação como capítulo de livro
- Publicação da dissertação/tese em livro

[2] A assinatura deve ser escaneada.

KÁTIA CILENE COSTA FERNANDES

Técnicas de otimização multiobjetivo e otimização estocástica para o roteamento de fluxos em redes

Tese apresentada ao Programa de Pós–Graduação do Instituto de Informática da Universidade Federal de Goiás, como requisito parcial para obtenção do título de Doutor em Ciência da Computação.

Área de concentração: Ciência da Computação

Orientador: Prof. Dr. Leizer de Lima Pinto

Co-Orientador: Prof. Dr. Kleber Vieira Cardoso

Goiânia
2019

Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema de Bibliotecas da UFG.

Costa Fernandes, Kátia Cilene
Técnicas de otimização multiobjetivo e otimização estocástica para o roteamento de fluxos em redes [manuscrito] / Kátia Cilene Costa Fernandes. - 2019.
LXXIX, 79 f.: il.

Orientador: Prof. Dr. Leizer de Lima Pinto; co-orientador Dr. Kleber Vieira Cardoso.
Tese (Doutorado) - Universidade Federal de Goiás, Instituto de Informática (INF), Programa de Pós-Graduação em Ciência da Computação em rede (UFG/UFMS), Goiânia, 2019.
Bibliografia.
Inclui gráfico, tabelas, algoritmos, lista de figuras, lista de tabelas.

1. Roteamento de fluxos em redes. 2. Otimização biobjetivo. 3. Algoritmo polinomial. 4. Otimização estocástica. 5. Técnica e constraint. I. de Lima Pinto, Leizer, orient. II. Título.

CDU 004

UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL
UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA
DOUTORADO EM CIÊNCIA DA COMPUTAÇÃO

Ata de Defesa de Tese de Doutorado

Aos vinte e dois dias do mês de março de dois mil e dezenove, no horário das catorze horas, foi realizada, nas dependências do Instituto de Informática da UFG, a defesa pública da Tese de Doutorado da aluna Kátia Cilene Costa Fernandes, matrícula no. 2014100102, intitulada **"Técnicas de otimização multiobjetivo e otimização estocástica para o roteamento de fluxos em redes"**.

A Banca Examinadora, constituída pelos professores:

Prof. Dr. Leizer de Lima Pinto – INF/UFG – orientador

Prof. Dr. Kleber Vieira Cardoso – INF/UFG - coorientador

Prof. Dr. Flávio Henrique Teles Vieira – EEMC/UFG

Prof. PhD. Elivelton Ferreira Bueno – goGeo.io

Prof. Dr. Antônio Jorge Gomes Abelém – UFPA

Prof. Dr. Yuri Abitbol de Menezes Frota – UFF

emitiu o resultado:

(X) Aprovado

( ) Aprovado com revisão

(A Banca Examinadora deve definir as exigências a serem cumpridas pelo aluno na revisão, ficando o orientador responsável pela verificação do cumprimento das mesmas.)

( ) Reprovado

com o seguinte parecer: _____

_____

_____

_____

Prof. Dr. Leizer de Lima Pinto

Kleber Vieira Cardoso

Prof. Dr. Kleber Vieira Cardoso

Flt Vieira

Prof. Dr. Flávio Henrique Teles Vieira

Elivelton F. Bueno

Prof. PhD. Elivelton Ferreira Bueno

Prof. Dr. Antônio Jorge Gomes Abelém

Yuri Abitbol V. Menezes Frota

Prof. Dr. Yuri Abitbol de Menezes Frota

Kátia Cilene Costa Fernandes

Graduação em Licenciatura Plena em Matemática pela Universidade Federal de Uberlândia (1993). Mestrado em Matemática pela Universidade Federal de Goiás (2003). Atualmente é professora efetiva do Instituto Federal de Educação, Ciência e Tecnologia de Goiás - Câmpus Anápolis (IFG).

Dedico este trabalho à minha filha Luíza Costa Cunha. Sem você nenhuma conquista valeria a pena. Ao meu marido, Marcus Antonio Peixoto da Cunha, pelo apoio incondicional.

Agradecimentos

A Deus por me dar saúde, inspiração e muita força para superar todas as dificuldades.

Ao meu orientador, Leizer de Lima Pinto, e ao meu co-orientador, Kleber Vieira Cardoso, por todo o tempo que dedicaram a me orientar durante o processo de realização desta tese.

A este instituto (INF/UFG) e todo seu corpo docente, além da direção e administração que me proporcionaram as condições necessárias para que eu alcançasse meus objetivos.

Ao Instituto Federal de Goiás (IFG), Câmpus Anápolis, por me proporcionar um tempo de afastamento e, aos meus colegas dessa instituição, principalmente os professores da área da Matemática, que facilitaram, muitas vezes, os meus horários de trabalho e até mesmo assumiram obrigações extras para me ajudar.

Aos meus pais, Ilda José Costa Fernandes e João Fernandes da Silva, que dignamente me ensinaram a importância da família e o caminho da honestidade e da persistência.

Enfim, a todos que contribuíram para a realização deste trabalho, seja de forma direta ou indireta, o meu muito obrigada!

A felicidade não se resume na ausência de problemas, mas sim na sua capacidade de lidar com eles.

Albert Einstein.

Resumo

Fernandes, K. C. C.. **Técnicas de otimização multiobjetivo e otimização estocástica para o roteamento de fluxos em redes**. Goiânia, 2019. 79p. Tese de Doutorado. Coordenação de Pós-Graduação em Ciência da Computação, Instituto de Informática, Universidade Federal de Goiás.

Neste trabalho estamos interessados em problemas de otimização ligados ao roteamento de fluxos em redes. Três modelos e um algoritmo exato e polinomial são apresentados. O primeiro modelo é um problema de programação inteira biobjetivo em que as funções objetivo referem-se ao balanceamento de carga da rede e ao comprimento dos caminhos por onde os fluxos são roteados. Um algoritmo exato e polinomial baseado na técnica ε-*constraint* é apresentado. O segundo modelo difere do primeiro no que tange aos pesos dos fluxos e às qualidades das arestas. Nele esses parâmetros podem assumir valores distintos. O último modelo trata-se de um problema estocástico mono-objetivo de roteamento de fluxos. Ele visa minimizar o gargalo da rede, respeitando um certo limite no comprimento dos caminhos por onde os fluxos são roteados. Além disso, as qualidades dos enlaces são variáveis aleatórias, que podem ser aproximadas por um conjunto discreto e finito de cenários. Implementações foram desenvolvidas em linguagem C++ utilizando o solver CPLEX para a resolução das instâncias. Topologias em grade e topologias aleatórias baseadas no modelo Barabási-Albert foram utilizadas em nossos experimentos computacionais. As configurações dos fluxos de rede definidos aqui são aquelas comumente usadas em redes de sensores sem fio e redes de malha sem fio. A análise dos resultados computacionais fornece ao tomador de decisão informações valiosas sobre quais fatores mais afetam as soluções.

Palavras–chave

Roteamento de fluxos em redes, otimização biobjetivo, algoritmo polinomial, otimização estocástica, técnica ε-constraint.

Abstract

Fernandes, K. C. C.. **Multiobjective optimization techniques and stochastic optimization for flow routing in networks**. Goiânia, 2019. 79p. PhD. Thesis. Coordenação de Pós-Graduação em Ciência da Computação, Instituto de Informática, Universidade Federal de Goiás.

In this work we are interested in optimization problems related to network flow routing. Three models and an exact and polynomial algorithm are presented. The first model is a bi-objective integer programming problem in which the objective functions refer to the load balancing of the network and the length of the paths through which the flows are routed. An exact and polynomial algorithm based on the ε-constraint technique is presented. The second model differs from the first one with respect to the weights of the flows and the link qualities. In these parameters can assume different values. The last model is a stochastic single-objective flow routing problem. It aims to minimize the bottleneck of the network, respecting a certain limit on the length of the paths through which flows are routed. In addition, the link qualities are random variables, which can be approximated by a discrete and finite set. Implementations were developed in C++ language using the CPLEX solver for the resolution of instances. Grid topologies and random topologies based on the Barabási-Albert model were used in our computational experiments. The network flow settings defined here are those commonly used in wireless sensor networks and wireless mesh networks. The analysis of computational results provides the decision maker valuable informations about which factors most affect the solutions.

Keywords

Network flow routing, biobjective optimization, polynomial algorithm, stochastic optimization, ε-constraint technique.

Sumário

Lista de Figuras

Lista de Tabelas

CAPÍTULO 1

Introdução

O mundo vem enfrentando cenários de intensos gargalos tanto na rede de transporte, pelo crescimento infrene da frota automobilística e o desenvolvimento mitigado das vias rodoviárias [Neto et al. 2011, Borges 2013], como na rede de comunicação, pelo desenvolvimento significativo dos equipamentos e das ferramentas de tecnologias eletrônicas [Heeks 2010]. Nesse sentido, se faz necessário o estudo e o desenvolvimento de técnicas de otimização tratando tais problemas.

O objetivo geral deste trabalho é estudar e desenvolver técnicas de otimização para a resolução de problemas de roteamento de fluxos em redes. Inicialmente, abordamos alguns cenários específicos de redes sem fio, como determinados *backhauls* de redes de acesso e redes em malha sem fio, formadas predominantemente por enlaces (arestas) com boa qualidade. Tradicionalmente, essas redes possuem alta capilaridade, ou seja, há múltiplas opções de caminho para a maior parte dos fluxos de rede.

No entanto, estratégias de balanceamento de carga são necessárias para fazer uso efetivo da capilaridade da rede, conforme descrito por [Pham e Perreau 2004]. Embora, do ponto de vista do operador de rede, uma solução de objetivo único focada apenas no balanceamento de carga possa ser satisfatória, essa solução é geralmente inadequada para muitos usuários da rede. A razão para isso é que os comprimentos de caminho dos fluxos não são levados em conta e isso pode criar rotas inaceitavelmente longas.

Com intuito de resolver os problemas de roteamento de fluxo em redes com essas características, formulamos um problema biobjetivo de programação inteira que visa minimizar, simultaneamente, o gargalo da rede e o custo total do roteamento dos fluxos. Para resolver esse problema nos baseamos no método ϵ-*constraint*, propondo um algoritmo exato e polinomial para a obtenção de um conjunto mínimo completo de soluções Pareto-ótimas [Pinto e Fernandes 2016, Pinto et al. 2019].

Recentemente, as redes sem fio de múltiplos saltos têm se popularizado, por exemplo, através de redes em malha sem fio usadas em empresas e domicílios, redes de sensores sem fio e redes veiculares usadas em cidades inteligentes. Nesses tipos de redes, o roteamento é essencial para garantir uma utilização eficiente dos recursos da rede, por exemplo, capacidade de processamento, comunicação e energia dos dispositivos. Além

disso, deve existir uma solução de roteamento dos fluxos que distribua o tráfego através de múltiplos caminhos, levando em consideração, as qualidades dos enlaces [Liu et al. 2012, Gálvez e Ruiz 2013]. Essa qualidade afeta a capacidade efetiva do enlace. Assim, o peso do fluxo sobre um enlace é também afetado pela qualidade. Além disso, o tráfego de rede de comunicação de dados é dominado por fluxos TCP (*Transmission Control Protocol*) ou similares, isto é, fluxos que são afetados pelo comprimento do caminho. Quanto mais longo for o caminho de um fluxo TCP, maior será o tempo (médio) de retorno das confirmações (ACKs) e, portanto, menor será sua vazão média. Como consequência, o comprimento do caminho de um fluxo também afeta seu peso sobre um enlace.

Considerando essas especificidades, criamos o segundo modelo biobjetivo de roteamento de fluxos. Nesse modelo há uma função objetivo totalizadora, que minimiza o custo total de todos os fluxos, e uma função gargalo, a qual é composta por duas métricas: a qualidade do enlace e o peso do fluxo de rede.

A incerteza sobre a qualidade de um enlace é uma propriedade comum à maioria das redes sem fio de múltiplos saltos. Embora o uso de alguma medida estatística, como a média, seja comum para representar a qualidade de um enlace (ou aresta) em um modelo de otimização, essa abordagem é uma simplificação que pode ser inapropriada em determinados cenários. Apenas para ilustrar, um enlace sem fio pode passar a maior parte do tempo flutuando entre dois estados ruim e bom, mas ser representado como um enlace regular (o qual corresponde a um estado raro ou inexistente para o referido enlace).

Portanto, é útil ter soluções de roteamento que sejam capazes de levar em conta essas flutuações. Caso contrário, fluxos podem ser encaminhados por enlaces que degradam seu desempenho. Isso tem motivado a preferência por abordagens heurísticas [Biswas e Morris 2005, Meng et al. 2016], dada a dificuldade de capturar a incerteza gerada pelas flutuações na qualidade dos enlaces em modelos de otimização, especialmente os modelos determinísticos clássicos. Por exemplo, a simples recomputação de rotas ótimas (quando a qualidade dos enlaces se altera) pode levar a instabilidade do roteamento [Ramachandran et al. 2007] e degradação do desempenho dos fluxos. Dependendo do tamanho da rede, essa recomputação regular pode se tornar computacionalmente inviável.

Para representar a incerteza na qualidade dos enlaces, utilizamos variáveis aleatórias cuja função de distribuição acumulada é conhecida. Formulamos então um modelo estocástico mono-objetivo para o roteamento de fluxos que visa minimizar o gargalo da rede. Como restrição, controlamos o comprimento total dos caminhos dos fluxos. No entanto, essa restrição e a função objetivo são acopladas através de um sobrepeso, cuja influência depende de sua probabilidade de ocorrer num determinado cenário. Assim, é possível exceder essa restrição se a penalidade total for inferior ao benefício obtido na redução do gargalo.

No próximo capítulo, apresentamos uma revisão bibliográfica associada a este trabalho. No Capítulo 3, abordamos o problema biobjetivo de roteamento de fluxos em redes, minimizando o gargalo da rede e o custo total de todos os fluxos. No Capítulo 4, tratamos uma variante do problema abordado no capítulo anterior, em que levamos em consideração os pesos dos fluxos e as qualidades dos enlaces na função gargalo. No Capítulo 5, apresentamos o modelo estocástico mono-objetivo citado no parágrafo anterior. No Capítulo 6, apresentamos nossas considerações finais.

CAPÍTULO 2

Trabalhos Relacionados

O objetivo deste capítulo é apresentar investigações de problemas multiobjetivo da área de otimização combinatória multiobjetivo, acompanhados de técnicas de geração de soluções para esses tipos de problemas. Outra área que abordamos nesse estudo é a otimização estocástica. Investigamos os tipos de modelos de problemas estocásticos mais comuns na literatura, seguidos de métodos de geração de cenários. Além disso, apresentamos métodos de avaliação da estabilidade dos resultados das soluções desses problemas.

2.1 Otimização combinatória multiobjetivo

A otimização combinatória e inteira consiste em problemas que maximizam ou minimizam uma função objetivo envolvendo várias variáveis de decisão sujeito a dois tipos de restrições, as restrições de igualdades/desigualdades e as restrições de integralidade sobre algumas ou todas as variáveis [Schrijver 2000]. Quando esses problemas envolvem duas ou mais funções objetivo, eles são denominados de problemas de otimização combinatória multiobjetivo (OCMO). Dentre as classes dessa área, destacam-se os problemas de programação inteira multiobjetivo (PIMO) e os problemas de fluxos em rede multiobjetivo [Clímaco, Ferreira e Captivo 1997, Ehrgott e Gandibleux 2000, Ehrgott e Gandibleux 2003, Ulungu e Teghem 1994].

Os problemas de fluxos em redes são uma classe de problemas em que os dados de entrada é uma rede de fluxo (um grafo com capacidades numéricas sobre suas arestas) e o objetivo é construir um fluxo, valores numéricos sobre cada aresta que respeite as restrições de capacidade e que tenham a quantidade de fluxos de entrada igual a quantidade de fluxos de saída em todos os vértices, exceto para determinados terminais designados [Ahuja, Magnanti e Orlin 1993]. Os problemas de fluxos em redes com múltiplos objetivos possuem uma extensa variedade de problemas de interesses práticos como, por exemplo, em redes de comunicação e em sistemas de transporte. Muitos desses problemas podem ser modelados e tratados como problemas de PIMO [Ehrgott e Gandibleux 2003, Ulungu e Teghem 1994].

Nos problemas multiobjetivo não existe um ótimo no sentido habitual mono-objetivo e sim um conjunto de soluções eficientes ou não-dominadas. Como os objetivos são conflitantes, então não existe uma solução que otimize, simultaneamente, todas as funções objetivo [Clímaco, Antunes e Alves 2003]. Essas soluções são classificadas como soluções *fracamente eficientes* e soluções *Pareto-ótimas* (*estritamente eficientes*) (veja [Chankong e Haimes 2008]). Como afirma [Martins 1984], nenhum objetivo em uma solução dominada pode ser melhorado sem piorar pelo menos uma das outras funções objetivo.

A literatura traz vários métodos para determinar todas as soluções eficientes, ou um subconjunto pré-definido dessas soluções, para os problemas de PIMO, denominados de métodos geradores. A técnica mais utilizada para achar soluções eficientes é a escalarização, empregada em quase todos os métodos exatos e em muitas técnicas heurísticas, que transforma o problema de PIMO em problemas com um único objetivo, os quais são resolvidos repetidamente [Alves e Costa 2012, Cohon 2013, Chankong e Haimes 2008, Ehrgott 2006].

Dentre esses métodos de escalarização para resolver problemas multiobjetivos, destacam-se o método das restrições (ϵ-*constraint*), apresentado inicialmente por [Marglin 1967]. No entanto, apareceu, pela primeira vez, com o nome "ϵ-*constraint*" em [Haimes e Wismer 1971] e, posteriormente, em [Haimes 1973]. Ainda em 1973, foi utilizado em problemas aplicados como de [Cohon e Marks 1973] (problema de programação linear multiobjetivo) e [Miller e Byers 1973]. Após seis anos, [Cohon, Church e Sheer 1979] exibiram o algoritmo do método das restrições de maneira generalizada para problemas multiobjetivo.

O método das restrições consiste em criar um problema restrito (subproblema mono-objetivo) onde a função objetivo é uma das p funções objetivo e as $(p-1)$ funções restante do problema multiobjetivo se transformam em restrições desse subproblema, resolvendo-o, repetidamente. Para os problemas multiobjetivo de programação linear, esse método é classificado como um método aproximado. Mas, para os problemas de PIMO e de caminho ele se torna um método muito utilizado em algoritmos exato.

Considere um problema multiobjetivo com p funções objetivo, $z(x) = [z_1(x), \cdots, z_p(x)]$. Para resolver esse problema, utilizando o método das restrições, cria-se subproblemas mono-objetivo que são definidos da seguinte forma:

$$\begin{aligned} &\min\ z_k(x) \\ &\text{sujeito a: } z_j(x) \leq \epsilon_j,\ j = 1, \cdots, p,\ j \neq k \\ &\qquad\qquad x \in X, \end{aligned}$$

onde $\epsilon_j \in \mathbb{R}$, para $j \in \{1, \cdots, k-1, k+1, \cdots, p\}$.

[Chankong e Haimes 2008] trazem resultados gerais sobre esse método.

Esses resultados para problemas multiobjetivo são: (1) Se x é uma solução ótima do subproblema mono-objetivo então x é uma solução fracamente eficiente do problema multiobjetivo (isto é, não existe nenhuma solução $\bar{x} \in X$ tal que $z(\bar{x}) < z(x)$). (2) Se a solução ótima, x, do subproblema mono-objetivo é única, então ela é uma solução *Pareto-ótima* para o problema multiobjetivo (isto é, não existe nenhuma solução $\bar{x} \in X$ tal que $z(\bar{x}) \leq z(x)$ e $z(\bar{x}) \neq z(x)$). (3) x é uma solução eficiente do problema multiobjetivo se, e somente se, x é uma solução ótima do subproblema, para todo $j = 1, ..., p$, onde $\epsilon_k = z_k(x)$, $k \neq j$.

Outra técnica de escalarização desenvolvida é um algoritmo para achar todos pontos não dominados de um problema de programação inteira biobjetivo genérico apresentada em [Chalmet, Lemonidis e Elzinga 1986]. O algoritmo consiste em resolver uma sequência de problemas de programação inteira com um único objetivo. Esse problema tem como função objetivo a soma ponderada das duas funções objetivo. Além disso, uma função objetivo torna-se restrição. Podemos citar também o trabalho de [Lokman e Köksalan 2013] que traz algoritmos exatos para achar todos os pontos não dominados de problemas de PIMO. O primeiro algoritmo melhora o algoritmo de [Sylva e Crema 2004], por reduzir o número de variáveis binárias e restrições. O segundo algoritmo emprega um procedimento de busca e resolve um número de modelos para o próximo ponto evitando quaisquer variáveis binárias adicionais.

Citamos, também, o trabalho de [Özlen e Azizoğlu 2009] que apresenta um algoritmo recursivo, baseado no método ϵ-*constraint*, para gerar todos os pontos não dominados para problemas de PIMO. Nesse trabalho não são apresentados experimentos computacionais. Em 2014, [Özlen, Burton e MacRae 2014] trazem uma melhoria desse algoritmo, apresentando, também, resultados computacionais.

Resultados de obras com técnicas de escalarização, para resolver problemas de PIMO, podemos citar trabalhos como de [Klein e Hannan 1982], que apresentam uma técnica determinando algumas ou todas as soluções eficientes. O algoritmo consiste em resolver uma sequência de problemas de programação linear inteira mono-objetivo, progressivamente, baseado no método ϵ-*constraint*. As restrições adicionadas em cada etapa são as mesmas daquelas adicionadas nas etapas anteriores, diferenciando apenas no seu lado direito.

[Ehrgott 2006] traz uma análise das técnicas de escalarização para resolver problemas de PIMO. Além disso, apresenta uma nova técnica de escalarização, com o uso de restrições sobre os valores objetivo e acréscimo de penalidade na função objetivo, denominado de método das restrições elásticas (*method of elastic constraints*). Nesse método, há combinação de características do método das restrições e do método da soma ponderada que tem como objetivo determinar todas as soluções eficientes do PIMO.

Nas abordagens do método ϵ-*constraint* para problemas biobjetivo em grafos,

em cada iteração, uma ou mais arestas são desconsideradas. Ou seja, trabalham com subgrafos, por exemplo [Martins 1984]. Em uma abordagem similar de [Gadegaard, Klose e Nielsen 2016], por exemplo, os custos para passar por essas arestas assumem valores suficientemente grandes, com base no gargalo da iteração anterior.

Quando trata-se de problemas multiobjetivo de caminho, as técnicas mais utilizadas nos algoritmos exatos são: rotulagem, classificação e duas fases. Para mais detalhamento veja [Clímaco e Pascoal 2012].

Na década de 80, [Hansen 1980] apresenta algoritmos de rotulagem, para diversos problemas de caminho biobjetivo, que trabalham diretamente sobre o grafo. Dentre os problemas, composto por uma função objetivo totalizadora e outra gargalo. As funções totalizadora e gargalo são funções que avaliam as soluções através da soma dos pesos de seus elementos (*MinSum*) e pelo pior peso de seus elementos (*MinMax* ou *MaxMin*), respectivamente. Posteriormente, [Clímaco e Martins 1982] apresentam um problema genérico com duas funções objetivo totalizadoras e um método para o problema, que classifica os caminhos mínimos, visando determinar os caminhos não-dominados. Dois anos depois, [Martins 1984] manifesta o interesse em problemas de caminho biobjetivo em que uma das funções é do tipo gargalo. Um algoritmo para determinar um conjunto mínimo completo de caminhos não-dominados é apresentado.

Problemas biobjetivo generalizados em grafos, que são modelados com duas medidas de desempenhos, sendo uma função totalizadora e outra gargalo, foi apresentado por [Berman, Einav e Handler 1990]. O artigo traz três algoritmos com o intuito de achar todas as soluções Pareto-ótimas, com base no método das restrições.

Para o problema de caminho tri-objetivo, com uma função custo e duas funções gargalo, [Pinto, Bornstein e Maculan 2009] trazem o primeiro algoritmo exato e polinomial para obtenção de um conjunto mínimo completo de soluções Pareto-ótimas. O método consiste em determinar caminhos mínimos em subgrafos obtidos por um conjunto restrito de arestas, de acordo com limites para as funções gargalo. No ano seguinte, [Pinto e Pascoal 2010] desenvolvem um algoritmo com melhor desempenho computacional, para o mesmo problema. Posteriormente, [Bornstein et al. 2012] generalizam e estendem o algoritmo anterior para problemas de otimização combinatória multiobjetivo, com uma função totalizadora e n funções gargalo.

Em diversos problemas em redes, é comum ter como restrição no roteamento de fluxos que cada fluxo seja roteado por um caminho (por exemplo, [Laporte e Osman 1995]). Um objetivo clássico nos problemas de roteamento de fluxos em redes é o balanceamento de carga, que consiste em rotear os fluxos minimizando o gargalo da rede. Outro objetivo comum consiste em minimizar o comprimento dos caminhos para o roteamento dos fluxos. Nesses problemas, os pesos associados aos elementos (enlaces) não são dados de entrada. Eles consistem dos fluxos que serão atribuídos aos enlaces.

Aplicações para esses problemas podem ser encontradas em [Gálvez e Ruiz 2013] ou [Mello et al. 2016]. Esses trabalhos apresentam algoritmos heurísticos para variantes desse problema.

2.2 Otimização estocástica

Outra área da otimização que vem despertando grande interesse no meio científico são os problemas de otimização cuja abordagem inclui a incerteza nos seus dados e, consequentemente, nas suas variáveis de decisão. Assim, alguns dados do problema podem ser representados como variáveis aleatórias. Isso se dá pelo fato de que existe uma grande quantidade de problemas, do mundo real, que vêm acompanhados da incerteza [Birge e Louveaux 2011, King e Wallace 2012].

Dentre esses problemas de incerteza, destacamos uma classe denominada de problemas de otimização combinatória estocástica (POCE). Neles, parte da informação sobre os dados do problema é desconhecida e o conhecimento sobre sua distribuição de probabilidade é assumido. Assim, numa aplicação real, normalmente, temos que definir/avaliar distribuições de probabilidade a partir de dados reais ou hipotéticos, não sendo uma tarefa muito fácil. E, além disso, a função objetivo exige mais computacionalmente do que em problemas de otimização combinatória determinístico.

Os problemas POCE têm uma vasta importância em problemas que tratam de situações como roteamento de veículos, onde há a incerteza, por exemplo, nas demandas de usuários e/ou no tempo de viagem. E são presentes, também, em problemas de roteamento em rede de informação, em que a incerteza se deve, por exemplo, à qualidade das conexões, à variabilidade do tráfego e aos pacotes de informações.

Uma investigação sobre problemas de roteamento de veículos estocástico é apresentado em [Gendreau, Laporte e Séguin 1996], destacando as contribuições mais importantes. Dentre elas podemos citar [Laporte, Louveaux e Mercure 1989] que trazem um modelo de penalidade limitada, acompanhado de algoritmos exato de relaxação restrita para resolvê-lo, utilizando várias distribuições para a demanda. Temos, também, [Bertsimas 1992] que propõem um algoritmo heurístico para resolver problemas de roteamento de veículos onde a demanda é incerta, utilizando técnicas de análise probabilísticas.

Em 2009, [Bianchi et al. 2009] apresentam meta-heurísticas para abordagens algorítmicas clássicas de otimização estocástica, acompanhadas de um tutorial das meta-heurísticas que, atualmente, é aplicado à otimização sob incerteza.

Dentre os POCEs, destacamos os problemas de programação inteira multiobjetivo estocástico. Em [Abbas e Bellahcene 2006] é proposto um algoritmo para resolver problemas de programação linear inteira multiobjetivo estocástico, que

combina a técnica de plano de corte, desenvolvida por [Abbas e Moulai 1999], e o método de decomposição *L-shaped* [Kall, Wallace e Kall 1994].

Em problemas de roteamento de fluxos em redes, podemos citar trabalhos aplicados a situações reais como de [Abdel-Rahman et al. 2016] que consideram o problema da alocação de recursos, em que as demandas dos usuários são incertas.

Na programação estocástica, os problemas de otimização vêm acompanhados de parâmetros que são variáveis aleatórias discretas ou contínuas, juntamente com uma distribuição probabilística. Na literatura, destacam-se os modelos de recurso, *chance-constraint*, dois estágios e o multiestágios [Sahinidis 2004].

Nos modelos de programação estocástica de dois estágios, as variáveis de decisão são divididas em dois estágios. As variáveis de primeiro estágio são aquelas que têm de ser decididas antes da realização real dos parâmetros incertos. Daí, uma vez que os eventos aleatórios se apresentarem, podem ser feitas outras melhorias de projeto ou de política operacional selecionando, a um certo custo, os valores das variáveis de segundo estágio ou de recurso [Sahinidis 2004, Shapiro e Philpott 2007].

Um problema de programação estocástica de dois estágios pode ser formulado da seguinte forma:

$$\begin{aligned} &\text{Minimizar} && h(x) + \mathbb{E}[f(x, \tilde{\xi})] \\ &\text{sujeito a:} && x \in X \subseteq \mathbb{R}^n \end{aligned}$$

onde $\tilde{\xi}$ é uma variável aleatória definida em um espaço de probabilidade $(\tilde{\Omega}, \tilde{\mathbb{A}}, \tilde{\mathbb{P}})$ (com $\tilde{\Omega}, \tilde{\mathbb{A}}$ e $\tilde{\mathbb{P}}$, respectivamente, denotando o conjunto de todos os resultados, uma coleção de variáveis aleatórias e as probabilidades atribuídas), e onde para qualquer realização dada ξ de $\tilde{\xi}$, temos:

$$\begin{aligned} & f(x, \xi) = \min\ g(\xi)^T y \\ &\text{sujeito a:}\ W(\xi)y \geq r(\xi) - T(\xi)x \\ & \qquad\qquad y \in Y \subseteq \mathbb{R}^m. \end{aligned}$$

Onde $W(\xi)y \geq r(\xi) - T(\xi)x$ é uma restrição envolvendo a variável aleatória, mas podendo ter um sobrepeso, $T(\xi)x$ para o cenário . Aqui, x e y, respectivamente, denotam as variáveis de primeiro estágio e as variáveis de segundo estágio. Os conjuntos associados, X e Y, são assumidos ser descritos por restrições lineares juntamente com algumas possíveis restrições de integralidade.

Os modelos de recurso utilizam ações de correções para diminuir a violação de restrições que podem surgir depois da realização das incertezas. Esse modelo foi proposto por [Dantzig 2004] e [Beale 1955] para problemas de dois estágios e, consequentemente, estende para os problemas de multiestágios.

Nos modelos *chance-constraint* o foco está na confiabilidade do sistema, isto é, na capacidade do sistema de atender a viabilidade em um ambiente incerto. Essa confiabilidade é expressa como um requisito mínimo sobre a probabilidade de satisfazer

restrições do problema [Sahinidis 2004].

Os modelos de multiestágios é a extensão dos modelos de programação estocástica de dois estágios, mas incluem complicações adicionais que permitem decisões revisadas em cada estágio de tempo com base na incerteza percebida até o momento. As informações de incerteza em uma programação estocástica de multiestágios são modeladas como uma árvore de cenários de camadas múltiplas [Birge e Louveaux 2011].

Para trabalhar com problemas de otimização estocástica faz-se necessário conhecer técnicas de geração de árvores de cenários. As árvores de cenários são estruturas de dados básicas para problemas de otimização estocástica de dois estágios e multiestágios. Essas árvores representam as discretizações dos processos estocásticos e, portanto, uma aproximação dos fenômenos reais [Pflug e Pichler 2015].

Em [Kaut e Wallace 2003] podem ser encontradas comparações de diferentes técnicas de geração de árvore de cenários. Dentre essas técnicas destaca-se o método *moment-matching* [Høyland, Kaut e Wallace 2003] que para utilizá-lo devemos conhecer, além da matriz correlação das variáveis aleatórias, os momentos (média, variância, assimetria e curtose) do processo estocástico. Outra técnica muita utilizada é baseada em simulação de Monte Carlo, denominada *aproximação por média amostral* (*SAA - sample average approximation*) primeiramente proposto por [Shapiro e Mello 1998].

Com intuito de reduzir o erro da simulação utilizando essas técnicas de geração de árvore de cenários, devemos aumentar o tamanho da árvore de cenários, até que atinja a precisão desejada. Porém, nem sempre é viável computacionalmente.

Para uma análise da qualidade e do tamanho dessas árvores, utilizamos testes denominados de *testes de estabilidade*. Há três tipos de testes que devem ser testados, teste *in-sample*, teste *out-of-sample* e *bias*.

Quando os valores ótimos da função objetivo, $f(\hat{x}_i;\tau_i)$ são (aproximadamente) os mesmos em todas árvores de cenários, τ_i, ou seja: $f(\hat{x}_i;\tau_i) \approx f(\hat{x}_j;\tau_j)$ com $i \neq j$, onde $\hat{x}_i$ e $\hat{x}_j$ são as soluções ótimas do problema para as árvores de cenários τ_i e τ_j, respectivamente, dizemos que a estabilidade é *in-sample*. Significa que estamos preocupados com a estabilidade do valor da função reportado pelo próprio problema.

Estabilidade *out-of-sample* ocorre quando os valores verdadeiros das funções objetivo,$f(\hat{x}_i;\xi)$, correspondente às soluções provenientes de diferentes árvores de cenários, são (aproximadamente) o mesmo valor. Ou seja, essa estabilidade é definida da seguinte forma: $f(\hat{x}_i;\xi) \approx f(\hat{x}_j;\xi)$, onde ξ é a árvore de referência.

Além disso, mesmo que ocorra estabilidade de *in-sample* e out-of-sample, ainda pode ocorrer problemas com a estabilidade. Então, esse problema é denominado de um *Bias* (viés). Para essa análise, deve verificar se $f(\hat{x}_i;\xi) \approx f(\hat{x}_j;\tau_i)$. Esse teste pode ser feito somente se podemos resolver o verdadeiro problema ($\min_x f(x;\xi)$) [King e Wallace 2012].

CAPÍTULO 3

Um problema de programação inteira biobjetivo de roteamento de fluxos

Os algoritmos exatos para resolver problemas de roteamento de fluxos em redes, com funções totalizadora e gargalo, como mencionamos no Capítulo 2, são limitados a resolver problemas em que os pesos, nos elementos que compõem as soluções, são dados de entrada. Ou seja, cada aresta da rede (i,j) recebe um valor b_{ij} associado à função gargalo. Então, o valor de gargalo de uma solução x é o maior b_{ij} dos links que compõem x.

Neste capítulo, um modelo de programação inteira biobjetivo de roteamento de fluxos com balanceamento de carga e de comprimento de caminhos é apresentado. Assim, os pesos associados aos elementos (enlaces) deixam de ser dados de entrada, pois consistem da quantidade de fluxos que serão atribuídos aos enlaces.

O algoritmo proposto para resolver esse modelo é de forma exata e se baseia no método das restrições. No final do algoritmo é gerado um conjunto mínimo completo de soluções Pareto-ótimas para o problema [Pinto e Fernandes 2016, Pinto et al. 2019].

3.1 O Problema (P)

Considere um conjunto de fluxos F, com $|F| = r$, a ser roteado em um grafo orientado $G = (V, E)$, onde V representa o conjunto de nós e E o conjunto de arestas, com $|V| = n$ e $|E| = m$. Cada fluxo $f \in F$ deve ser roteado por um único caminho em G, indo da sua origem $s^f \in V$ ao seu destino $d^f \in V$. Para cada aresta $(i,j) \in E$, indo de $i \in V$ para $j \in V$, e fluxo $f \in F$, definimos uma variável binária x_{ij}^f, que irá indicar se o fluxo f vai passar (ou não) pela aresta (i,j). Considere, ainda, c_{ij}^f o custo, não negativo, para o fluxo f utilizar a aresta (i,j).

Assim, o modelo de programação inteira biobjetivo, que formulamos para o problema, de rotear os r fluxos, minimizando o gargalo da rede e o custo total, é apresentado a seguir:

$$(P)\quad \textit{minimizar}\left\{\max_{(i,j)\in E}\left\{\sum_{f\in F} x_{ij}^{f}\right\}\right\} \tag{3-1}$$

$$\textit{minimizar}\left\{\sum_{f\in F}\sum_{(i,j)\in E} c_{ij}^{f}.x_{ij}^{f}\right\} \tag{3-2}$$

sujeito a:

$$\sum_{(i,j)\in E} x_{ij}^{f} - \sum_{(j,i)\in E} x_{ji}^{f} = \begin{cases} 1, & \text{se } i = s^{f} \\ 0, & \forall i \in V - \left\{d^{f}, s^{f}\right\} \\ -1, & \text{se } i = d^{f} \end{cases} \qquad \forall f \in F \tag{3-3}$$

$$x_{ij}^{f} \in \{0,1\},\ \forall (i,j) \in E \text{ e } \forall f \in F \tag{3-4}$$

Enquanto (3-1) busca minimizar a carga da(s) aresta(s) mais congestionada(s), (3-2) busca minimizar o custo total do roteamento dos fluxos. As restrições (3-3) garantem que cada fluxo $f \in F$ saia da sua origem, s^f, e chegue ao seu destino, d^f, passando por um único caminho.

Podemos observar que se $c_{ij}^{f} = 1$, $\forall\,(i,j) \in E$ e $\forall f \in F$, o objetivo de minimizar o custo total do roteamento se reduz ao objetivo de minimizar o total de saltos percorridos por todos os fluxos, podendo ser utilizado em muitas aplicações reais (por exemplo, [Mello et al. 2016]).

Temos que (3-1) é denominada de uma função gargalo e (3-2) uma função totalizadora. Essas funções objetivo são conflitantes, pois minimizar a carga da(s) aresta(s) mais congestionada(s) deve implicar custos maiores (caminhos mais longos) para roteamento dos fluxos. Da mesma forma, minimizar o custo total de todos os fluxos deve levar a um maior valor para a função gargalo.

Antes de apresentarmos o algoritmo para resolver o problema (P) vamos trazer algumas definições necessárias para a continuidade deste capítulo.

Vamos adotar a notação para induzir ordenação sobre $\mathbb{R}^2$ como segue. Sejam $z^1 = (z_1^1, z_2^1) \in \mathbb{R}^2$ e $z^2 = (z_1^2, z_2^2) \in \mathbb{R}^2$, então: (i) $z^1 \leq z^2 \Leftrightarrow z_k^1 \leq z_k^2,\ \forall k = 1,2$ e $z^1 \neq z^2$; (ii) $z^1 < z^2 \Leftrightarrow z_k^1 < z_k^2,\ \forall k = 1;2$.

Considere X, o *conjunto de soluções viáveis* de (P), e $z(x) = [z_1(x), z_2(x)]$, o *vetor objetivo* associado a $x \in X$, onde:

$$z_1(x) = \max_{(i,j)\in E}\left\{\sum_{f\in F} x_{ij}^{f}\right\} \quad \text{e} \quad z_2(x) = \sum_{f\in F}\sum_{(i,j)\in E} c_{ij}^{f}.x_{ij}^{f}.$$

Definição 3.1 Para $x, x' \in X$, dizemos que x *domina* x' quando $z(x) \leq z(x')$.

Definição 3.2 Uma solução viável $x' \in X$ é uma *solução fracamente eficiente* se não existe nenhuma $x \in X$ tal que $z(x) < z(x')$.

Definição 3.3 Uma solução viável $x \in X$ é uma *solução Pareto-ótima* (estritamente eficiente) se não existe nenhuma outra solução em X que domine x.

Definição 3.4 O conjunto $X^* \subseteq X$, formado por soluções Pareto-ótimas, é um *conjunto mínimo completo* quando: 1) para cada par de soluções $x^*, \bar{x} \in X^*$ temos $z(x^*) \neq z(\bar{x})$; e, 2) para qualquer solução Pareto-ótima $x \in X$ existe $x^* \in X^*$ tal que $z(x) = z(x^*)$.

Por definição, o conjunto das soluções fracamente eficientes inclui as soluções estritamente eficientes. Entretanto, como em [Clímaco, Antunes e Alves 2003], por razões práticas, quando falarmos de soluções fracamente eficientes, neste trabalho, não vamos considerar as estritamente eficientes.

Aplicação 1 Suponha uma via rodoviária representada por um grafo onde, cada trecho (aresta) ligando duas cidades (nós), exista um ponto de parada (fiscalização obrigatória ou pedágio). A função z_2 pode representar o total de paradas dos caminhões ($c_{ij}^f = 1, \forall (i,j) \in E$ e $\forall f \in F$), de uma determinada companhia de transporte, que trafegam por esta via. A função z_1 visa medir a densidade do tráfico no trecho mais congestionado. A companhia pode estar interessada em, simultaneamente, minimizar a quantidade total de paradas dos seus caminhões e balancear a carga na via, minimizando o gargalo.

3.2 O Algoritmo

Nesta seção, apresentamos um algoritmo baseado no método das restrições (ϵ-*constraint*) com o intuito de acharmos um conjunto mínimo completo de soluções Pareto-ótimas para o problema (P). Em cada iteração, um subproblema mono-objetivo é definido e resolvido, o qual é um problema de programação inteira 0-1 que denominamos por (P_ϵ).

A função objetivo de (P_ϵ) é dada por z_2, isto é, a função totalizadora (3-2) do problema biobjetivo (P). As restrições de (P_ϵ) são (3-3) e (3-4) de (P), juntamente com uma restrição sobre o gargalo, a qual é definida através do parâmetro inteiro e positivo, que denotamos por ϵ. Esse parâmetro é usado para controlar os valores da função gargalo e para garantir a otimalidade do algoritmo:

$$\max_{(i,j) \in E} \left\{ \sum_{f \in F} x_{ij}^f \right\} \leq \epsilon. \tag{3-5}$$

A restrição (3-5) não é linear. Assim o subproblema não é um problema de programação linear inteira (PLI) e, naturalmente, ele não pode ser resolvido por PLI. Mas, podemos substituir essa restrição, baseado no Lema 1, afim de transformar esse subproblema em um PLI.

Lema 1 *As restrições* $\sum_{f \in F} x_{ij}^f \leq \epsilon$, $\forall (i,j) \in E$, *são equivalentes à restrição* $\max_{(i,j) \in E} \left\{ \sum_{f \in F} x_{ij}^f \right\} \leq \epsilon$.

Logo, depois da substituição da restrição (3-5) pelas restrições $\sum_{f \in F} x_{ij}^f \leq \epsilon$, $\forall (i,j) \in E$, temos o seguinte subproblema mono-objetivo PLI:

$$(P_\epsilon) \quad \textit{minimizar} \left\{ \sum_{f \in F} \sum_{(i,j) \in E} c_{ij}^f . x_{ij}^f \right\}$$

sujeito a:

$$\sum_{(i,j) \in E} x_{ij}^f - \sum_{(j,i) \in E} x_{ji}^f = \begin{cases} 1, & \text{se } i = s^f \\ 0, & \forall i \in V - \{d^f, s^f\} \\ -1, & \text{se } i = d^f \end{cases} \qquad \forall f \in F \tag{3-6}$$

$$\sum_{f \in F} x_{ij}^f \leq \epsilon, \ \forall (i,j) \in E \tag{3-7}$$

$$x_{ij}^f \in \{0,1\}, \ \forall (i,j) \in E \text{ e } \forall f \in F$$

A escolha de manter a função totalizadora (3-2) como função objetivo, e tratar a função gargalo (3-1) como restrição, se deve ao fato de (3-2) ser uma função linear e (3-1) não ser. Porém, as restrições associadas ao parâmetro ϵ, que limitam os valores de (3-1), são lineares. Logo, esta estratégia nos permite resolver o problema (P) por programação linear.

Inicialmente fazemos $\epsilon = r$ (quantidade de fluxos a ser roteado) e, após a obtenção da solução ótima $\bar{x}$ para (P_ϵ), fazemos $\epsilon = z_1(\bar{x}) - 1$. Dessa forma, o gargalo da nova solução, $\bar{x}$, obtida na iteração seguinte, será menor do que o da anterior, renomeada para $\hat{x}$. Então, caso o valor de z_2 seja o mesmo para essas duas soluções, temos que a solução anterior é dominada pela nova solução, daí a solução anterior é deletada. Assim, segue o algoritmo até uma iteração em que (P_ϵ) seja inviável. O algoritmo completo é apresentado a seguir.

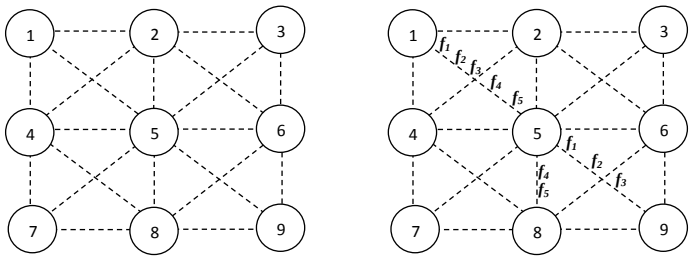
Algoritmo 1

1. $X^* \leftarrow \emptyset$; $\epsilon \leftarrow r$.
2. ***Enquanto*** (P_ϵ) *não for inviável* ***faça***
3. $\quad \bar{x} \leftarrow$ *solução ótima de* (P_ϵ)
4. $\quad X^* \leftarrow X^* \cup \{\bar{x}\}$
5. $\quad \epsilon \leftarrow z_1(\bar{x}) - 1$
6. $\quad$ ***Se*** $|X^*| > 1$ ***e*** $z_2(\bar{x}) = z_2(\hat{x})$ ***então***
7. $\quad\quad X^* \leftarrow X^* - \{\hat{x}\}$
8. $\quad \hat{x} \leftarrow \bar{x}$

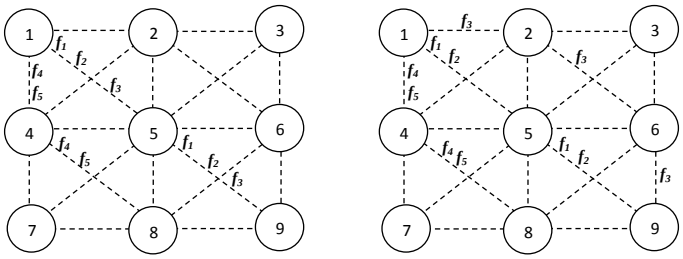
O decremento de exatamente uma unidade no valor de ϵ, de uma iteração para a próxima, é necessário para garantir a obtenção de um conjunto mínimo completo de soluções Pareto-ótimas. Como os valores de z_1 são inteiros, a atualização de ϵ garante que qualquer solução, com z_1 menor do que o da última solução obtida, será viável para o subproblema atual. Porém, embora ϵ sofra esse pequeno decréscimo, a solução obtida pode ter um valor para z_1 menor do que este limite. Por exemplo, se em uma dada iteração temos $\epsilon = 15$ e obtemos uma solução com $z_1 = 10$, na iteração seguinte o limite passa a ser $\epsilon = 9$.

Exemplo de funcionamento 3.1 Considere $G = (V, E)$ o grafo grade 3-por-3 (Figura 3.1(a)) e $F = \{f_1, f_2, f_3, f_4, f_5\}$ o conjunto de fluxos. Suponhamos que os fluxos f_1, f_2 e f_3

tenham a mesma origem e o mesmo destino, nos vértices 1 e 9, respectivamente, e os fluxos f_4,e f_5 tenham a mesma origem e o mesmo destino, nos vértices 1 e 8, respectivamente. E, além disso, considere $c_{ij}^f = 1,\ \forall (i,j) \in E,\ \forall f \in F$. Aqui, queremos minimizar o total de saltos de todos os fluxos e o número de fluxos da aresta mais congestionada.



(a) *Grade 3-por-3* (b) *Primeira iteração*



(c) *Segunda iteração* (d) *Terceira iteração*

Figura 3.1: Execução do Algoritmo 1 para o exemplo 3.1.

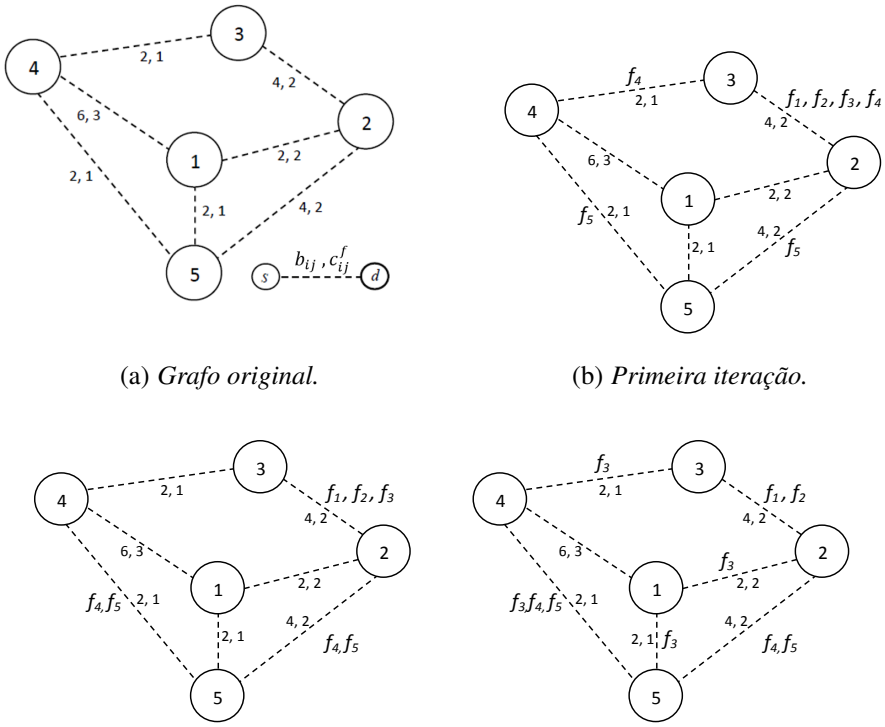
A seguir a Tabela 3.1 apresenta as soluções ótimas de P_ϵ geradas pelo Algoritmo 1 e,no final do algoritmo, gera um conjunto mínimo completo de soluções Pareto-ótima X^* de (P).

Tabela 3.1: Soluções geradas pelo Algoritmo 1 do Exemplo 3.1

Iteração (k)	ϵ	(P_ϵ)	$z_1(x^k)$	$z_2(x^k)$	X^*
1	5	Viável	5	10	$\{x^1\}$
2	4	Viável	3	10	$\{x^2\}$
3	2	Viável	2	11	$\{x^2, x^3\}$
4	1	Inviável	-	-	$\{x^2, x^3\}$

Na primeira iteração, faça $\epsilon = r = 5$. Resolve (P_ϵ) e tem-se a solução ótima $\bar{x}^1$ e $z_1(x^1) = 5$ (veja Figura 3.1(b)). Agora, atualiza ϵ, para $\epsilon = z_1(x^1) - 1 = 5 - 1 = 4$, e resolve (P_ϵ), tem-se a a solução ótima x^2 e $z_1(x^2) = 3$ (veja Figura 3.1(c)). Fazendo o teste de dominância da solução atual com a solução anterior, a partir da segunda iteração, tem-se que, na iteração 2, $z_2(x^2) = z_2(x^1) = 10$ então x^2 domina x^1, daí remove a solução x^1 de X^*. Na iteração 3, tem-se que $z_2(x^3) > z_2(x^2)$, logo x^3 não domina x^2. Na quarta iteração P_ϵ é inviável. Portanto, um conjunto mínimo completo das soluções Pareto-ótimas é $X^* = \{x^2, x^3\}$, como mostra a Tabela 3.1.

Exemplo de funcionamento 3.2 Considere o conjunto de fluxos $F = \{f_1, f_2, f_3, f_4, f_5\}$ a ser roteado no grafo da Figura 3.2(a), onde o nó 2 é a origem de todos os fluxos, o nó 3 é o destino dos fluxos f_1, f_2 e f_3, e o nó 4 é o destino de f_4 e f_5. Neste exemplo, vamos considerar que na rede já tenha uma quantidade de fluxos, b_{ij}, nas arestas $(i,j) \in E$, assim, teremos $z_1(\bar{x}) = \max_{(i,j)\in E}\{b_{ij} + \sum_{f\in F} x_{ij}^f\}$. Além disso, existe um custo c_{ij}^f para cada fluxo $f \in F$ atravessar uma certa aresta $(i,j) \in E$.



(a) *Grafo original.* (b) *Primeira iteração.*

(c) *Segunda iteração.* (d) *Terceira iteração.*

Figura 3.2: Execução do Algoritmo 1 para o exemplo 3.2.

Na primeira iteração tem-se $\epsilon = r + \max_{(i,j)\in E} b_{ij} = 5 + 6 = 11$ e obtemos a solução ótima x^1, com $z_1(x^1) = 8$ (oito fluxos na aresta $(2,3)$) e $z_2(x^1) = 12$ (custo total para o roteamento dos fluxos), conforme Figura 3.2(b). Na segunda iteração, resolvendo (P_ϵ) para $\epsilon = 7$, obtém-se a solução ótima x^2, com $z_1(x^2) = 7$ e $z_2(x^2) = 12$, conforme Figura 3.2(c). Como ocorre empate no valor de z_2, ou seja, x^2 domina x^1, então é eliminada a solução x^1 do conjunto de soluções X^*. Na terceira iteração, para $\epsilon = 6$, obtém-se x^3, com $z_1(x^3) = 6$ e $z_2(x^3) = 15$, conforme Figura 3.2(d). Na iteração seguinte o algoritmo é finalizado, após detectar que (P_ϵ), com $\epsilon = 5$, é inviável. A Tabela 3.2 resume a execução do algoritmo.

Tabela 3.2: Soluções geradas pelo Algoritmo 1 do Exemplo 3.2.

Iteração (k)	ϵ	(P_ϵ)	$z_1(x^k)$	$z_2(x^k)$	X^*
1	11	Viável	8	12	$\{x^1\}$
2	7	Viável	7	12	$\{x^2\}$
3	6	Viável	6	15	$\{x^2, x^3\}$
4	5	Inviável	-	-	$\{x^2, x^3\}$

A seguir vamos mostrar que a matriz tecnológica (matriz dos coeficientes), formada pelos coeficientes das restrições dos problemas mono-objetivo (P_ϵ), é totalmente unimodular. Para isso, antes vamos apresentar propriedades que dão condições necessárias e/ou suficiente para uma matriz ser totalmente unimodular (TU). Uma matriz diz-se *Totalmente Unimodular* se qualquer submatriz quadrada dessa matriz tiver determinante igual a -1, 0 ou $+1$. Além disso, apresentamos algumas propriedades envolvendo matriz totalmente unimodular e problemas de programação linear inteira com matriz tecnológica TU. As demonstrações desses resultados (propriedades) podem ser encontradas em [Tavares 2005, Ferreira e Wakabayashi 1996, Hoffmann e Kruskal 1956, Maurras, Truemper e Akguel 1981, Schrijver 1998].

Propriedade 3.1 *As propriedades que decorrem imediatamente da definição de Unimodularidade Total são as seguintes:*

a. *Toda a submatriz de uma matriz Totalmente Unimodular é Totalmente Unimodular. Em particular, todos os elementos de uma matriz Totalmente Unimodular são 0, 1 ou -1.*
b. *A transposta de uma matriz Totalmente Unimodular também é Totalmente Unimodular.*
c. *Se A é Totalmente Unimodular e P é uma matriz de permutação de dimensão adequada então a matriz AP, ou PA, é Totalmente Unimodular.*
d. *Se multiplicarmos uma linha ou coluna de uma matriz Totalmente Unimodular por -1, a matriz obtida também é Totalmente Unimodular.*
e. *Se duplicarmos uma linha (ou uma coluna) de uma matriz Totalmente Unimodular, a matriz obtida ainda é Totalmente Unimodular.*
f. *Se acrescentarmos uma linha ou uma coluna com no máximo uma entrada não nula a uma matriz Totalmente Unimodular, a matriz obtida ainda é Totalmente Unimodular.*

Propriedade 3.2 *Se C e D são matrizes TU então:*

a. $\begin{pmatrix} C & O \\ O & D \end{pmatrix}$ *é TU, onde O representa matrizes nulas.*
b. $\begin{pmatrix} C \\ I \end{pmatrix}$ *é TU, onde I é uma matriz identidade.*

Propriedade 3.3 *Uma matriz A de números inteiros é Totalmente Unimodular se, e somente se, para todo o vector b inteiro, $P = \{x \in \mathbb{R}^n : Ax \leq b, x \geq 0\}$ é um poliedro inteiro.*

Sejam $A_1, A_2, \cdots, A_r$, matrizes dos coeficientes das restrições (3-6) para os fluxos $f_1, f_2, \cdots, f_r \in F$, respectivamente. E, as matrizes dos coeficientes das restrições (3-7) para cada fluxo, são matrizes identidades I_m. Seja A a matriz tecnológica de (P_ϵ), depois de inserido as m variáveis de folga nas restrições (3-7) associadas para ϵ, ou seja, suponha que $Ax = b$ represente a forma matricial das restrições do subproblema (P_ϵ), temos que os elementos de A são iguais a ± 1 ou zero.

Daí, podemos escrever a matriz A em blocos, da seguinte forma:

$$A = \begin{pmatrix} A_1 & O & \cdots & O & O \\ O & A_2 & \cdots & O & O \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ O & O & \cdots & A_{r-1} & O \\ O & O & \cdots & O & A_r \\ I_m & I_m & \cdots & I_m & I_m \end{pmatrix}$$

Observe que A_i, $\forall\, i = 1, \cdots, r$, são matrizes totalmente unimodular, pois representam os coeficientes das restrições de problema de rede de fluxo de cada fluxo $f \in F$. E, I_m, também, são TU, pois são matrizes identidades de ordem m.

Temos que a matriz transposta A' de A é dada por:

$$A' = \begin{pmatrix} A'_1 & O & \cdots & O & O & I_m \\ O & A'_2 & \cdots & O & O & I_m \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ O & O & \cdots & A'_{r-1} & O & I_m \\ O & O & \cdots & O & A'_r & I_m \\ O & O & \cdots & O & O & I_m \end{pmatrix}$$

Pela Propriedade (3.1-b.) temos que A'_i, $\forall\, i = 1, \cdots, r$, são matrizes TU.

Aplicando a Propriedade (3.2-a.), repetidamente, temos que a submatriz, formada pelas r primeiras colunas de blocos de A', é TU, que denotaremos de $\bar{A}$, e $\bar{I}$ a submatriz formada por todas as r primeiras matrizes identidades da última coluna de blocos de A'. Observe que $\bar{I}$ é TU: basta aplicarmos a propriedade (3.2-b.), repetidamente. Com isso podemos escrever A' da seguinte forma:

$$A' = \begin{pmatrix} \bar{A} & \bar{I} \\ O & I_m \end{pmatrix}$$

Observe que se aplicarmos apenas operações elementares (soma e subtrações) sobre as linhas obteremos a seguinte matriz equivalente de A':

$$B = \begin{pmatrix} \bar{A} & O \\ O & I_m \end{pmatrix}$$

Ou seja, $A' \sim B$ e, além disso, B é TU pela propriedade (i-b). Logo, $A'x = b$ tem somente soluções inteiras então, pela propriedade (3.3), temos que A' é TU.

Assim, se A' é TU, pela propriedade (i-a) temos que $(A')'$ é TU. Mas, $(A')' = A$. Portanto, A é TU.

Sabemos que se um problema de programação linear inteira com um único objetivo, ou seja, problemas do tipo $\max\{cx : Ax \leq b, x \geq 0\}$ ou $\min\{yb : yA \geq c, y \geq 0\}$, em que a matriz tecnológica é uma matriz de inteiros totalmente unimodular e b, c são vectores inteiros então o problema pode ser resolvido em tempo polinomial (veja [Maurras, Truemper e Akguel 1981]). Como a matriz tenológica A de (P_ϵ) é TU e o vetor do lado direito de (P_ϵ) tem somente valores inteiros então, (P_ϵ) pode ser resolvido em tempo polinomial.

Sabemos que se x é uma solução ótima de (P_ϵ), para algum ϵ, com $1 \leq \epsilon \leq r$, esta solução x poderá ser uma solução fracamente eficiente para (P) (veja [Chankong e Haimes 2008]). Mas, observe que não há solução alguma $x \in X$ que domina qualquer solução $\bar{x} \in X^*$, no final do algoritmo. Ou seja, o conjunto X^*, no final do algoritmo, não tem soluções dominadas e nem soluções fracamente eficientes, pois se for gerada alguma solução fracamente eficiente esta é removida na próxima iteração do Algoritmo 1. De fato, veja a Proposição 3.1 que se segue.

Proposição 3.1 *No final do Algoritmo 1 todas as soluções em X^* são Pareto-ótimas para (P).*

Prova. Seja x uma solução Pareto-ótima qualquer de (P) e x^k uma solução qualquer em X^*, no final do algoritmo, obtida resolvendo-se o problema (P_ϵ). Mostraremos, a seguir, que x^k não é dominado por x, o que implica que x^k é Pareto-ótima de (P). Para o caso em que $z_1(x) > z_1(x^k)$ temos, obviamente, que x^k não é dominado por x. No caso de $z_1(x) \leq z_1(x^k)$, temos que x é viável para (P_ϵ). Como a solução ótima em relação a z_2, obtida em (P_ϵ), foi x^k, temos $z_2(x^k) \leq z_2(x)$. Logo, x^k somente seria dominado por x se $z_2(x) = z_2(x^k)$ para $z_1(x) < z_1(x^k)$. Mas, $z_1(x) < z_1(x^k)$ implica em x ser viável para algum $(P_{\bar{\epsilon}})$, $\bar{\epsilon} < \epsilon$. Neste caso a comparação feita na iteração seguinte à que x^k foi obtida nos levaria a excluir x^k, pois $z_2(x) = z_2(x^k)$, o que seria um absurdo, uma vez que $x^k \in X^*$, no final do algoritmo. Observe que se x^k foi obtido na última iteração do algoritmo, então z_1 atinge seu valor mínimo em x^k; consequentemente, $z_1(x^k) \leq z_1(x)$. No caso de $z_1(x^k) = z_1(x)$ teríamos x viável em (P_ϵ) o que levaria $z_2(x^k) \leq z_2(x)$. Portanto, x^k não é dominado por x. □

Proposição 3.2 *Para qualquer solução x, Pareto-ótima para (P), existe uma solução equivalente em X^* no final do Algoritmo 1.*

Prova. Seja ϵ o menor valor inteiro maior ou igual a $z_1(x)$ para o qual uma iteração do algoritmo foi executada. Isto implica que x é viável para (P_ϵ). Seja x^k a solução ótima de (P_ϵ) obtida nesta iteração. Consequentemente, temos que $z_2(x^k) \leq z_2(x)$. Por outro lado, como x é solução Pareto-ótima, então $z_1(x^k) \geq z_1(x)$. Veja que o caso $z_1(x^k) > z_1(x)$ é impossível, pois com isso teríamos executado uma outra iteração $\bar{\epsilon} < \epsilon$ tal que $\bar{\epsilon} \geq z_1(x)$. Logo, $z_1(x^k) = z_1(x)$ e $z_2(x^k) = z_2(x)$. □

Portanto, o conjunto X^*, no final do Algoritmo 1, é um conjunto mínimo completo de soluções Pareto-ótimas para (P). Todas as soluções $x^k \in X^*$ são Pareto-ótimas (Teorema 3.1) e para qualquer solução Pareto-ótima x existe uma solução equivalente em X^*, isto é, com o mesmo vetor objetivo (Teorema 3.2). Além disso, pelo Passo 6 do Algoritmo 1, todas as soluções $x \in X^*$ têm vetores objetivo distintos.

Proposição 3.3 *A complexidade do Algoritmo 1 é $O(r\,g(\epsilon))$, onde r é o número de fluxos e $g(\epsilon)$ é a complexidade de (P_ϵ).*

Prova. O pior caso, para o gargalo, é quando todos os fluxos atravessam uma mesma aresta, isto é, o gargalo é igual a r. Na segunda iteração, o pior caso, seria que o gargalo fosse igual a $r-1$. Seguindo desta forma, a última iteração, no melhor caso, é

que o gargalo seja igual a 1, assim teríamos, no máximo, r iterações. Portanto, como a complexidade de cada iteração é $g(\epsilon)$, temos que a complexidade do Algoritmo 1 é $O(r\,g(\epsilon))$. □

Consequência 3.1 *O Algoritmo 1 é polinomial.*

Prova. De fato, como (P_ϵ) pode ser resolvido polinomialmente e pela Proposição 3.3, a complexidade do *Algoritmo* 1 é $O(r\,g(\epsilon))$ então esse algoritmo é polinomial. □

Consequência 3.2 *A cardinalidade de X^*, no final do Algoritmo 1, é limitada superiormente por r.*

Prova. Pela Proposição 3.3 , o número máximo de iterações é r. Além disso, pelo Algoritmo 1, no máximo uma solução é inserida em X^*, por iteração. Logo, a cardinalidade de X^* é limitado superiormente pelo o número de fluxos, ou seja, $|X^*| \leq r$. □

3.3 Resultados computacionais

As implementações do algoritmo apresentado neste capítulo são feitas em linguagem C++ e compiladas com o Microsoft Visual C++ 2010. O modelo matemático é implementado em C++ utilizando a biblioteca Concert do CPLEX 12.4 e os experimentos são executados em um PC com processador Intel Core 2 Duo T6400 e memória RAM de 4GB.

Os resultados computacionais, aqui apresentados, são referentes à implementação do Algoritmo 1 para resolver do problema (P). Nesse caso o problema (P) busca rotear os r fluxos, minimizando o gargalo da rede e o custo total de todos os fluxos, definidos sobre o grafo grade e o grafo aleatório, utilizando o modelo Barabási-Albert (BA) [Barabási e Albert 1999].

A metodologia que usamos para gerar os custos c_{ij}^f é similar à de [Alvelos e Carvalho 2003]. Nós geramos tanto instâncias com valores dos custos das arestas sendo o mesmo para todos os fluxos, quanto instâncias com diferentes valores. Primeiramente, consideramos custos iguais, $c_{ij}^f = 1$, para todas arestas $(i,j) \in E$ e para todos fluxos $f \in F$, e, posteriormente, os custos c_{ij}^f, escolhidos aleatoriamente dentre os elementos do conjunto $\{1,2,3,4,5\}$.

Quanto aos nós de origem e de destino de cada fluxo $f \in F$, eles são gerados aleatoriamente considerando duas configurações de distribuição dos fluxos: (i) origens quaisquer (múltiplas origens) e destinos quaisquer (múltiplos destinos), (ii) origens quaisquer e único destino. Em cada configuração, avaliamos dois cenários, um com 80 fluxos e outro com 300 fluxos, empregando 100 instâncias para cada um.

3.3.1 Análise dos resultados para os custos das arestas todos iguais

Os custos assumem todos valores iguais, ou seja, $c_{ij}^{f} = 1$ para todas arestas $(i,j) \in E$ e para todos os fluxos $f \in F$.

3.3.1.1 Análise dos resultados para grafo grade

Nessa primeira análise consideramos o grafo grade, a-por-b, definido da seguinte forma: seja V o produto cartesiano $\{1,2,\cdots,a\} \times \{1,2,\cdots,b\}$, ou seja, o conjunto de todos os pares ordenados (u,v) com $u \in \{1,2,\cdots,a\}$ e $v \in \{1,2,\cdots,b\}$. Digamos que dois elementos (u,v) e (u',v') de V são adjacentes se $u = u'$ e $|v - v'| = 1$ ou se $v = v'$ e $|u - u'| = 1$ ou se $u' = u+1$ e $v' = v+1$ ou se $u' = u+1$ e $v' = v-1$, onde $|u - u'|$ representa o módulo da diferença entre os dois números inteiros u e u'. Essa relação de adjacência define um grafo sobre o conjunto V de vértices, como mostra a Figura 3.3. Aqui consideramos $a = b = 10$.

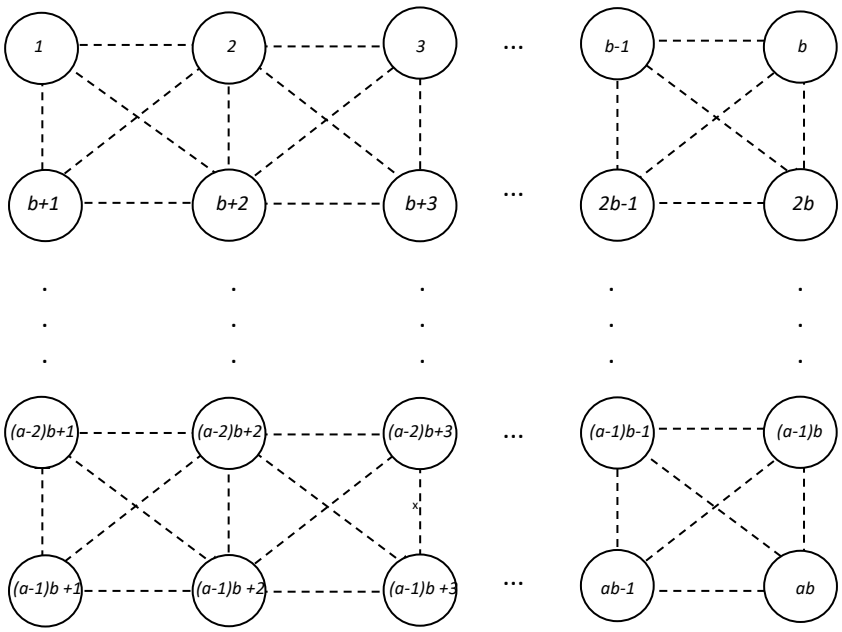


Figura 3.3: Grade a-por-b.

Dentre as métricas analisadas temos o que denominamos de *amplitude total*. Ela representa a diferença entre o valor do custo total de todos os fluxos da última solução Pareto-ótima ($z_2^f(\bar{x})$) e do valor do custo total de todos os fluxos da primeira solução Pareto-ótima ($z_2^o(\bar{x})$), geradas pelo Algoritmo 1, onde $\bar{x} \in X^*$.

A Tabela 3.3 mostra o número de fluxos (r), a cardinalidade do conjunto mínimo completo ($|X^*|$), o número de iterações que P_ϵ foi resolvido (It), as coordenadas do vetor solução da primeira solução Pareto-ótima gerada (z_1^o e z_2^o), as coordenadas do vetor solução da última solução Pareto-ótima gerada (z_1^f e z_2^f) e o tempo (t) de execução do algoritmo, em segundos. Os resultados apresentados são médias e desvios padrões para 100 instâncias.

Para a primeira configuração de distribuição dos fluxos, observamos que a amplitude total dos valores de $z_2(\bar{x})$, onde $\bar{x} \in X^*$, em cada instância, não ultrapassa 23 saltos, para $r = 80$, e 46 saltos, para $r = 300$ (Figuras 3.4(a) e 3.4(b)). Cada Fluxo, em

Tabela 3.3: Média e desvio padrão dos resultados sobre 100 instâncias, considerando $c^f_{ij} = 1$.

Topologia Grade								
Origens e destinos quaisquer								
r	Valores	$\|X^*\|$	It	z^o_1	z^o_2	z^f_1	z^f_2	t
80	Média	2	5	2	375	1	382	79,13
	Desvio Padrão	0,5	0,9	0,2	18,3	0,5	18,6	54,66
300	Média	1	8	4	1,408	4	1.410,0	1.243,31
	Desvio Padrão	0,4	1,5	0,1	37,3	0,4	37,9	1.112,62
Origens quaisquer e mesmo destino								
r	Valores	$\|X^*\|$	It	z^o_1	z^o_2	z^f_1	z^f_2	t
80	Média	6	20	17	379	13	400	158,78
	Desvio Padrão	2,1	9,2	3,4	65,0	3,8	66,8	73,68
300	Média	16	65	62	1.386	47	1.455	1.996,60
	Desvio Padrão	6,6	29,3	13,3	233,4	13,5	241,3	1.001,50

média, faz 6 saltos por iteração, com desvio padrão de 7 saltos, para $r = 80$. Para $r = 300$, uma média de 3 saltos, com desvio padrão de 7 saltos. A média é pequena devido ao fato do Algoritmo 1 ter gerado, em muitas instâncias, $|X^*| = 1$ (Figura 3.9(a)).

Para o total de saltos de cada fluxo $f \in F$, observamos que para $r = 80$, no mínimo $81,25\%$ dos fluxos mantêm o total de saltos inalterados, comparando a primeira solução Pareto ótima com a última solução Pareto-ótima gerada pelo o Algoritmo 1, e para $r = 300$, no mínimo $89,3\%$ dos fluxos. Ou seja, na última solução Pareto-ótima gerada, a maioria dos fluxos, ainda, utiliza o caminho de comprimento mínimo para chegarem aos seus destinos.

Enquanto que grande parte dos demais fluxos aumentam a quantidade de saltos para cada fluxo, na última solução Pareto-ótima gerada, em até $66,7\%$, para $r = 80$, e em até 75% para $r = 300$, comparado com o total de saltos de cada fluxo da primeira solução Pareto-ótima gerada. Raramente ocorre de que um determinado comprimento de caminho de um fluxo dobre de tamanho ($11,25\%$ das instâncias para $r = 80$ e 1% das instâncias para r=300). Isso acontece em, no máximo, dois fluxos por instância (Figuras 3.5(a) e 3.5(b)).

Quanto ao gargalo ($z_1(\bar{x})$) obtivemos baixos valores, como mostra os gráficos das Figuras 3.7(a) e 3.7(b). Esse fato ocorre porque nesse tipo configuração de distribuição de fluxos (origens e destinos múltiplos) os fluxos ficam mais espalhados na rede. Consequentemente, poucas iterações são necessárias (Figura 3.6(a)) e, assim, com pempo médio de 79,129 segundos, com desvio padrão de 54,657 segundos para $r = 80$ (Figura 3.8(a)). Mas, para $r = 300$ o tempo, em média é 1.243,313 segundos, com desvio padrão de 1.112,62 segundos, devido ao aumento expressivo na quantidade de variáveis de decisão ($m.r$) do problema (P_ϵ) (Figura 3.8(b)).

Assim, como é de se esperar, tendo poucas iterações, são geradas poucas soluções Pareto-ótimas em cada instância, o que, na prática, poderá facilitar a tomada de decisão (veja gráfico da função distribuição cumulativa (cdf) - Figura 3.9(a)).

Para a segunda configuração de distribuição dos fluxos (múltiplas origens e um único destino) verificamos que, em pouquíssimas instâncias, é gerado $|X^*| = 1$ (5% das instâncias, para os dois valores de r considerados) e nas demais instâncias, $|X^*|$ atinge valores superiores a dois, independente do valor de r, como mostra a Figura 3.9(b).

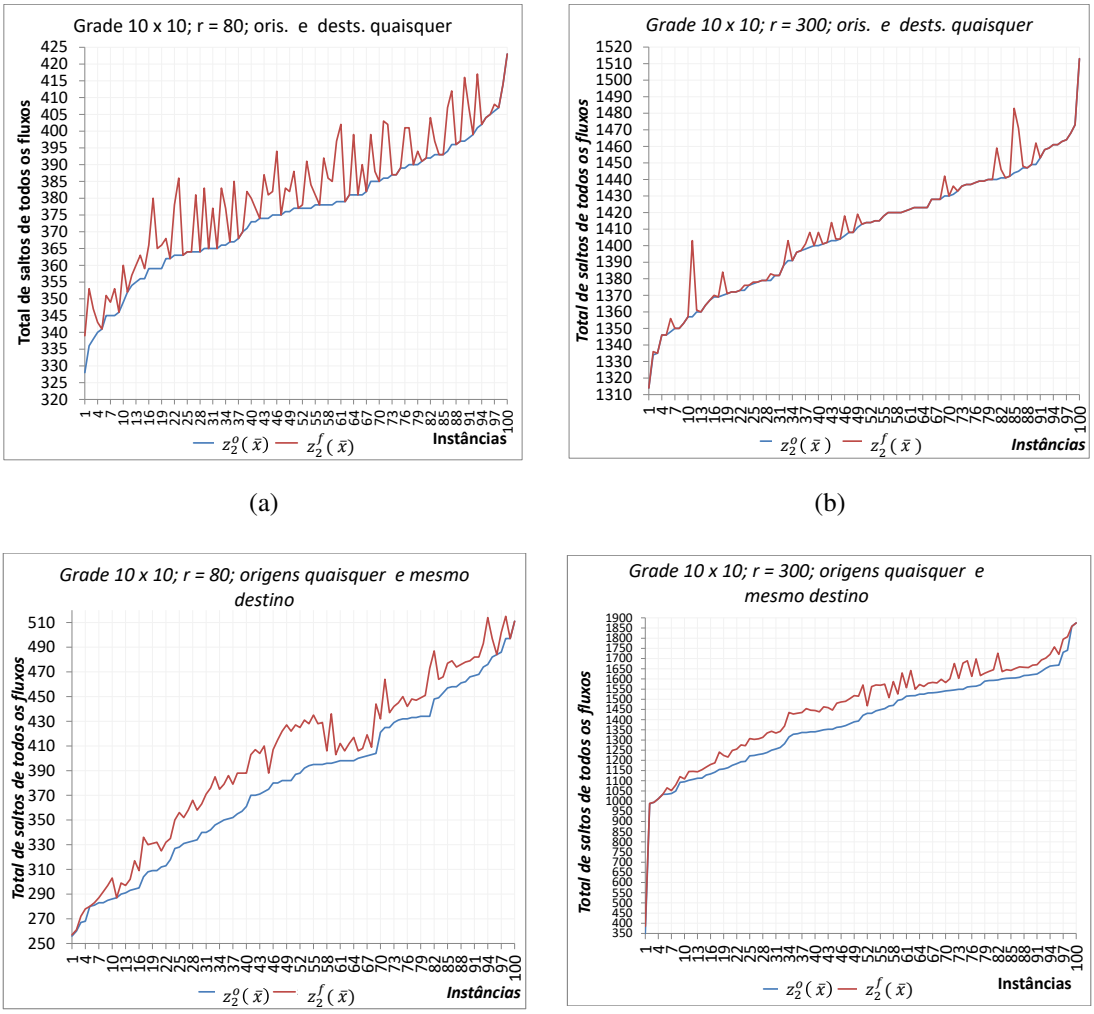


Figura 3.4: Total de saltos de todos os fluxos da primeira solução ($z_2^o(\bar{x})$) e da última solução ($z_2^f(\bar{x})$) Pareto-ótimas gerada pelo Algoritmo 1, em cada instância.

A amplitude total dentre os valores de $z_2(\bar{x})$, com $\bar{x} \in X^*$ de cada instância varia de zero a 45 saltos, para $r = 80$ e de zero a 149 saltos para $r = 300$, como podemos ver nos gráficos das Figuras 3.4(c) e 3.4(d).

Em cada fluxo, observamos que o comprimento de caminho para $r = 80$, no mínimo $61,25\%$ dos fluxos, em cada instância, mantêm o total de saltos inalterados, comparando a primeira solução Pareto ótima com a última solução Pareto-ótima, gerada pelo o Algoritmo 1. E, quando consideramos $r = 300$, no mínimo $63,3\%$ dos fluxos. No restante dos fluxos, para $r = 80$, em 37% das instâncias, no máximo três fluxos por instância dobram o tamanho do comprimento do caminho na última solução Pareto-ótima em relação à primeira solução Pareto-ótima, e em 8% das instâncias apenas um fluxo por instância triplicou o tamanho do caminho e, para as demais instâncias, o acréscimo não ultrapassa $67,7\%$ em relação ao tamanho do caminho mínimo de cada fluxo. Para $r = 300$, nos demais fluxos, em 52% das instâncias, no máximo seis fluxos por instância dobraram o tamanho do comprimento do caminho na última solução Pareto-ótima gerada em relação à primeira solução Pareto-ótima e em 13% das instâncias, no máximo dois

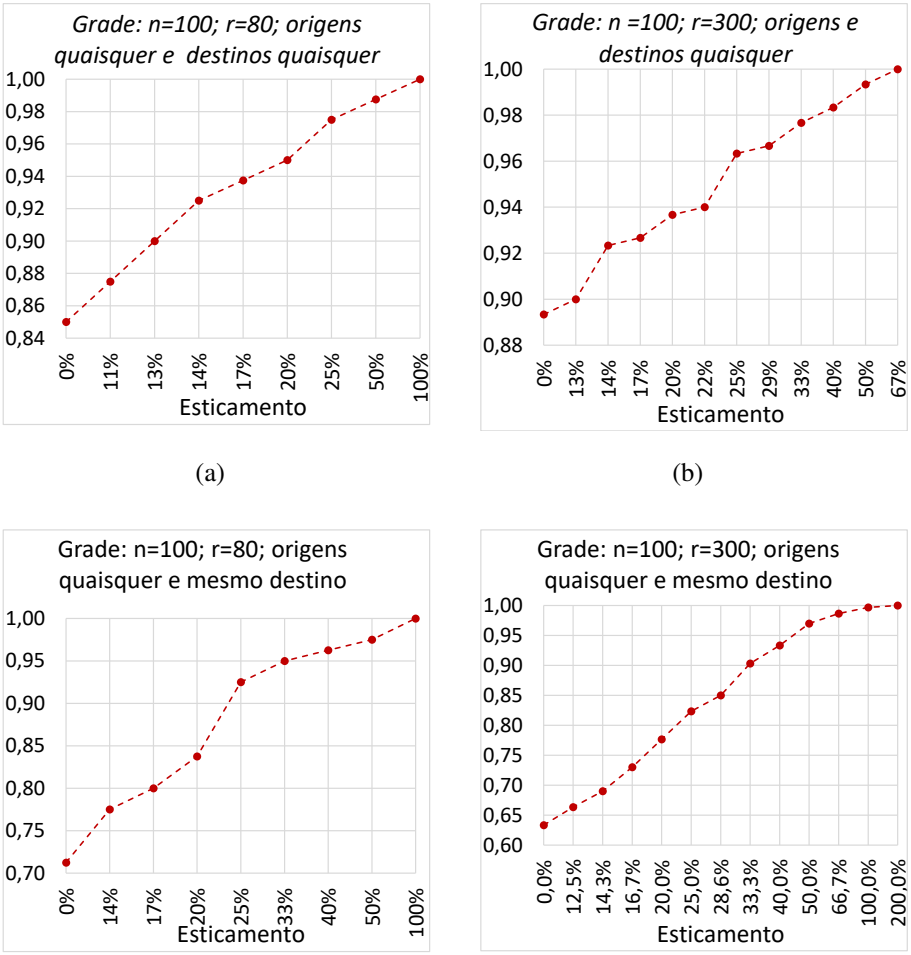


Figura 3.5: A função distribuição cumulativa da porcentagem de aumento no esticamento do comprimento dos caminhos de cada fluxo da última solução Pareto-ótima em relação a primeira solução Pareto-ótima.

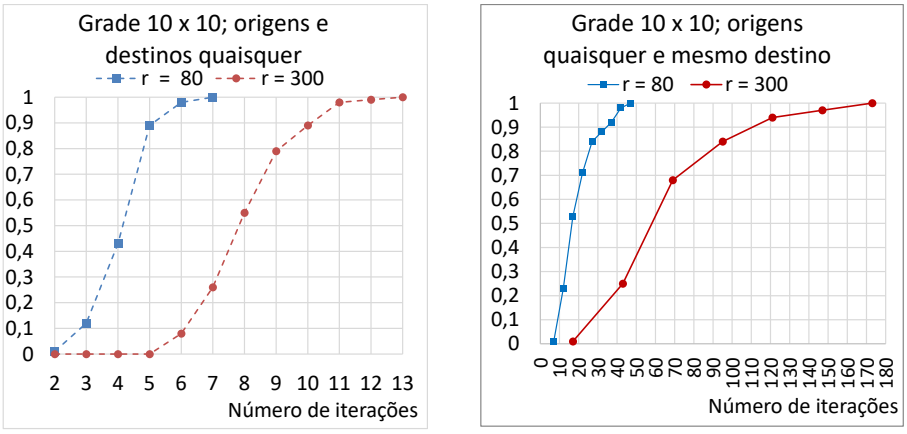
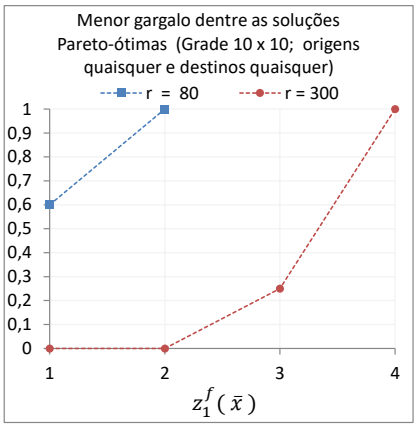
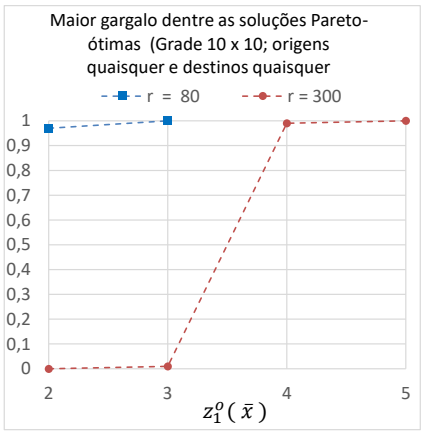


Figura 3.6: A função distribuição cumulativa do número de iterações em cada instância.

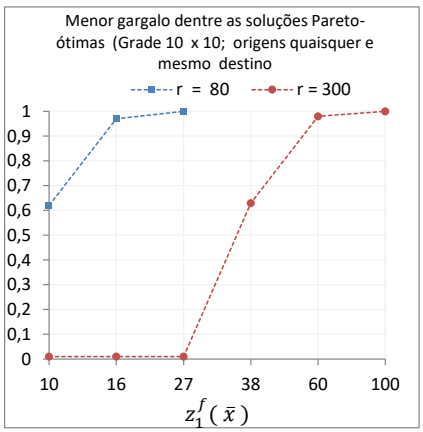
fluxos por instância triplicou o tamanho do caminho e as demais instâncias o acréscimo



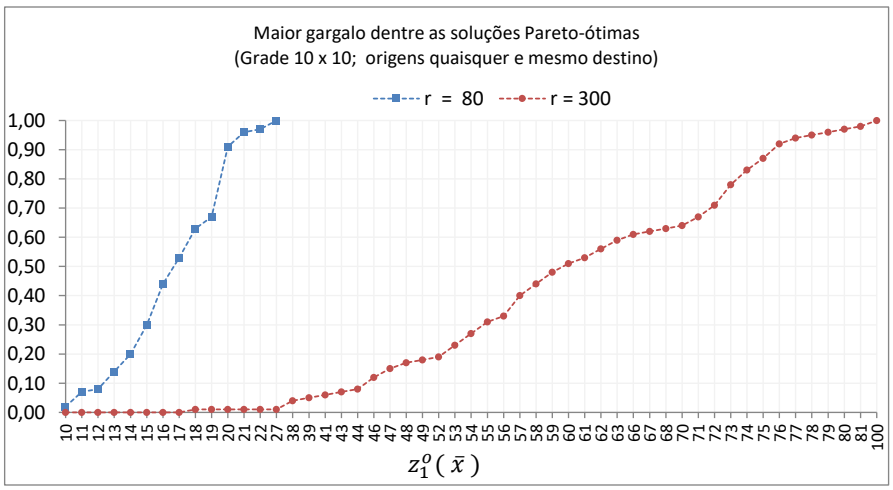
(a)



(b)



(c)



(d)

Figura 3.7: A função distribuição cumulativa dos gargalos $z_1^o(\bar{x})$, da primeira solução (maior gargalo), e $z_1^f(\bar{x})$, da última solução (menor gargalo), Pareto-ótimas geradas pelo Algoritmo 1.

não ultrapassa 66,7% (Figuras 3.5(c) e 3.5(d)).

Nas instâncias onde $|X^*| > 1$, vimos a necessidade de medirmos se existiria ou

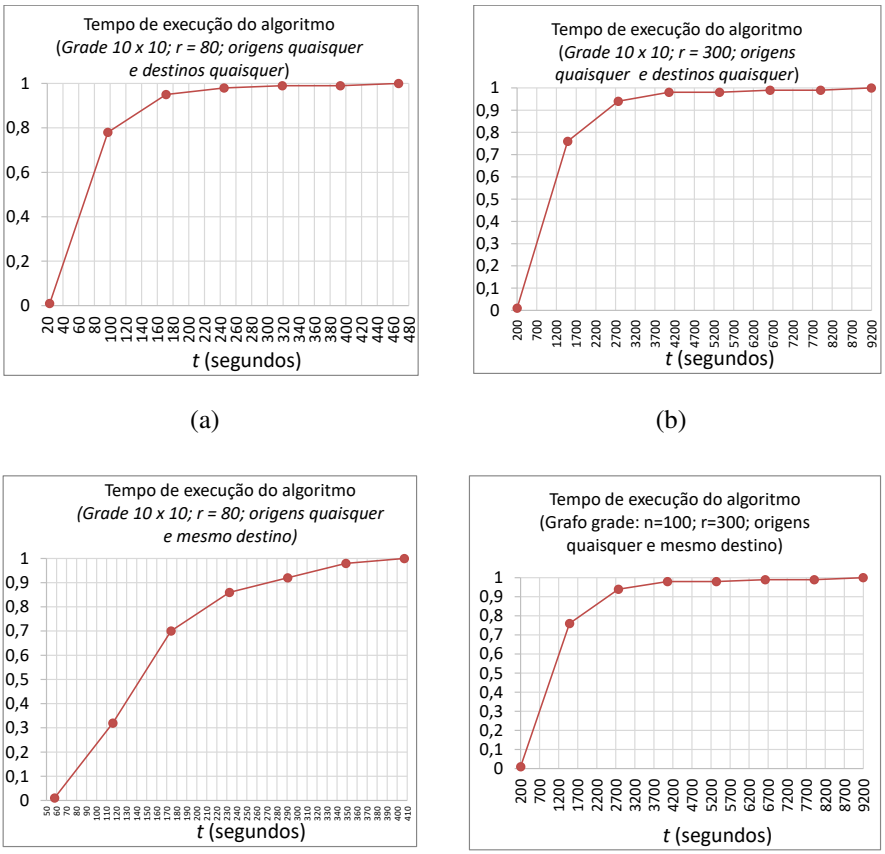


Figura 3.8: A função distribuição cumulativa do tempo de execução do Algoritmo 1 em cada instância.

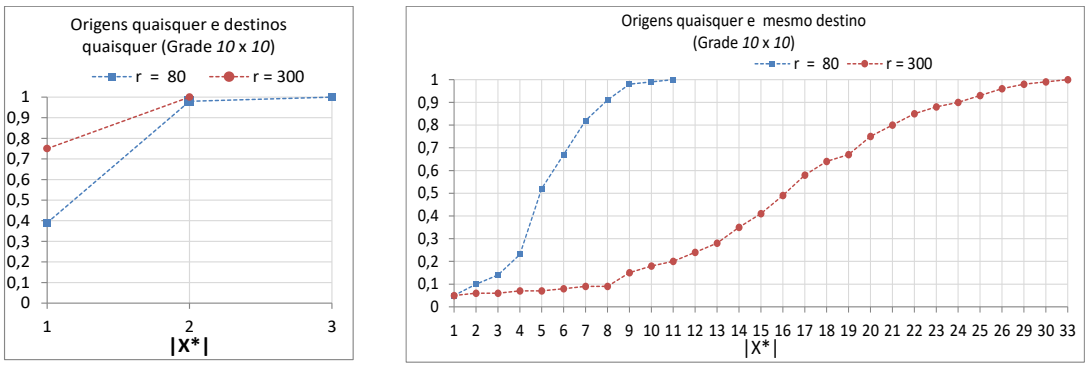


Figura 3.9: A função distribuição cumulativa da cardinalidade do conjunto X^*. A linha pontilhada representa que os resultados para $|X^*|$ que apareceram nos testes foram somente os discriminados nas abscissas.

não penalidades consideráveis de um ou mais fluxos, quanto ao esticamento do total de saltos de cada fluxo da última solução Pareto-ótima em relação a primeira solução Pareto-ótima. Para isso, utilizamos o índice de justiça de Jain [Jain, Chiu e Hawe 1984], que resulta em um número entre 0 e 1. Aqui, esse índice mede a "igualdade" do esticamento do total de saltos de cada fluxo e, quanto mais próximo de 1 o resultado do índice, menor a penalidade entre esses fluxos. É óbvio que o índice não atingirá o valor

máximo (=1). A Figura 3.10 mostra que, em todos os cenários, o índice de justiça foi superior a 0,93.

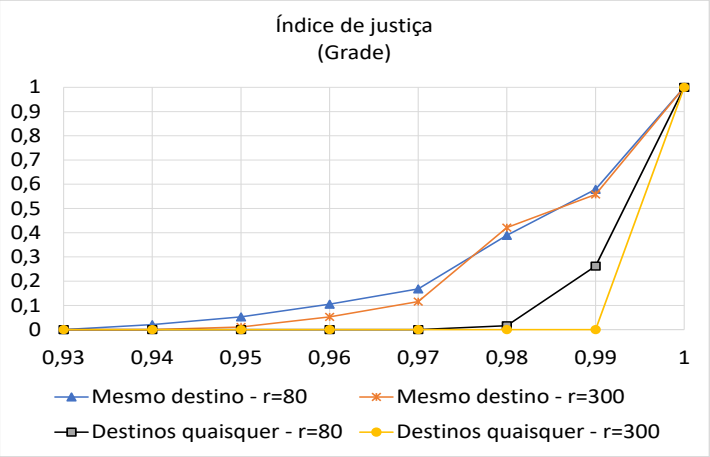


Figura 3.10: A função distribuição cumulativa do índice de justiça.

Observamos uma especificidade para a configuração de distribuição dos fluxos quando todos os r fluxos vão para um mesmo destino d. Considere que m_d representa a quantidade de aresta que chega ao destino d. Assim, o menor valor que z_1 pode assumir para P_ϵ ser viável é $z_1^f(\bar{x}) = \left\lceil \frac{r}{m_d} \right\rceil$, se $r \geq m_d$, veja Figura 3.7(c). Portanto, para essa configuração, podemos prever uma iteração a menos no algoritmo, basta executar o algoritmo até $\epsilon = \epsilon_{min} = \left\lceil \frac{r}{m_d} \right\rceil$. Logo, temos que $|X^*| \leq r - \left\lceil \frac{r}{m_d} \right\rceil + 1$. As Figuras 3.7(c) e 3.7(d) mostram a cdf do gargalo $z_1(\bar{x})$ da última e da primeira solução Pareto-ótima, respectivamente, gerada pelo Algoritmo 1, em cada instância. Observamos que o gargalo máximo da última iteração é de 27 fluxos, para $r = 80$, e 100 fluxos, para $r = 300$. Além disso, 50% das instâncias iniciam com um gargalo inferior a 17 fluxos e 60 fluxos, para $r = 80$ e para $r = 300$, respectivamente.

Quanto ao número de iterações, ele não é tão grande (Figura 3.6(b)) quando comparado com a quantidade que poderia ocorrer (Teorema 3.3), pois o gargalo nas primeiras iterações cai drasticamente, isto é, $z_1(\bar{x}) < \epsilon$. Ocorre, também, muitas soluções dominadas; isso se dá pela própria topologia do grafo grade que, na maioria das vezes, tem mais de uma opção de caminhos de mesmo tamanho. Assim, quanto mais fluxos mais soluções dominadas. Além disso, na metade das instâncias ocorre um tempo inferior a 15 segundos, para $r = 80$, e inferior a 1.200 segundos, para $r = 300$ (Figuras 3.8(c) e 3.8(d)).

3.3.1.2 Análise dos resultados para grafo aleatório

O grafo aleatório utilizado é o modelo denominado Barabási-Albert (BA) [Barabási e Albert 1999]. Esse modelo gera redes com distribuição lei de potência, característica observada em vários sistemas reais, tais como a internet e redes sociais. O grafo inicia-se com n_o nós. Novos nós são adicionados à rede, um a um, e cada um desses novos nós faz conexões com outros $\bar{n}$ nós já existentes na rede, com $\bar{n} \leq n_o$. A probabilidade do novo nó ℓ se conectar a um outro nó i já existente é dada por $p_i = \frac{k_i}{\sum_{j \in V} k_j}$,

onde k_i é o grau do nó i e $\sum_{j \in V} k_j$ é a soma de todos os graus dos nós j já existentes na rede. Esse processo de escolha de conexão é repetido então para cada uma das $\bar{n}$ conexões que o novo nó irá fazer. A quantidade de novos nós que serão adicionados na rede será denotado por $\bar{t}$. Consideramos $n = 100$, $n_o = 3$, $\bar{n} = n_o$, $\bar{t} = 97$ e $m = 2.\bar{n}.\bar{t}$.

Quanto aos nós de origem e de destino de cada fluxo $f \in F$, foram gerados aleatoriamente, considerando as mesmas configurações de distribuição dos fluxos que utilizamos na subseção anterior e, também, consideramos a mesma quantidade de fluxos ($r = 80$ e $r = 300$).

A Tabela 3.4 mostra o número de fluxos (r), a cardinalidade do conjunto mínimo completo ($|X^*|$), o número de iterações que P_ϵ foi resolvido (It), as coordenadas do vetor solução da primeira solução Pareto-ótima gerada (z_1^o e z_2^o), as coordenadas do vetor solução da última solução Pareto-ótima gerada (z_1^f e z_2^f) e o tempo (t) de execução do algoritmo, em segundos. Os resultados apresentados são médias e desvios padrões para 100 instâncias dos conjuntos de resultados.

Tabela 3.4: Média e desvio padrão dos resultados sobre 100 instâncias, considerando $c_{ij}^f = 1$.

Topologia aleatória								
Origens e destinos quaisquer								
r	Valores	$\|X^*\|$	It	z_1^o	z_2^o	z_1^f	z_2^f	t
80	Média	2	4	2	208	2	213	27.61
	Desvio Padrão	0,7	0,7	0,5	6,4	0,5	8,5	5,04
300	Média	3	6	5	778	3	786	174,88
	Desvio Padrão	0,9	0,9	0,9	11,0	0,2	12,1	45,07
Origens quaisquer e mesmo destino								
r	Valores	$\|X^*\|$	It	z_1^o	z_2^o	z_1^f	z_2^f	t
80	Média	7	19	25	211	19	222	127,63
	Desvio Padrão	5,2	8,4	8,1	26,8	7,3	22,6	56,67
300	Média	23	66	97	796	76	833	1.511,35
	Desvio Padrão	18,0	33,1	27,6	76,9	24,4	64,9	764,28

As Figuras 3.11(a) e 3.11(b) mostram a cdf, para a configuração (i) de distribuição dos fluxos, da amplitude total dos valores de $z_2(\bar{x})$, onde $\bar{x} \in X^*$ em cada instância. Em média, 6 saltos por instância, com desvio padrão de 6 saltos para $r = 80$, e para $r = 300$, uma média de 8 saltos, com desvio padrão de 5 saltos (a maior amplitude para $r = 80$ foi 20 saltos e para $r = 300$, foi de 42 saltos).

Quanto ao total de saltos da última solução Pareto ótima comparado com a primeira solução Pareto-ótima, geradas pelo algoritmo em cada instância, de cada fluxo, observamos que para $r = 80$, no mínimo $82,5\%$ dos fluxos, em cada instância, não sofrem alterações na quantidade de saltos. Os que sofrem alterações, na maioria das instâncias, obtiveram um aumento de até $66,7\%$, com exceção de 20% das instâncias que ocorreram de dobrar o tamanho do comprimento do caminho (no máximo em dois fluxos em cada instância) e, em uma única instância ocorreu um aumento de 150% em um único fluxo.

Para $r = 300$, no mínimo $94,7\%$ dos fluxos, em cada instância, não sofrem alterações na quantidade de saltos. Os demais comprimentos de caminhos, na maioria

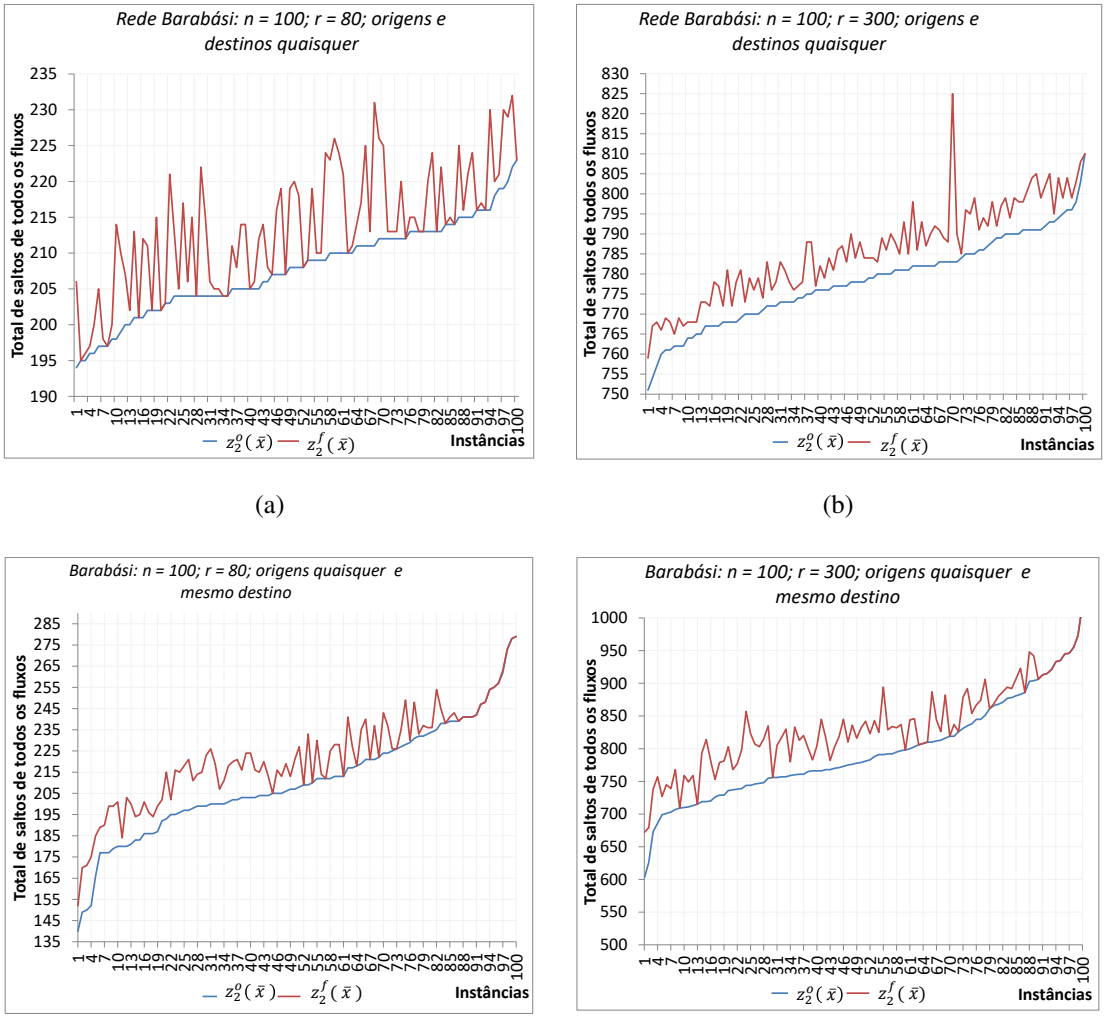


Figura 3.11: Total de saltos de todos os fluxos da primeira solução ($z_2^o(\bar{x})$) e da última solução ($z_2^f(\bar{x})$) Pareto-ótimas gerada pelo Algoritmo 1, em cada instância.

das instâncias, obteve um aumento de até 50%, exceto em sete instâncias que dobram o tamanho do comprimento do caminho (em um único fluxo de cada instância) e, em duas instâncias ocorrem um aumento de 66,7% (Figuras 3.12(a) e 3.12(b)).

A Figura 3.13 mostra que, em todos os cenários, o índice de justiça de Jain foi superior a 0,92.

O fato de termos baixos valores de $z_1(\bar{x})$ (gargalo), como mostra os gráficos das Figuras 3.16(a) e 3.16(b), isso acarreta em poucas iterações (Figura 3.17(a)) e, consequentemente, obtivemos tempos baratos, em média 27,607 segundos, com desvio padrão de 5,041 segundos para $r = 80$ e para $r = 300$, em média 174,882 segundos, com desvio padrão de 45,068 segundos (Figuras 3.14(a) e 3.14(b)), e poucas soluções Pareto-ótimas, em cada instância, como podemos observar na cdf da Figura 3.15(a).

Quando consideramos todos os fluxos com o mesmo (único) destino, verificamos que 23% e 22% das instâncias geraram $|X^*| = 1$, para $r = 80$ e para $r = 300$, respectivamente, e a maior cardinalidade de X^* é de 22 e 79 soluções Pareto-ótimas, para $r = 80$ e $r = 300$, respectivamente, como mostra a Figura 3.15(b).

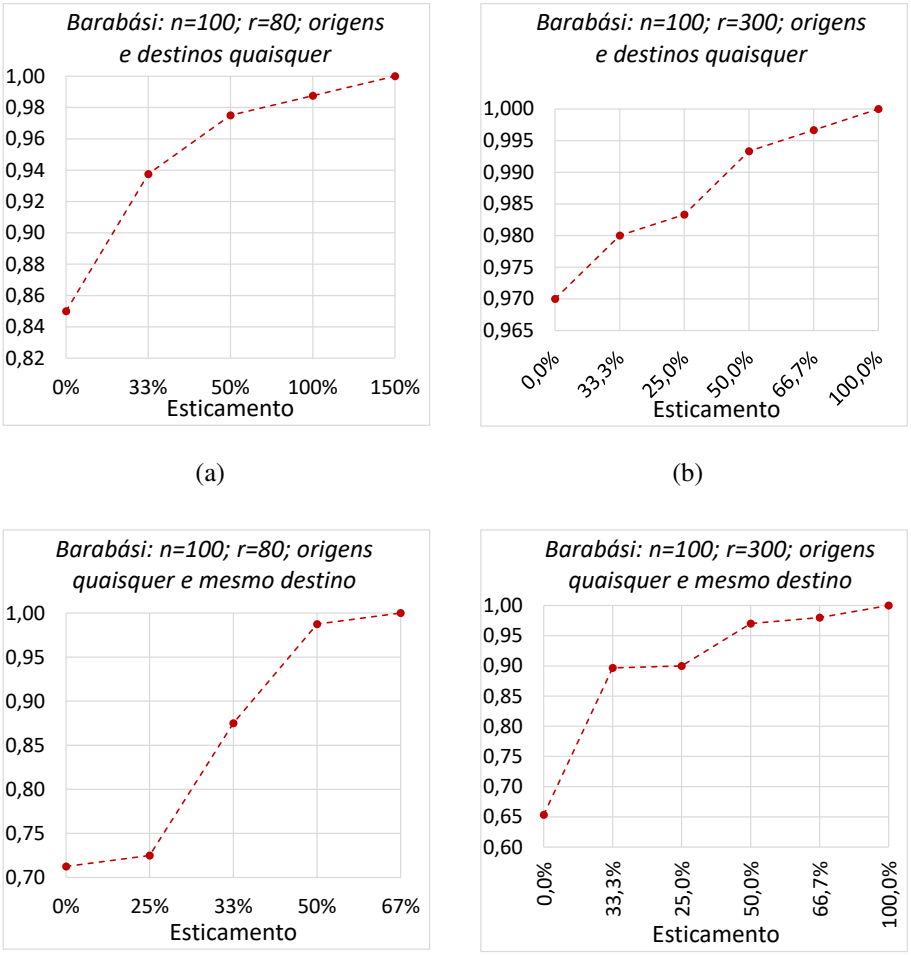


Figura 3.12: A função distribuição cumulativa da porcentagem de aumento no esticamento do comprimento dos caminhos de cada fluxo da última solução Pareto-ótima em relação a primeira solução Pareto-ótima.

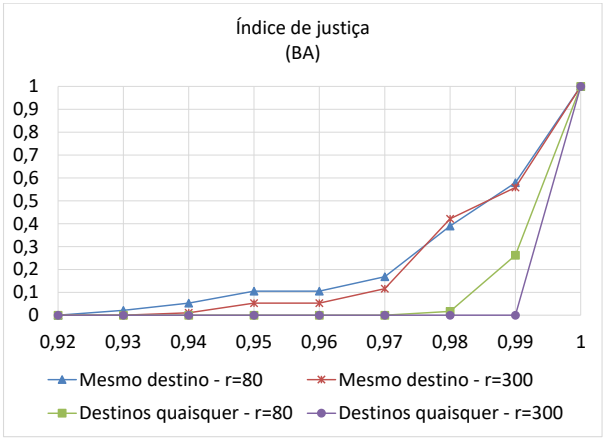


Figura 3.13: A função distribuição cumulativa do índice de justiça.

A amplitude total dentre os valores de $z_2(\bar{x})$, com $\bar{x} \in X^*$ de cada instância varia de zero a 26 saltos, para $r = 80$ e de zero a 113 saltos para $r = 300$, como podemos ver nos gráficos das Figuras 3.11(c) e 3.11(d). Para o total de saltos de cada fluxo observamos

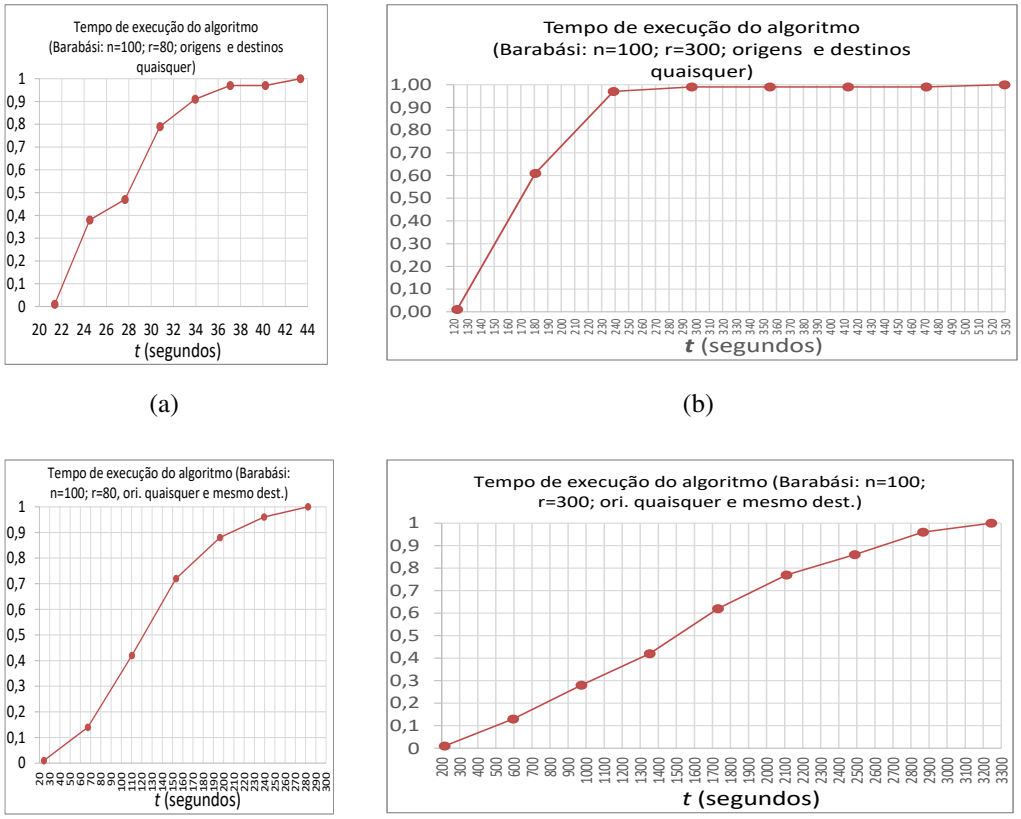


Figura 3.14: A função distribuição cumulativa do tempo de execução do Algoritmo 1 em cada instância.

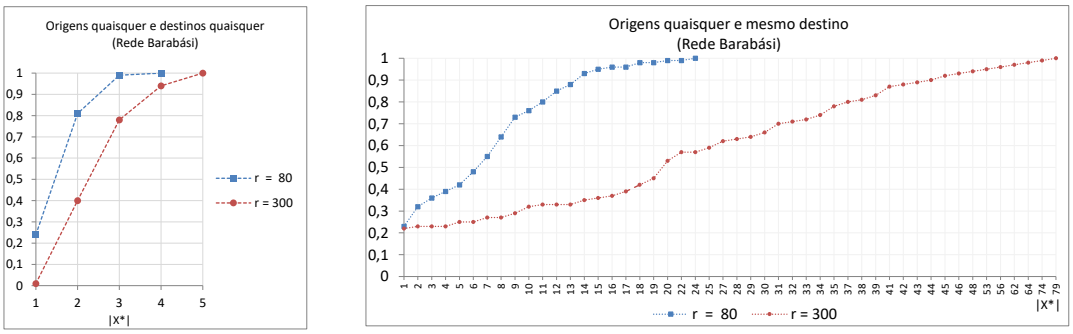
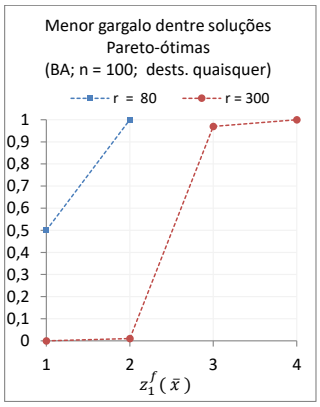


Figura 3.15: A função distribuição cumulativa da cardinalidade do conjunto X^*. A linha pontilhada representa que os resultados para $|X^*|$ que apareceram nos testes são somente os discriminados nas abscissas.

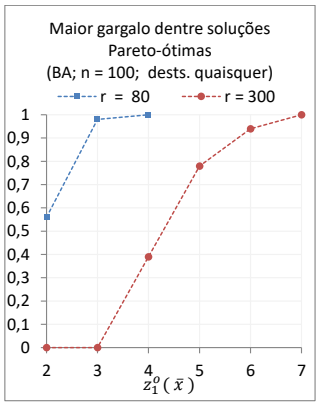
que para $r = 80$ pelo menos 70% dos fluxos, em cada instância, não sofrem alterações na quantidade de saltos, quando comparamos a última solução Pareto-ótima com a primeira solução Pareto ótima. Nos demais fluxos há um aumento de até $66,7\%$ em 85% das instâncias, até 100% em 14% das instâncias, mas ocorrendo no máximo, em cinco fluxos. O tamanho de um fluxo triplicou a quantidade de saltos somente em uma única instância (ver Figura 3.12(c)).

Para $r = 300$, no mínimo $73,3\%$ dos fluxos de cada instância não sofrem

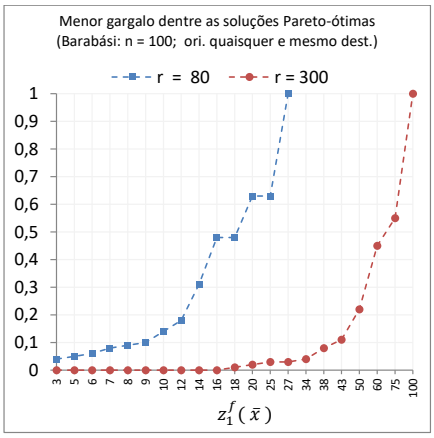
alterações na quantidade de saltos e os demais fluxos há um aumento de até 66,7%, com exceção de 15% que, no máximo 7 fluxos em cada instância, dobram o comprimento do caminho (Figura 3.12(d)).



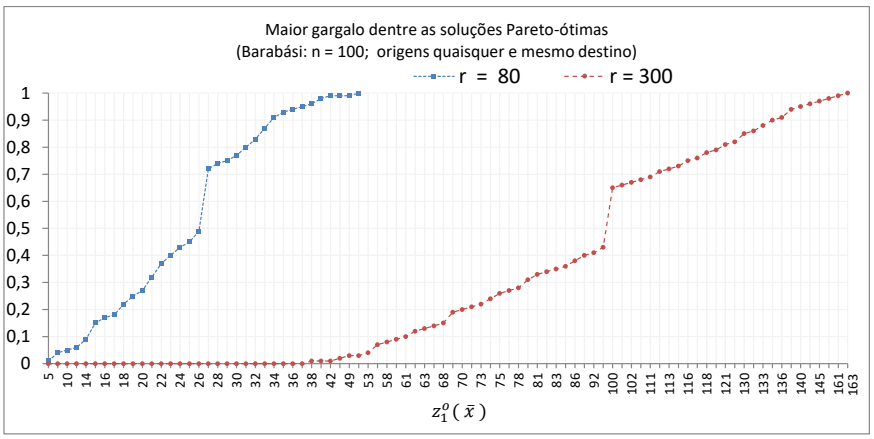
(a)



(b)



(c)



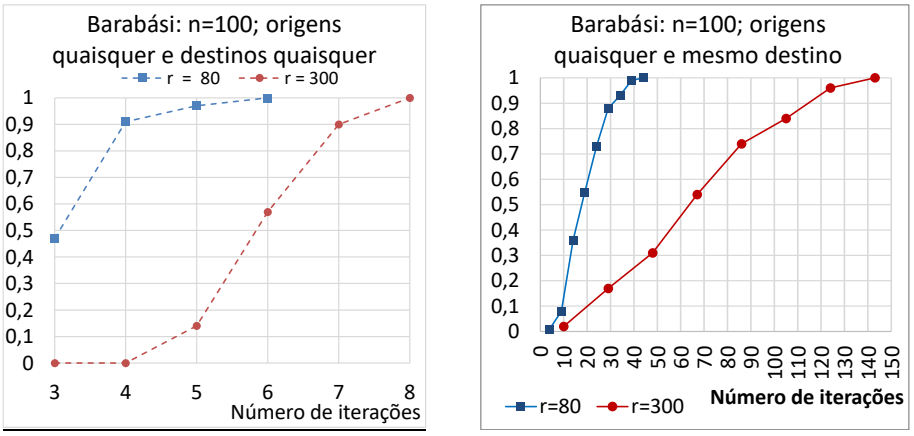
(d)

Figura 3.16: A função distribuição cumulativa dos gargalos da primeira solução ($z_1^o(\bar{x})$) e da última solução ($z_1^f(\bar{x})$), Pareto-ótimas geradas pelo Algoritmo 1.

O maior gargalo é de 50 fluxos, para $r = 80$, e 163 fluxos, para $r = 300$, ou seja,

em nenhuma instância, ocorre de atingir o gargalo máximo igual a r. Já, o menor gargalo para $r = 80$ é de 3 fluxos e para $r = 300$, 18 fluxos. Mais detalhes pode ser visto nos gráficos das Figuras 3.16(c) e 3.16(d) que mostram a cdf dos gargalos $z_1(\bar{x})$ que ocorrem na primeira e na última solução Pareto-ótima, respectivamente, gerada pelo Algoritmo 1, em cada instância.

A Figura 3.17(b) mostra que em 50% das instâncias ocorre menos do que 20 iterações, para $r = 80$ e menos do que 65 iterações, para $r = 300$.



(a) (b)

Figura 3.16: A função distribuição cumulativa da quantidade de iterações em cada instância.

3.3.2 Análise dos resultados para custos variados nas arestas

Os resultados computacionais são referentes à implementação do Algoritmo 1 para a resolução do problema (P). Nesse caso o problema (P) busca rotear os r fluxos, minimizando o gargalo da rede e minimizar o custo total de todos os fluxos, definidos sobre o grafo grade e o grafo aleatório, utilizando o modelo Barabási-Albert (BA) [Barabási e Albert 1999]. Os custos foram gerados aleatoriamente com $c_{ij}^f \in \{1,2,3,4,5\}$, para todas arestas $(i,j) \in E$ e para todos os fluxos $f \in F$.

Embora esses resultados não adicionem informações relevantes ao trabalho, eles estendem sensivelmente a avaliação computacional.

As tabelas a seguir mostram o número de fluxos (r), a cardinalidade do conjunto mínimo completo $(|X^*|)$, o número de iterações que P_ϵ foi resolvido (It), as coordenadas do vetor objetivo da primeira solução Pareto-ótima gerada $(z_1^o$ e $z_2^o)$, as coordenadas do vetor objetivo da última solução Pareto-ótima gerada $(z_1^f$ e $z_2^f)$ e o tempo (t) de execução do algoritmo, em segundos. Os resultados apresentados são médias e desvios padrões para 100 instâncias dos conjuntos de resultados.

A cardinalidade do conjunto X^*, com custos c_{ij}^f variados, acarreta em um acréscimo significativo, comparado com os resultados quando consideramos os custos das arestas todos iguais $(c_{ij}^f = 1)$. Isso ocorre pois, com custos variados, a chance de acharmos uma única solução ótima para P_ϵ, em cada iteração, aumenta, assim essa solução é Pareto-ótima para (P) (veja [Chankong e Haimes 2008]), ou seja, das soluções geradas pelo algoritmo, poucas são deletadas (veja Figura 3.17). Além disso, o número de iterações aumenta, pois como os custos das arestas não são todos iguais, temos mais

Tabela 3.5: Média e desvio padrão sobre 100 instâncias com custos variados.

Topologia Grade								
Origens e destinos quaisquer								
r	Valor	$\|X^*\|$	It	z_1^o	z_2^o	z_1^f	z_2^f	t
80	Média	7	9	8	608	1	807	162,98
	Desvio Padrão	1,5	1,4	1,4	28,4	0,5	96,6	255,32
300	Média	20	24	23	2.335	4	2.934	1.086,53
	Desvio Padrão	2,8	3,0	2,8	78,7	0,3	153,5	664,27
Origens quaisquer e mesmo destino								
r	Valor	$\|X^*\|$	It	z_1^o	z_2^o	z_1^f	z_2^f	t
80	Média	28	35	39	601	12	712	77,64
	Desvio Padrão	11,7	12,4	12,9	96,9	3,3	108,6	30,45
300	Média	105	126	152	2277	48	2711	1.220,98
	Desvio Padrão	51,5	49,8	55,0	415,5	14,8	430,7	504,88

Tabela 3.6: Média e desvio padrão sobre 100 instâncias com custos variados.

Topologia aleatória								
Origens e destinos quaisquer								
r	Valor	$\|X^*\|$	It	z_1^o	z_2^o	z_1^f	z_2^f	t
80	Média	5	6	5	387	1	440	14,17
	Desvio Padrão	1,2	1,3	1,1	13,7	0,5	36,3	2,77
300	Média	12	17	14	1.456	3	1.643	155,23
	Desvio Padrão	2,2	2,7	2,2	26,7	0,3	42,9	22,58
Origens quaisquer e mesmo destino								
r	Valor	$\|X^*\|$	It	z_1^o	z_2^o	z_1^f	z_2^f	t
80	Média	29	36	46	378	19	459	67,83
	Desvio Padrão	14,6	14,0	17,3	70,2	7,1	60,3	27,97
300	Média	97	117	165	1.474	69	1.749	885,81
	Desvio Padrão	56,6	54,1	67,2	273,0	26,5	240,8	420,41

opções de caminhos com comprimentos diferentes para cada fluxo, assim $\epsilon \in \{1, 2, \cdots, r\}$ assume mais valores desse conjunto.

Consequentemente, o tempo de execução do algoritmo aumenta, também. Obviamente, a amplitude do custo total de todos os fluxos da última e da primeira soluções Pareto-ótima gerada pelo o algoritmo também ocorre um aumento significativo, veja as Tabelas 3.5 e 3.6.

Portanto, se assumíssemos uma quantidade suficientemente grande de valores distintos para os custos das arestas, a chance de ocorrer que nenhuma solução gerada seja deletada, ou seja, que todas as soluções geradas sejam Pareto-ótimas de (P), é muito grande.

Nos resultados computacionais, observamos que para instâncias com fluxos de origens e/ou destinos variados, o número de soluções Pareto-ótimas no conjunto mínimo

completo é extremamente baixo. Isso consiste em um resultado importante, pois facilita a tomada de decisão em torno de qual solução escolher para fazer o roteamento dos fluxos.

Alguns resultados (tabelas e gráficos) extras se encontram disponíveis em https://github.com/LABORA-INF-UFG/paper-LKKN-2018-extras.

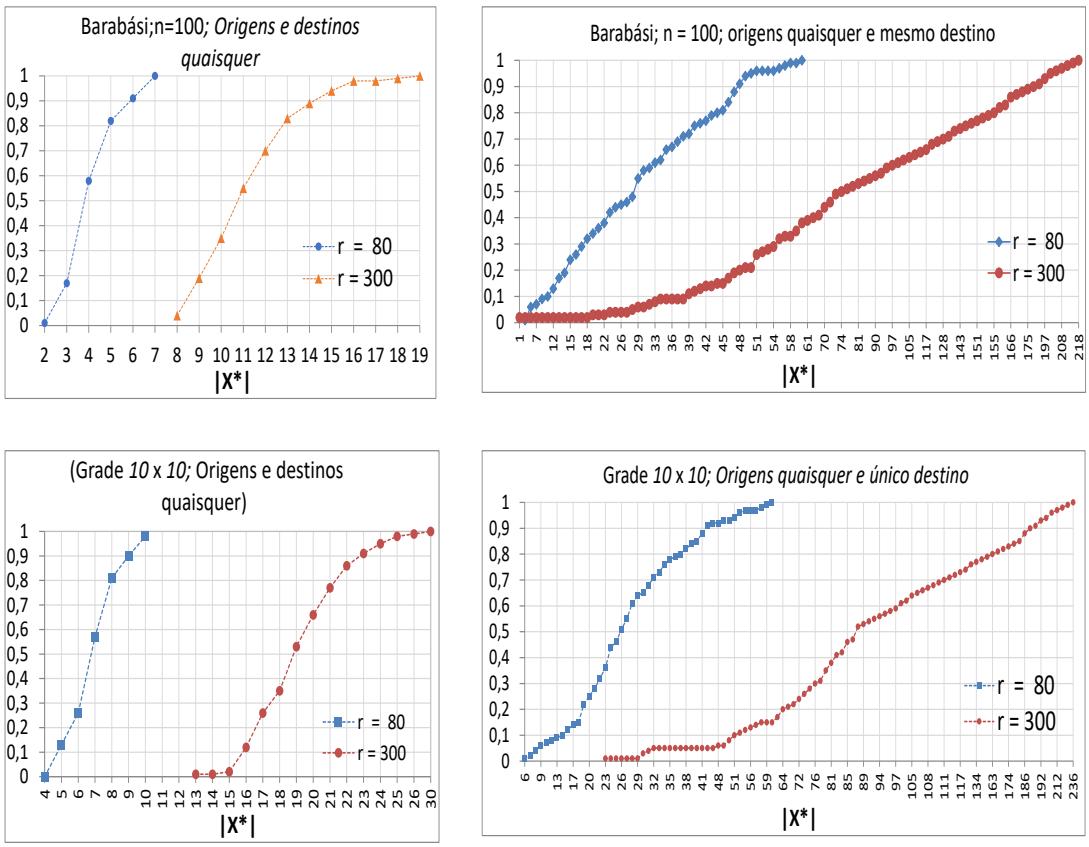


Figura 3.17: A função distribuição cumulativa da cardinalidade de X^* com custos variados.

CAPÍTULO 4

Um problema biobjetivo de roteamento de fluxos com diferentes qualidades nas arestas

O trabalho apresentado neste capítulo deu origem ao artigo [Fernandes, Pinto e Cardoso 2019]. Nele apresentamos um modelo de problema de roteamento de fluxos em redes de múltiplos saltos nos quais os enlaces podem assumir qualidades diferentes. Ou seja, o que diferencia esse modelo do modelo apresentado anteriormente é que nessa proposta a função gargalo leva em conta os pesos dos fluxos e as qualidades dos enlaces (arestas), ao invés de considerarmos apenas o número de fluxos sobre as arestas. Surge, daí, o modelo biobjetivo ($\overline{P}$), apresentado a seguir.

4.1 O Problema ($\overline{P}$)

Considere um conjunto de fluxos F, com $|F| = r$, a ser roteado em um grafo orientado $G = (V, E)$, onde V representa o conjunto de nós e E o conjunto de arestas, com $|V| = n$ e $|E| = m$. Cada fluxo $f \in F$ deve ser roteado por um único caminho em G, indo da sua origem $s^f \in V$ ao seu destino $d^f \in V$. Para cada aresta $(i,j) \in E$, indo de $i \in V$ para $j \in V$, e fluxo $f \in F$, definimos uma variável binária x_{ij}^f, que irá indicar se o fluxo f vai passar (ou não) pela aresta (i,j).

Seja q_{ij} um valor inteiro não negativo associado a cada aresta $(i,j) \in E$, correspondendo à qualidade dessa aresta, com $q_{ij} \in \{q_1, q_2, \cdots, q_\alpha\}$ e $q_1 < q_2 < \cdots < q_\alpha$. Considere, ainda, $p^f \in \mathbb{Z}_{\geq 0}$ o peso de cada fluxo $f \in F$.

A soma ponderada desses dois parâmetros, denotada por w_{ij}^f, corresponde ao peso do fluxo $f \in F$ para atravessar a aresta $(i,j) \in E$, isto é, $w_{ij}^f = \beta_1.p^f + \beta_2.q_{ij}$, $\forall f \in F$ e $\forall (i,j) \in E$, onde os valores β_1, $\beta_2 \in \mathbb{Z}_{\geq 0}$ são parâmetros que permitem atribuir o grau de importância de p^f e q_{ij}, respectivamente. Sem perda de generalidade assumimos que quanto maior o valor de w_{ij}^f pior será esse peso.

O último parâmetro definido é c_{ij}^f, o custo para o fluxo f utilizar a aresta (i,j).

Assim, temos o gargalo da rede, definido como a carga da(s) aresta(s) mais pesada(s), dado por $z_1(x) = \max\limits_{(i,j) \in E} \left\{ \sum\limits_{f \in F} w_{ij}^f.x_{ij}^f \right\}$ e o custo total de todos os fluxos dado por $z_2(x) = \sum\limits_{f \in F} \sum\limits_{(i,j) \in E} c_{ij}^f.x_{ij}^f$.

O modelo de programação biobjetivo, formulado para o problema, de rotear os r fluxos, minimizando o gargalo da rede e o custo total do roteamento, segue-se abaixo.

$$(\overline{P}) \quad minimizar \left\{ \max_{(i,j) \in E} \left\{ \sum_{f \in F} w_{ij}^f . x_{ij}^f \right\} \right\} \tag{4-1}$$

$$minimizar \left\{ \sum_{f \in F} \sum_{(i,j) \in E} c_{ij}^f . x_{ij}^f \right\} \tag{4-2}$$

sujeito a:

$$\sum_{(i,j) \in E} x_{ij}^f - \sum_{(j,i) \in E} x_{ji}^f = \begin{cases} 1, & \text{se } i = s^f \\ 0, & \forall i \in V - \left\{ d^f, s^f \right\} \\ -1, & \text{se } i = d^f \end{cases} \qquad \forall f \in F \tag{4-3}$$

$$x_{ij}^f \in \{0,1\}, \ \forall (i,j) \in E \text{ e } \forall f \in F \tag{4-4}$$

As restrições garantem que cada fluxo $f \in F$ saia da sua origem, s^f, e chegue ao seu destino, d^f, passando por exatamente um caminho.

As funções (4-1) e (4-2) são conflitantes, pois minimizar a carga da(s) aresta(s) mais pesada(s) deve na maioria das vezes implicar em custos maiores (ou caminhos mais longos quando $c_{ij}^f = 1$) para roteamento dos fluxos, mas nunca custos menores. Da mesma forma, minimizar o custo total (ou número total de saltos) deve levar a um maior valor para a função gargalo.

Nesse modelo observe que se considerássemos $\beta_1 = 1$ e $\beta_2 = 0$ teríamos como objetivo da função gargalo, minimizar o máximo da soma dos pesos dos fluxos sobre as arestas e se $\beta_1 = 0$ e $\beta_2 = 1$, a função gargalo estaria minimizando a carga máxima sobre as arestas, levando em conta as qualidades das arestas.

4.2 O Algoritmo

A fim de resolver $(\overline{P})$, nós utilizamos o Algoritmo 1 com algumas adaptações. No final do algoritmo obtemos um conjunto mínimo completo, X^*, de soluções Pareto-ótimas para $(\overline{P})$

Considere o subproblema mono-objetivo, denotado por $(\overline{P}_\epsilon)$, onde a função objetivo é dada por z_2, isto é, a função totalizadora (4-2) do problema biobjetivo $(\overline{P})$. As restrições de $(\overline{P}_\epsilon)$ são (4-3) e (4-4) de $(\overline{P})$, juntamente com uma restrição sobre o gargalo, a qual é definida através do parâmetro inteiro positivo, denotado por ϵ. Esse parâmetro é usado para controlar os valores da função gargalo e para garantir a otimalidade do algoritmo. O subproblema de programação linear inteira é apresentado a seguir.

$$(\overline{P}_\epsilon) \quad minimizar \left\{ \sum_{f \in F} \sum_{(i,j) \in E} c_{ij}^f . x_{ij}^f \right\}$$

sujeito a:

$$\sum_{(i,j)\in E} x_{ij}^f - \sum_{(j,i)\in E} x_{ji}^f = \begin{cases} 1, & \text{se } i = s^f \\ 0, & \forall i \in V - \{d^f, s^f\} \\ -1, & \text{se } i = d^f \end{cases} \qquad \forall f \in F$$

$$\sum_{f\in F} w_{ij}^f . x_{ij}^f \leq \epsilon\,,\ \forall (i,j) \in E \tag{4-5}$$

$$x_{ij}^f \in \{0,1\}\,,\ \forall (i,j) \in E \text{ e } \forall f \in F$$

A escolha de manter a função totalizadora (4-2) como função objetivo, e tratar a função gargalo (4-1) como restrição é pelo mesmo motivo do Problema (P): (4-2) ser uma função linear e (4-1) não ser. Porém, as restrições (4-5) associadas ao parâmetro ϵ, limitando os valores de (4-1), são lineares. Logo, esta estratégia nos permite resolver o problema $(\overline{P})$ por programação linear.

Para resolver o problema $(\overline{P})$, adaptamos o Algoritmo 1 proposto no capítulo anterior, como apresentado a seguir.

Algoritmo 2

1. $X^* \leftarrow \emptyset$; $\epsilon \leftarrow \epsilon_0$.
2. ***Enquanto*** $(\overline{P}_\epsilon)$ *não for inviável* ***faça***
3. $\quad \bar{x} \leftarrow$ *solução ótima de* $(\overline{P}_\epsilon)$
4. $\quad X^* \leftarrow X^* \cup \{\bar{x}\}$
5. $\quad \epsilon \leftarrow z_1(\bar{x}) - 1$
6. $\quad$ ***Se*** $|X^*| > 1$ ***e*** $z_2(\bar{x}) = z_2(\hat{x})$ ***então***
7. $\quad\quad X^* \leftarrow X^* - \{\hat{x}\}$
8. $\quad \hat{x} \leftarrow \bar{x}$

O algoritmo resolve $(\overline{P}_\epsilon)$ repetidamente, empregando, em cada iteração, uma atualização no valor de ϵ. Como é levado em consideração as qualidades das arestas e os pesos dos fluxos em z_1, então o maior gargalo ocorrerá quando todos os fluxos passarem pela aresta de pior qualidade. Por esse motivo iniciamos com $\epsilon = \epsilon_0$, onde $\epsilon_0 = \sum_{f\in F} \left[\beta_1 . p^f + \beta_2 . q_\alpha\right]$. E, após a obtenção da solução ótima $\bar{x}$ para $(\overline{P}_\epsilon)$, fazemos $\epsilon = z_1(\bar{x}) - 1$. Dessa forma, o gargalo da nova solução, $\bar{x}$, obtida na iteração seguinte, será menor do que o da anterior, renomeada para $\hat{x}$. Então, caso o valor de z_2 seja o mesmo para essas duas soluções, temos que a solução anterior é dominada pela nova solução, daí a solução anterior é deletada. Assim, segue o algoritmo até uma iteração em que $(\overline{P}_\epsilon)$ seja inviável. É importante observar que para qualquer solução viável x para o problema $(\overline{P})$ temos $z_1(x) \in \mathbb{Z}_{\geq 0}$, pois os valores de w_{ij}^f e x_{ij}^f são inteiros não negativos. Logo, não existe nenhuma solução viável com valor de z_1 maior do que $z_1(\bar{x}) - 1$ e menor do que $z_1(\bar{x})$. Isso é o que garante que no final, o Algoritmo 2 gera um conjunto mínimo completo de soluções Pareto-ótimas X^* to $(\overline{P})$.

Observe que o valor mínimo que ϵ pode atingir (melhor roteamento possível com relação ao gargalo) é $\epsilon_f = \beta_1 . \overline{p} + \beta_2 . q_1$, onde $\overline{p} = \max_{f\in F}\{p^f\}$. De fato, pois atingimos o menor gargalo quando as arestas que passam os fluxos são as arestas de melhor

qualidade, ou seja, os valores das qualidades dessas arestas são todas iguais a q_1 e, além disso, em cada uma delas passam um único fluxo. Como, $\overline{p} \geq p^f$, $\forall f \in F$ então $\beta_1.\overline{p} \geq \beta_1.p^f \Leftrightarrow \beta_1.\overline{p} + \beta_2.q_1 \geq \beta_1.p^f + \beta_2.q_1$

Como ϵ decresce pelo menos uma unidade em cada iteração, $(\overline{P}_\epsilon)$ é resolvido no máximo para $\hat{e}$ iterações, onde $\hat{e} = \epsilon_0 - \epsilon_f + 1$. Portanto, a complexidade do algoritmo para resolver o problema $(\overline{P})$ é $O(\hat{e}.g(\epsilon))$, onde $g(\epsilon)$ é a complexidade de $(\overline{P}_\epsilon)$. Além disso, a cardinalidade de X^* é limitada por $\hat{e}$, isto é, $|X^*| \leq \hat{e}$, pois no máximo uma solução ótima é inserida em X^* por iteração.

4.3 Resultados Computacionais

Os resultados computacionais são referentes à implementação do Algoritmo 1 para a resolução do problema $(\overline{P})$, definido sobre grafos. As implementações foram feitas em linguagem C++ e compiladas com o Microsoft Visual C++ 2010. O modelo matemático foi implementado em C++ utilizando a biblioteca Concert do CPLEX 12.4 e os experimentos foram executados em um PC com processador Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz e memória RAM de 16Gb.

A topologia do grafo considerada é o grafo aleatório do modelo Barabási-Albert (BA). Consideramos $n = 100$, $n_o = 3$, $\bar{n} = n_o$, $\bar{t} = 97$ e $m = 2.\bar{n}.\bar{t}$. É utilizado o mesmo grafo, gerado aleatoriamente, para todos os cenários considerados. Os nós de origem e de destino de cada fluxo $f \in F$ também foram gerados aleatoriamente, considerando duas configurações distintas para a distribuição dos fluxos. Na primeira, todos os fluxos têm o mesmo destino e as origens são variadas. Na segunda, todos os fluxos têm origens e destinos quaisquer. A quantidade de fluxos considerada foi de $r = 40$ e $r = 80$.

Sabemos que a qualidade da aresta afeta a capacidade de efetividade da aresta e, consequentemente, o peso do fluxo sobre a aresta é, também, afetado pela qualidade da aresta. Assim, consideramos $p^f = \left\lceil \frac{\Gamma^*}{\Gamma^f} \right\rceil$, onde Γ^f é o comprimento de caminho mínimo de cada fluxo $f \in F$ e $\Gamma^* = \max_{f \in F}\{\Gamma^f\}$.

Para esses parâmetros do problema, consideramos $\beta_1 = \left\lceil \theta_1.\frac{q_\alpha}{\overline{p}} \right\rceil$, onde $\overline{p} = \max_{f \in F}\{p^f\}$ e $\beta_2 = \theta_2$. Os pares θ_1 e θ_2, de entradas inteiras, assumiram os seguintes valores: $(\theta_1;\theta_2) = (1;1),\ (1;10)$ e $(10;1)$.

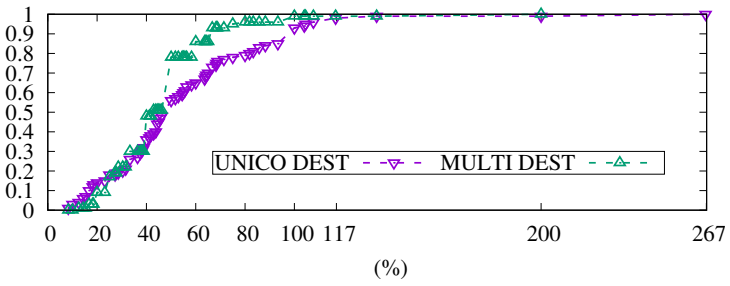
Para cada aresta foram designados, aleatoriamente, um dos três valores que representam as qualidades das arestas, isto é, qualidades baixa, média e alta. Em outras palavras, $q_{ij} \in \{q_1, q_2, q_3\}$, ou seja, $\alpha = 3$. Os custos $c^f_{ij} = 1$, $\forall f \in F$ e $\forall (i,j) \in E$.

As métricas para a avaliação do algoritmo na topologia utilizada são: a cardinalidade do conjunto mínimo completo das soluções Pareto-ótimas $(|X^*|)$, o número de iterações e o tempo de execução. Adotamos linhas pontilhadas nos gráficos das funções de distribuição cumulativa da cardinalidade e do número de iterações para mostrar que os valores ocorridos nos testes foram, somente, os discriminados nas abscissas, não havendo nenhum valor entre eles.

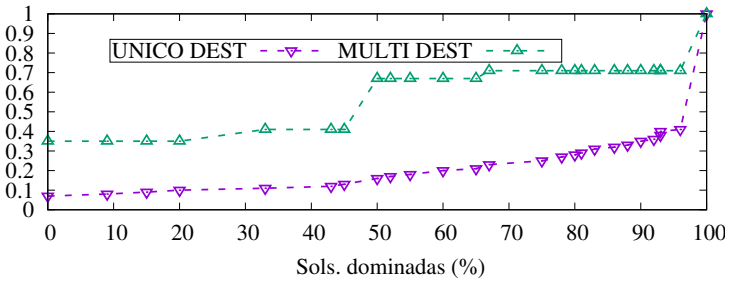
4.3.1 Avaliação Computacional

Nesta seção o objetivo é mostrar que a abordagem apresentada no Capítulo 3 não garante a geração de um conjunto mínimo completo de soluções Pareto-ótimas para

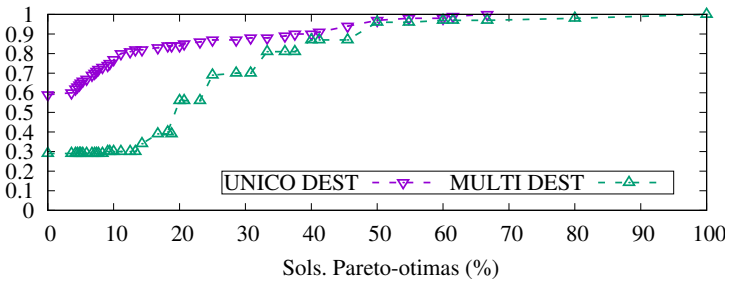
o problema ($\overline{P}$). Para essa análise consideramos 80 fluxos e $\theta_1 = \theta_2 = 1$.



(a)



(b)



(c)

Figura 4.1: Comparação da abordagem apresentada no Capítulo 3 em relação a abordagem apresentada neste Capítulo.

A Figura 4.1(a) mostra a função distribuição cumulativa (cdf) da porcentagem da cardinalidade do conjunto de soluções, que denotaremos aqui por $\overline{X}$, gerado pela abordagem apresentada no Capítulo 3 em relação a cardinalidade do conjunto mínimo completo de soluções Pareto-ótimas X^* de ($\overline{P}$). Em ambas abordagens consideramos os mesmos conjuntos de cenários. Nas duas configurações de distribuição dos fluxos, observamos que na maioria das instâncias $|\overline{X}| < |X^*|$. Apenas em 7% das instâncias $|\overline{X}| \geq |X^*|$, quando consideramos a configuração, único destino e, para configuração, múltiplos fluxos, somente 2% das instâncias $|\overline{X}| \geq |X^*|$. Isso ocorre pois como aqui consideramos os pesos (qualidade e pesos dos fluxos) sobre as arestas, o parâmetro ϵ inicia com um valor bem maior do que a quantidade de fluxos a ser roteado, então a possibilidade de acharmos mais soluções na abordagem atual é maior do que a abordagem anterior.

Agora, o intuito é analisar a porcentagem de soluções pertencentes a $\overline{X}$ que são dominadas para ($\overline{P}$). A Figura 4.1(b) mostra a cdf da porcentagem dessas soluções de $\overline{X}$ que são soluções dominadas para o problema ($\overline{P}$). Quando determinamos os vetores

objetivo para $(\overline{P})$, utilizando as soluções de $\overline{X}$, observamos que em 60% das instâncias obtemos 96% ou mais dessas soluções são dominadas para o problema $(\overline{P})$, quando consideramos a configuração, único destino. Para a configuração, múltiplos destinos, em 33% das instâncias observamos que metade ou mais dessas soluções são soluções dominadas para $(\overline{P})$. Na primeira abordagem, o algoritmo, para a próxima iteração, diminui um fluxo no limite de fluxos que atravessará as arestas, aqui diminuir uma unidade muitas vezes nem mexamos na quantidade de fluxos que atravessarão as arestas, mas pode ser que só trocamos fluxos de pesos diferentes para diminuir o gargalo do roteamento, assim a solução dessa abordagem não necessariamente será solução Pareto-ótima para (P).

A Figura 4.1(c) apresenta a cdf da porcentagem da quantidade de soluções de $\overline{X}$ que são Pareto-ótimas para $(\overline{P})$ (isto é, que são soluções também pertencentes a X^*) em relação a cardinalidade de X^* do problema $(\overline{P})$. Observamos que quando consideramos a configuração, único destino, em 60% das instâncias nenhuma solução de $\overline{X}$ é solução Pareto-ótima para o problema $(\overline{P})$ e, para configuração, múltiplos destinos, 30% das instâncias. Além disso, quando consideramos a configuração de distribuição dos fluxos, único destino, a abordagem anterior (Capítulo 3) não gera um conjunto mínimo completo para $(\overline{P})$ em nenhuma das instâncias e para configuração de distribuição dos fluxos, múltiplos destinos, gera um conjunto mínimo completo em apenas 2% das instâncias e isto somente ocorre quando $|X^*| \leq 2$.

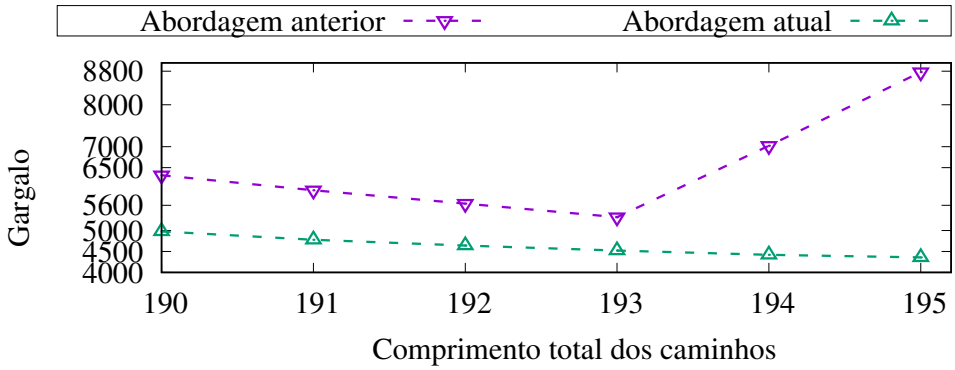


Figura 4.2: Comparação do vetor objetivo das soluções de $\overline{X}$ com o vetor objetivo de soluções de X^*.

Além disso, para configuração múltiplas origens e um único destino, os valores dos gargalos das soluções de $\overline{X}$ e que são dominadas para $(\overline{P})$, assumem valores bem maiores do que os valores de $z_1(\bar{x})$ das soluções Pareto-ótimas de $(\overline{P})$ (por exemplo, quando $\theta_1 = 1$ e $\theta_2 = 10$, veja Figura 4.2).

Os resultados para $\theta_1 > \theta_2$ e $\theta_1 < \theta_2$ não estão apresentados aqui, mas o comportamento geral é similar ao apresentado quando $\theta_1 = \theta_2$.

Essas análises mostram que a abordagem apresentada no capítulo anterior não garante a geração de um conjunto mínimo completo de soluções Pareto-ótimas para o problema $(\overline{P})$. Isso se dá pelo fato de que nessa abordagem não é levado em conta a qualidade e os pesos dos fluxos sobre os enlaces.

4.3.2 Análise dos resultados

Os testes mostram que o número de iterações é, principalmente, influenciado pela distribuição dos fluxos. Ou seja, quando consideramos a configuração de distribuição dos

fluxos, origens e destinos quaisquer, o número de iterações é bem menor do que o número de iterações quando consideramos a configuração origens quaisquer e mesmo destino (veja Figura 4.3). O fato disso ocorrer é que quando as distribuições de fluxos têm origens e destinos quaisquer ocorrem o espalhamento dos fluxos pela rede. Assim, temos poucos fluxos sobre cada aresta, implicando em pequenas iterações e, para origens quaisquer e mesmo destino, há maior concentrações de fluxos nas arestas, assim acarretando em mais iterações.

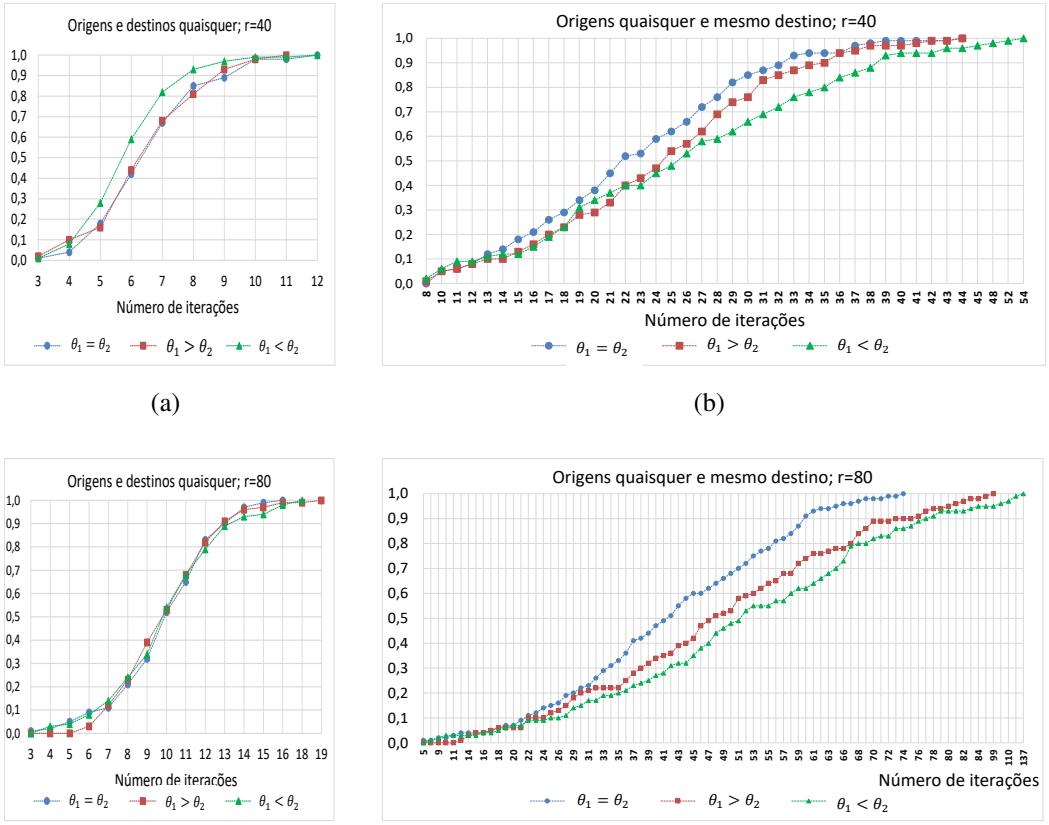
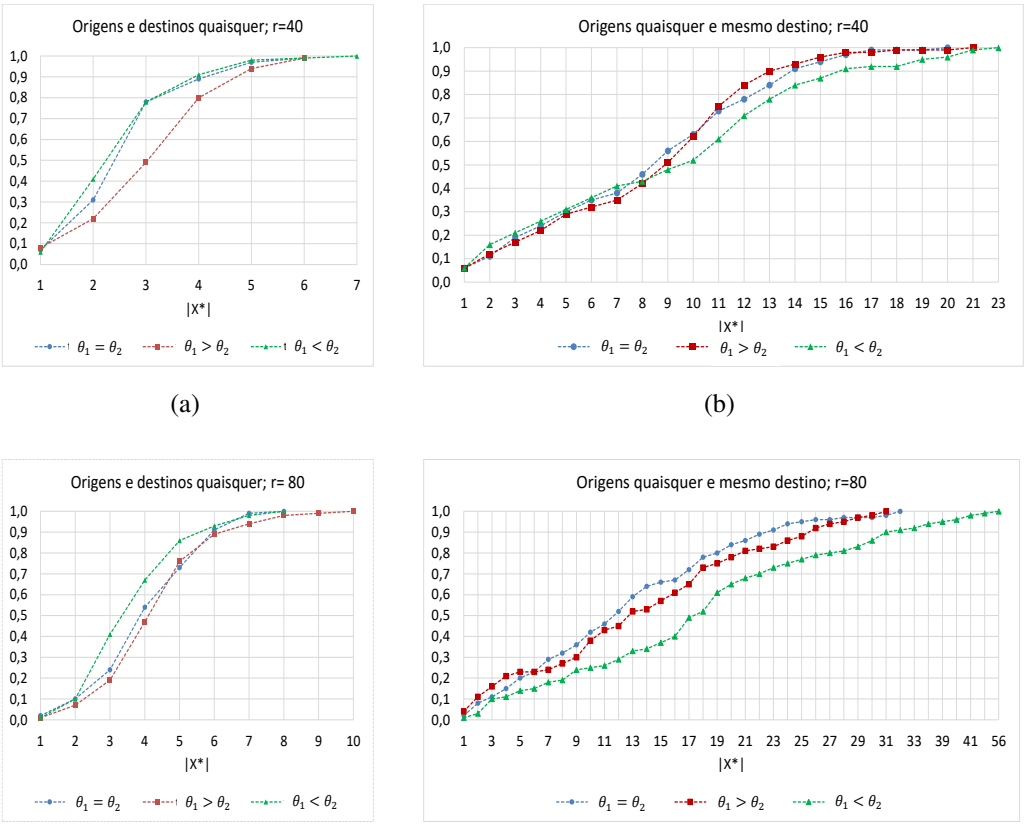


Figura 4.3: A função distribuição cumulativa do número de iterações.

Logo, em cada instância, é gerado poucas soluções Pareto-ótimas, para o primeiro tipo de distribuições de fluxos, e muito mais soluções Pareto-ótimas, para distribuições dos fluxos, origens quaisquer e mesmo destino (veja Figura 4.4).

Quanto ao tempo de execução do nosso algoritmo, podemos observar que, em cada instância, quanto maior o número de fluxos a serem roteados maior é esse tempo e, também, o tempo tem relação com a distribuição desses fluxos, ou seja, o tempo para configurações de origens quaisquer e mesmo destino é bem mais alto do que as que têm origens e destinos quaisquer, pelo falo de ter mais iterações na segunda configuração da distribuição dos fluxos (Figura 4.5).

Além disso, em distribuições de fluxos com origens quaisquer e mesmo destino, quando $\theta_1 > \theta_2$ pode ocorrer tempos bem mais altos do que para $\theta_1 < \theta_2$ e $\theta_1 = \theta_2$ (para $r = 40$ menos de 5% das instâncias e para $r = 80$ menos de 15% das instâncias, veja Figuras 4.5(b) e 4.5(d)). Esse fato ocorre, pois apesar de estarmos minimizando o gargalo levando em conta a qualidade da aresta e os pesos dos fluxos na aresta, quando consideramos



(a) (b)

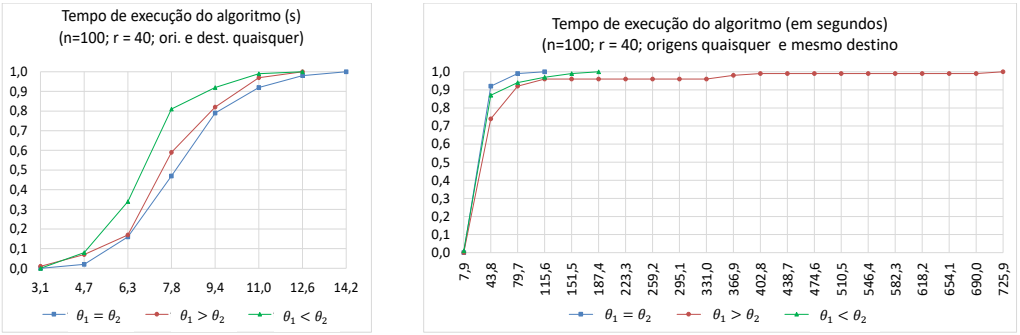
(c) (d)

Figura 4.4: A função distribuição cumulativa da cardinalidade do conjunto mínimo completo de soluções Pareto-ótimas.

$\theta_1 > \theta_2$, na verdade estamos dando maior grau de importância em minimizar o total dos pesos dos fluxos nas arestas, assim pode acarretar em mais combinações de resultados para soluções geradas. Assim, tratando, muitas vezes, em apenas permutar fluxos e, não necessariamente diminuir a quantidade de fluxos na(s) aresta(s) do gargalo.

As Figuras 4.6 e 4.7 mostram a função distribuição cumulativa do decréscimo do gargalo e do aumento do total de saltos, em porcentagens, da última solução Pareto-ótima em relação a primeira solução Pareto-ótima geradas pelo nosso algoritmo, respectivamente. Na maioria dos casos analisados o decréscimo máximo do gargalo ocorre quando $\theta_1 < \theta_2$, independente das configurações de tráfegos dos fluxos (veja Figuras 4.6(b), 4.6(d), 4.6(a) e 4.6(c)).

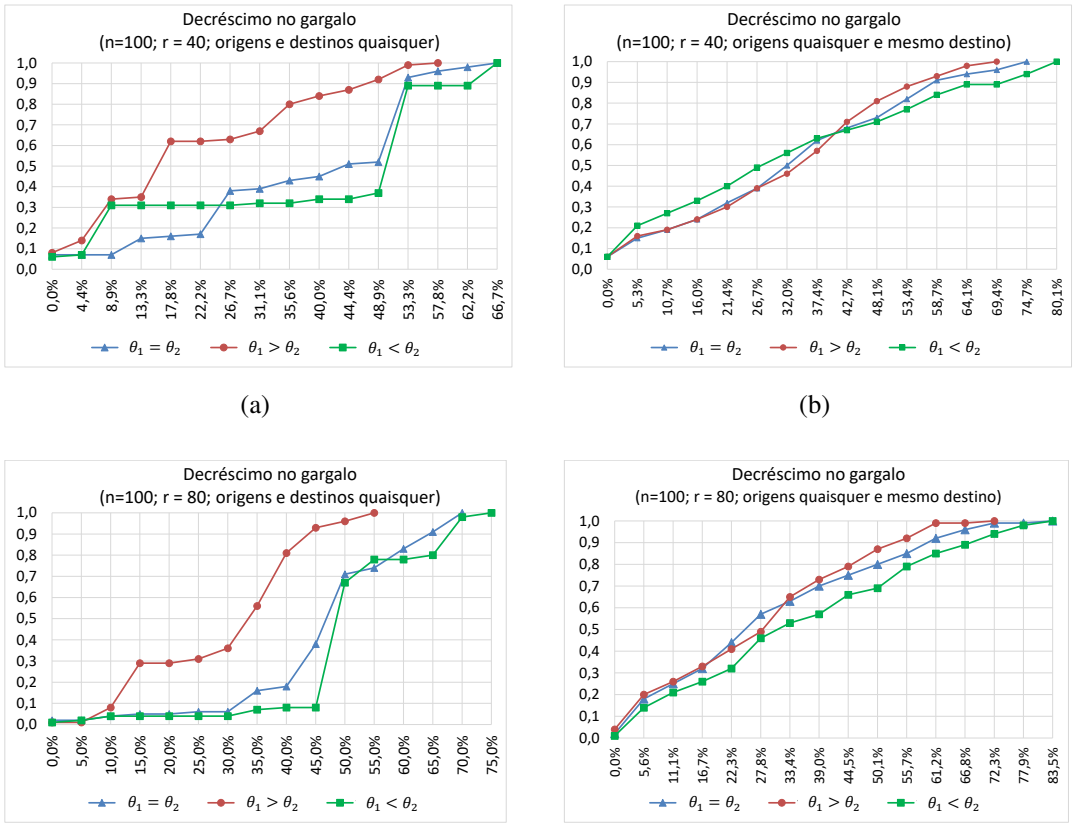
O aumento do total de saltos não ultrapassa de $42,7\%$. Como é de se esperar, independente das configurações para a distribuição dos fluxos, quanto maior o número de fluxos a serem roteados menor foi o aumento no total de saltos da última solução Pareto-ótima em relação a primeira solução Pareto-ótima. Ou seja, com o aumento na quantidade de fluxos, diminui a amplitude entre os valores de z_2, de uma iteração para outra (veja Figuras 4.7(a), 4.7(b), 4.7(c) e 4.7(d)).



(a) (b)

(c) (d)

Figura 4.5: A função distribuição cumulativa do tempo de execução.

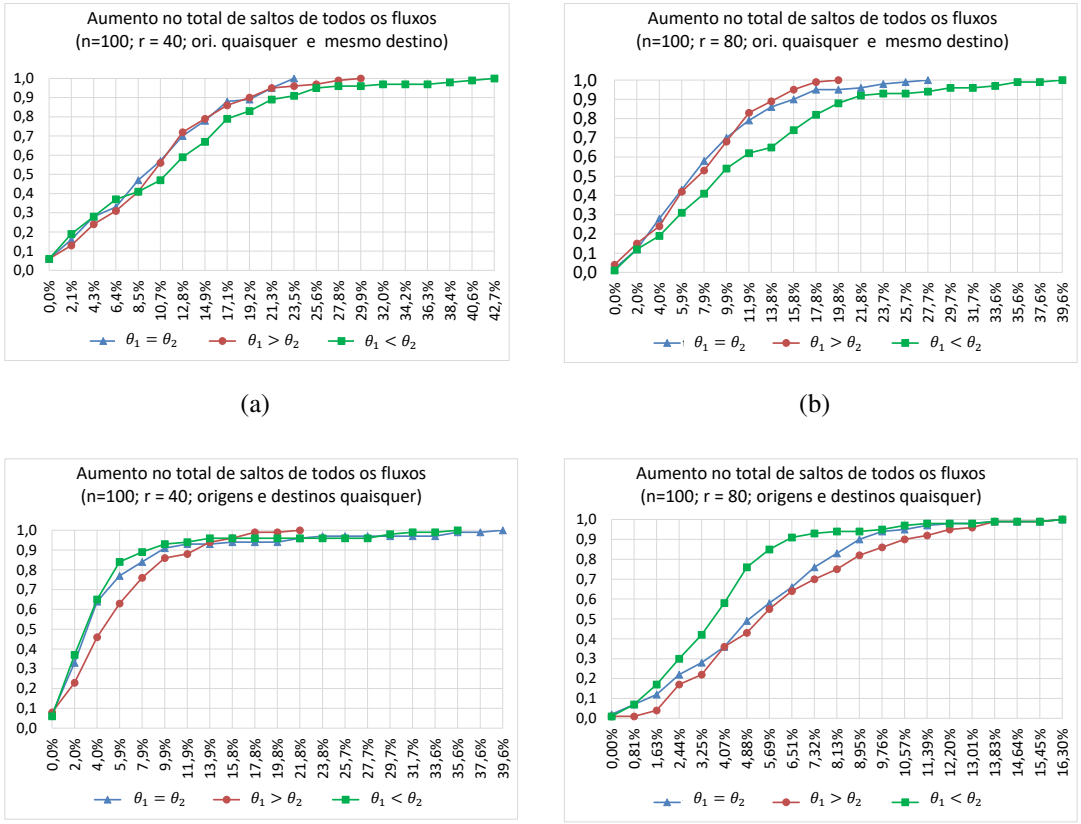


(a) (b)

(c) (d)

Figura 4.6: A função distribuição cumulativa das porcentagens de decréscimo do gargalo da última solução Pareto-ótima em relação a primeira solução Pareto-ótima.

Esse decréscimo da amplitude do total de saltos, para quantidade maiores de fluxos, acontece por termos menos possibilidades para o roteamento, isto é, na primeira iteração já existem fluxos atravessando uma quantidade muito superior de arestas do que em roteamento com menos fluxos. O melhor resultado para o aumento do total de saltos ocorre nas configurações de origens e destinos quaisquer com roteamento de 80 fluxos, não ultrapassando 16,3% de acréscimo no total de saltos (veja Figura 4.7(d)).



(a) (b)

(c) (d)

Figura 4.7: A função distribuição cumulativa das porcentagens do aumento do total de saltos de todos os fluxos da última solução Pareto-ótima em relação a primeira solução Pareto-ótima.

Neste modelo, $(\overline{P})$, nas soluções Pareto-ótimas geradas, observamos que o total de saltos de cada fluxo atinge um limite superior do que o total de saltos de cada fluxo das soluções Pareto-ótimas do modelo (P). Isso ocorre, pois em (P) o gargalo é dado pela quantidade máxima de fluxos que atravessam as arestas e aqui essa quantidade é multiplicada por valores de (w_{ij}^f). Daí, muitas vezes, para minimizar o gargalo, há um esticamento significante no total de saltos de cada fluxo comparado com minimizar o gargalo considerando apenas a quantidade de fluxos que passam pelas arestas.

CAPÍTULO 5

Um problema estocástico dois estágios de roteamento de fluxos

O trabalho apresentado neste capítulo deu origem ao artigo [Fernandes et al. 2019]. Nele apresentamos um modelo mono-objetivo de programação estocástica de dois estágios de roteamento de fluxos em redes que leva em conta as flutuações nas qualidades dos enlaces (arestas), ou seja, aqui as qualidades dos enlaces (arestas) são variáveis aleatórias discretas. O objetivo do problema é minimizar o gargalo da rede, roteando r fluxos, o qual tem como restrições o controle dos comprimentos dos caminhos de todos os fluxos (custo total de todos os fluxos). A restrição e função objetivo são acopladas através de um sobrepeso, cuja influência depende da probabilidade de ocorrer esse sobrepeso em determinado evento.

Para esse modelo, apresentamos, também, a análise da solução eficiente para as seguintes configurações: quando as variáveis aleatórias, qualidades dos enlaces, não há correlação entre os eventos e quando há correlação entre eles, considerando a distribuição normal, que representam as qualidades incertas dos enlaces. Para todas essas situações, examinamos a estabilidade da solução do modelo utilizando os testes *in-sample*, *out-of-sample* e *bias*.

5.1 O problema mono-objetivo determinístico

Considere um conjunto de fluxos F, com $|F| = r$, a ser roteado em um grafo orientado $G = (V, E)$, onde V representa o conjunto de nós e E o conjunto de arestas, com $|V| = n$ e $|E| = m$. Cada fluxo $f \in F$ deve ser roteado por um único caminho em G, indo da sua origem $s_f \in V$ ao seu destino $d_f \in V$. Para cada aresta $(i,j) \in E$, indo de $i \in V$ para $j \in V$, e fluxo $f \in F$, definimos uma variável binária x_{ij}^f, que irá indicar se o fluxo f vai passar (ou não) pela aresta (i,j).

Seja $q_{ij} \in \mathbb{Z}_{\geq 0}$ um valor associado a cada aresta $(i,j) \in E$, correspondendo a qualidade dessa aresta, com $q_{ij} \in \{q_1, q_2, \cdots, q_\alpha\}$ e $q_1 < q_2 < \cdots < q_\alpha$. Seja, ainda, c_{ij}^f o custo, não negativo, para o fluxo f utilizar a aresta $(i,j) \in E$.

Assim, dado um conjunto de fluxos F, a topologia do grafo G, o custo total mínimo do roteamento de cada fluxo $f \in F$, denotado por Γ^f, e o gargalo da rede, definido como a carga da(s) aresta(s) mais pesada(s), dado por $\max\limits_{(i,j)\in E}\left\{\sum\limits_{f\in F} q_{ij}.x_{ij}^f\right\}$, formulamos um modelo mono-objetivo de programação 0-1, para o problema de rotear os r fluxos, minimizando o gargalo da rede, da seguinte forma:

$$(P^*) \quad \textit{minimizar} \left\{ \max_{(i,j) \in E} \left\{ \sum_{f \in F} q_{ij}.x_{ij}^f \right\} \right\} \tag{5-1}$$

sujeito a:

$$\sum_{f \in F} \sum_{(i,j) \in E} c_{ij}^f.x_{ij}^f \leq \epsilon . \sum_{f \in F} \Gamma^f \tag{5-2}$$

$$\sum_{(i,j) \in E} x_{ij}^f - \sum_{(j,i) \in E} x_{ji}^f = \begin{cases} 1, & \text{se } i = s^f \\ 0, & \forall i \in V - \left\{ d^f, s^f \right\} \\ -1, & \text{se } i = d^f \end{cases} \qquad \forall f \in F \tag{5-3}$$

$$x_{ij}^f \in \{0,1\}, \ \forall (i,j) \in E \text{ e } \forall f \in F \tag{5-4}$$

No tráfego de rede de comunicação de dados (por exemplo, dominado por fluxos TCP), os fluxos são afetados pelo comprimento do caminho. Quanto mais longo for o caminho de um fluxo, maior será o tempo (médio) de retorno das confirmações e, portanto, menor será sua vazão média. Por esse motivo, controlamos na restrição (5-2) o comprimento total dos caminhos (custo total de todos os fluxos), onde $\epsilon \geq 1$ é um fator de esticamento dado. Assim, essa restrição garante que o custo total de todos os fluxos não seja maior do que o limite dado pelo parâmetro $\epsilon \sum_{f \in F} \Gamma^f$. Observe que se $\epsilon = 1$ então todos os fluxos são roteados pelos caminhos de custo total mínimo de todos os fluxos.

As restrições (5-3) garantem que cada fluxo $f \in F$ saia da sua origem s^f e chega ao seu destino d^f, passando por exatamente um caminho.

5.1.1 Análise de sensibilidade

A análise de sensibilidade consiste em investigar a estabilidade da solução ótima em vista de possíveis variações nos parâmetros do modelo. O objetivo desta seção é mostrar, através de uma análise de sensibilidade, se o modelo (P^*) é eficiente para resolver problemas de roteamento de fluxos em redes sem fio, quando há flutuações nos valores das qualidades dos enlaces.

Assim, apresentamos uma análise da solução ótima do problema (P^*), considerando uma rede aleatória com 10 nós e a topologia é gerada utilizando o modelo de Barabási-Albert [Barabási e Albert 1999] (veja Figura 5.1). Inicialmente, assumimos os valores das qualidades dos enlaces $q_{ij} \in \{10, 40, 70, 90\}$ e $c_{ij}^f = 1, \forall f \in F$ e $\forall (i,j) \in E$. Consideramos, ainda, que as flutuações das qualidades de cada enlace podem ser representadas por uma distribuição normal, onde os valores médios das qualidades dos enlaces pertencem a $\{10, 40, 70, 90\}$ e o desvio padrão são todos iguais a 20, exceto para $q_1 = 10$, no qual tem um desvio padrão igual a 10. Além disso, os fluxos são distribuídos, aleatoriamente, de acordo com as duas configurações: (1) origens e destinos múltiplos, e (2) origens múltiplas e um único destino.

Na análise das soluções ótimas, variando os valores das qualidades dos enlaces, analisamos três situações: (*i*) o valor da qualidade de uma única aresta escolhida aleatoriamente, com valor de qualidades q, assumindo todos os valores inteiros

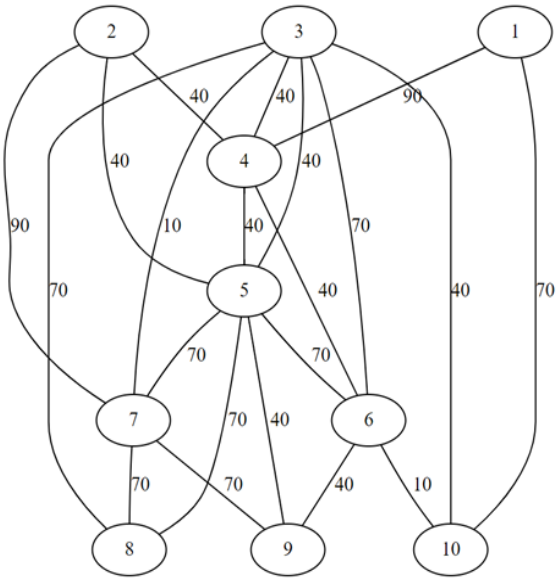


Figura 5.1: Grafo aleatório.

pertencentes ao intervalo $[q-20, q+20]$, (ii) o valor da qualidade de duas arestas escolhidas, aleatoriamente, com valores de qualidades q, assumindo todos os valores inteiros pertencentes ao intervalo $[q-20, q+20]$, (iii) todas as arestas, com valores de qualidades q, assumindo todos os valores inteiros pertencentes ao intervalo $[q-20, q+20]$ e, os demais enlaces não sofrem alterações nas qualidades.

Resolvemos o modelo (P^*) para todas essas variações, assim, obtemos o vetor de solução ótima $\bar{x}_{q+i}$ e o valor ótimo $F(\bar{x}_{q+i})$ de (P^*), com $i \in \mathbb{Z}$ tal que $-20 \leq i \leq 20$.

Posteriormente, comparamos as rotas ótimas das variações das qualidades com as rotas ótimas quando o(s) enlace(s) de qualidade q, assumem: a pior qualidade, $q+20$, a qualidade mediana, q, e a melhor qualidade, $q-20$. Isso foi feito para todos os casos $\Big((i), (ii)$ e $(iii)\Big)$.

Quando variamos os enlaces com qualidade $q = 70$, nas duas configurações de distribuição dos fluxos, observamos que as rotas ótimas, $\bar{x}_{q-20}, \bar{x}_q$ e $\bar{x}_{q+20}$, são diferentes das demais rotas ótimas $\bar{x}_{q+i}$. Fizemos essa mesma análise variando os enlaces de qualidade $q = 40$. Para a configuração (1) de distribuição dos fluxos, quando comparamos todas as soluções ótimas $\bar{x}_{q+i}$ com as três soluções $\bar{x}_{q-20}, \bar{x}_q$ e $\bar{x}_{q+20}$, apenas, em um único caso ocorreu de ter a mesma rota ótima ($\bar{x}_{q+20} = \bar{x}_{q+16}$). E, considerando a configuração (2) de distribuição dos fluxos, observamos que, no caso (i), variando uma única aresta, apenas em uma única situação as rotas são iguais ($\bar{x}_{q-20} = \bar{x}_q$); no caso (ii), variando duas arestas, ocorreram apenas duas situações em que as rotas são iguais ($\bar{x}_q = \bar{x}_{q+3}$ e $\bar{x}_{q+20} = \bar{x}_{q+19}$); e, considerando o caso (iii), todas as rotas são diferentes.

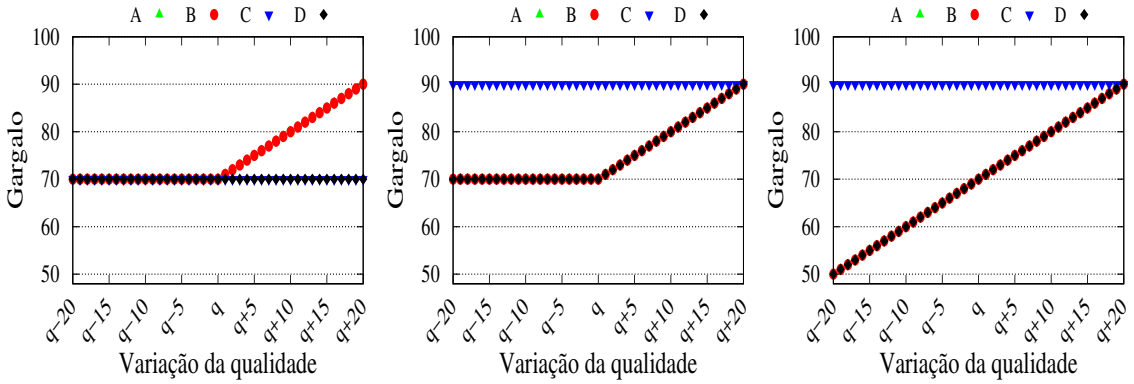
Independente da configuração de distribuição dos fluxos, observamos que variando os valores das qualidades das arestas, nesses cenários, obtemos $\bar{x}_q \neq \bar{x}_{q+i}$, com $i \neq 0$, na maioria dos casos. Portanto, a variação no valor das qualidades dos enlaces mostra que, novas rotas de roteamento de fluxos são necessárias, ou seja, o modelo é muito sensível às variações das entradas (valores das qualidades). Mas, reprocessar as rotas, por exemplo, em redes sem fio com múltiplos saltos, sempre que a condição do canal se altera, provoca instabilidade das rotas, impactando as aplicações que utilizam

essa rede.

Outra análise que consideramos foi observar se essas três soluções ($\bar{x}_{q-20}$, $\bar{x}_q$ e $\bar{x}_{q+20}$) são equivalentes para as demais variações de q. Ou seja, usamos essas rotas para fazermos os roteamentos dos fluxos na rede quando os enlaces de qualidade q tenham sofrido oscilações e calculamos o gargalo da rede para cada uma dessas oscilações, que denotamos por $F_{q+i}(\bar{x}_{\bar{q}})$, onde $\bar{q} \in \{q-20, q, q+20\}$ e $i \in \mathbb{Z}$ tal que $-20 \leq i \leq 20$. Portanto, o objetivo é comparar o gargalo $F_{q+i}(\bar{x}_{\bar{q}})$ com o valor ótimo $F(\bar{x}_{q+i})$.

As Figuras 5.2 e 5.3 mostram esses resultados para as configurações de distribuição dos fluxos com origens e destinos múltiplos e origens múltiplas e um único destino, respectivamente. Na legenda, onde usamos a notação A, B, C e D, lê-se $F_{q+i}(\bar{x}_{q-20})$, $F_{q+i}(\bar{x}_q)$, $F_{q+i}(\bar{x}_{q+20})$ e $F(\bar{x}_{q+i})$, respectivamente.

Quando consideramos a configuração de distribuição dos fluxos, origens e destinos quaisquer, e utilizamos a rota ótima $\bar{x}_{q-20}$, no caso (i), em 50% das variações da qualidade $q+i$, o valor $F_{q+i}(\bar{x}_q)$ é pior do que o valor ótimo $F(\bar{x}_{q+i})$ (veja a Figura 5.2(a)). Nos casos (ii) e (iii), temos que em 99% das variações da qualidade $q+i$, o valor $F_{q+i}(\bar{x}_{q+20})$ é pior do que o valor $F(\bar{x}_{q+i})$ (veja Figura 5.2(b) e 5.2(c)).

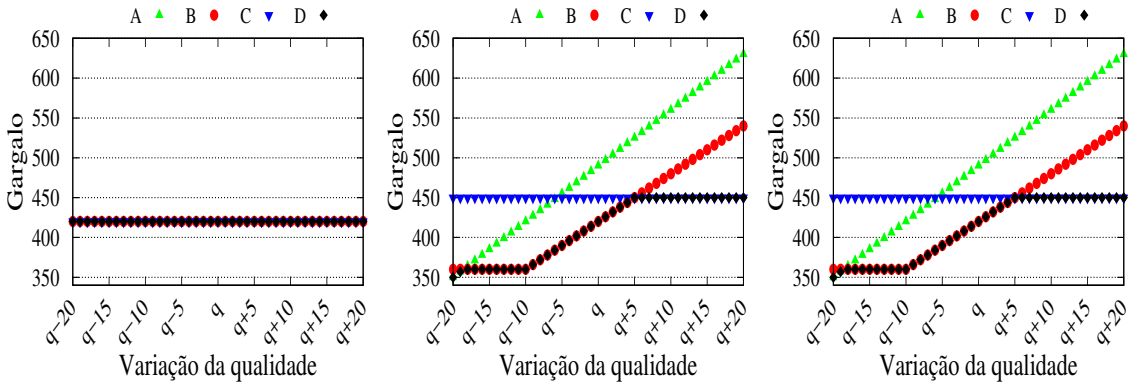


(a) *Variando uma única aresta.* (b) *Variando duas arestas.* (c) *Variando todas as arestas.*

Figura 5.2: Análise do gargalo de (P^*) para variação das qualidades $q = 70$ quando os fluxos são distribuídos pela configuração (1) de distribuição de fluxos.

Para a configuração de distribuição dos fluxos, origens quaisquer e único destino, quando utilizamos a rota ótima $\bar{x}_{q-20}$, nos casos (ii) e (iii), em 97,5% das variações da qualidade $q+i$, o valor $F_{q+i}(\bar{x}_{q-20})$ é pior do que o valor ótimo $F(\bar{x}_{q+i})$ (veja as Figuras 5.3(b) e 5.3(c)). Usando a rota ótima $\bar{x}_q$, nos casos (ii) e (iii), temos que em 37,5% das variações da qualidade $q+i$, o valor $F_{q+i}(\bar{x}_q)$ é pior do que o valor $F(\bar{x}_{q+i})$ (veja as Figuras 5.3(b) e 5.3(c)). E, por último, quando consideramos a rota ótima $\bar{x}_{q+20}$, nos casos (ii) e (iii), 62,5% das variações da qualidade, $F_{q+i}(\bar{x}_{q+20})$ é pior do que o valor ótimo $F(\bar{x}_{q+i})$.

Portanto, neste contexto de roteamento de fluxos em redes sem fio, a otimização estocástica nos permite capturar alguns aspectos que a determinística não é capaz, ou seja, ela é preparada para lidar com as flutuações dos valores das qualidades dos enlaces, independentemente da quantidade de enlaces que sofrem flutuações em suas qualidades. Assim, vimos a necessidade de trabalharmos com um modelo estocástico para problema de roteamento de fluxos em redes sem fio, onde os valores das qualidades dos enlaces são incertos, ou seja, eles são variáveis aleatórias.



(a) *Variando uma única aresta.* (b) *Variando duas arestas.* (c) *Variando todas as arestas.*

Figura 5.3: Análise do gargalo de (P^*) para variação das qualidades $q = 70$ quando os fluxos são distribuídos pela configuração (2) de distribuição de fluxos.

5.2 O problema estocástico

Como mostrado na seção anterior, o modelo determinístico (P^*) não é eficiente para resolver problemas roteamento de fluxos em redes sem fio, onde as qualidades dos enlaces flututam regularmente em escala de tempo muito pequena. Para lidar com essas deficiências da abordagem determinística, introduzimos, nesta seção, um modelo de programação estocástica de dois estágios.

5.2.1 O problema (P^τ)

Em problemas estocásticos de dois estágios, é necessário ter uma distribuição de probabilidade discreta de cardinalidade limitada que representa a aproximação do comportamento das variáveis aleatórias. As saídas da discretização são chamadas de *cenários*, e toda a distribuição, conjunto de cenários, é denominado de *árvore de cenários*. Essas árvores de cenários são estruturas de dados básicas para esses tipos de problemas, representando as discretizações dos processos estocásticos, portanto, uma aproximação dos fenômenos reais [Høyland, Kaut e Wallace 2003, Kaut e Wallace 2003, Pflug e Pichler 2015].

Agora, apresentamos o nosso modelo estocástico de roteamento de fluxos em redes sem fio, minimizando o gargalo da rede e, na restrição, controlamos o custo total de todos os fluxos (comprimento total dos caminhos). Seja τ a árvore de cenários descrevendo a incerteza do problema e os cenários desse conjunto representado por ξ. Considere $q_{ij}^{\xi} \in \mathbb{Z}_{\geq 0}$ uma variável aleatória discreta associada a cada aresta (enlace), $(i,j) \in E$, correspondendo ao valor da qualidade dessa aresta no cenário $\xi, \in \tau$ e seja c_{ij}^{f}, o custo, não negativo, para o fluxo f atravessar a aresta $(i,j) \in E$.

Assim, dado um conjunto de fluxos F, a topologia do grafo G, o custo total mínimo do roteamento de cada fluxo $f \in F$, denotado por Γ^f, e o gargalo da rede, definido como a carga da(s) aresta(s) mais pesada(s), dado por $\max\limits_{\substack{(i,j)\in E \\ \xi\in\tau}} \left\{ \sum\limits_{f \in F} q_{ij}^{\xi}.x_{ij}^{f} \right\}$, formulamos um modelo estocástico de dois estágios de programação não-linear, para o problema de rotear os r fluxos, minimizando o gargalo da rede, da seguinte forma:

$$(P^{\tau}) \quad minimizar \left\{ \max_{\substack{(i,j)\in E \\ \xi\in\tau}} \left\{ \sum_{f\in F} q_{ij}^{\xi}.x_{ij}^{f} \right\} + \hat{c} \sum_{\xi\in\tau} p^{\xi}.w^{\xi} \right\} \tag{5-5}$$

sujeito a:

$$\sum_{f\in F}\sum_{(i,j)\in E} c_{ij}^{f}.x_{ij}^{f} - w^{\xi} \leq \epsilon . \sum_{f\in F} \Gamma^{f}, \quad \forall \xi \in \tau \tag{5-6}$$

$$\sum_{(i,j)\in E} x_{ij}^{f} - \sum_{(j,i)\in E} x_{ji}^{f} = \begin{cases} 1, & \text{se } i = s^{f} \\ 0, & \forall i \in V - \left\{ d^{f}, s^{f} \right\} \\ -1, & \text{se } i = d^{f} \end{cases} \quad \forall f \in F$$

$$x_{ij}^{f} \in \{0,1\}, \ \forall (i,j) \in E \text{ e } \forall f \in F$$

$$w^{\xi} \in \mathbb{R}_{+}, \ \forall \xi \in \tau, \tag{5-7}$$

onde $\hat{c}$ é a penalidade unitária para o sobrepeso do custo total no cenário ξ, denotado por w^{ξ}. Além disso, p^{ξ} é a probabilidade desse sobrepeso ocorrer no cenário ξ e $\sum_{\xi\in S} p^{\xi}.w^{\xi}$ representa a esperança dos valores de sobrepeso do problema.

Em (5-6), $\epsilon \geq 1$ é um fator de esticamento dado. As restrições (5-6) controlam o comprimento total dos caminhos dos fluxos. Observe que essas restrições e a função objetivo são acopladas através do sobrepeso w^{ξ}, cuja influência depende da probabilidade, p^{ξ}, do sobrepeso ocorrer no cenário ξ. Assim, é possível exceder essas restrições se a penalidade total for inferior ao benefício obtido na redução do gargalo. Ou seja, essas restrições garantem que o comprimento total (custo total) de todos os fluxos, em cada cenário ξ, não sejam maiores do que o limite dado pelo parâmetro $\epsilon \sum_{f\in F} \Gamma^{f}$, mas se ocorrer um sobrepeso w^{ξ}, para cada cenário ξ, o modelo sofrerá uma penalidade de custo $\hat{c}$ para cada unidade de sobrepeso.

As demais restrições garantem que cada fluxo $f \in F$ saia da sua origem, s^{f}, e chegue ao seu destino, d^{f}, passando por exatamente um caminho.

A função objetivo busca minimizar a carga da(s) aresta(s) mais pesada(s) mas garantindo que a penalidade no sobrepeso do custo total de cada cenário seja o mínimo possível, ou seja, busca uma solução (roteamento dos r fluxos em G) em que a aresta mais congestionado da rede seja o menos congestionado possível, minimizando, também, o sobrepeso no custo total.

5.3 Geração de cenários

Neste trabalho, para geração das árvores de cenários, utilizamos uma técnica baseada em simulação de Monte Carlo denominada *aproximação por média amostral* (sample average aproximation - SAA), primeiramente proposto por [Shapiro e Mello 1998].

Em [Shapiro 2003], ele comenta que as vantagens da SAA são a facilidade de implementação numérica, podendo usar o software já existentes, boas propriedades de convergência, inferência estatística bem desenvolvida: validação e análise de erros, regras de parada, dentre outras. Para geração das árvores de cenários utilizando esse método é necessário, como dados de entrada, a matriz correlação das variáveis aleatórias, os valores máximo e mínimo de cada variável aleatória e a distribuição que representa essas variáveis.

Para árvore de referência, foi utilizado método *moment-matching* [Høyland, Kaut e Wallace 2003]. Consiste de uma aproximação eficiente e necessita, como dados de entrada, dos primeiros quatro momentos centrais especificados das distribuições marginais (média, desvio padrão, assimetria e curtose) e a matriz de covariância das variáveis aleatórias e, é muito eficaz para árvores de tamanhos grandes. A utilização de técnicas diferentes para geração de árvores de cenários e de referência é para garantir a confiabilidade dos resultados.

Considere que para cada um dos p tamanhos distintos, $\zeta = [\zeta_1, \zeta_2, \cdots, \zeta_p]$, de árvores de cenários geradas, k árvores foram geradas. Denotamos essas árvores por $\tau_i^{(j)}$, com $i \in \{1, \cdots, p\}$ e $j \in \{1, \cdots, k\}$. Assim, o número total de árvores geradas são $\kappa = k.p$ árvores. Denotamos por $\tilde{\tau}$ a árvore de referência.

No problema estocástico (P^τ) podemos observar que tem $\eta_i = m.r + \zeta_i$ variáveis de decisão, com $i \in \{1, \cdots, p\}$, ou seja, o número de variáveis do problema varia com o tamanho da árvore de cenários considerada.

5.3.1 Procedimento para análise de estabilidade

Como comentamos, anteriormente, para reduzir o erro das árvores de cenários geradas, devemos fazer uma análise da qualidade e do tamanho dessas árvores, para isso utilizamos os *testes de estabilidade*. Vamos aplicar os testes *in-sample*, *out-of-sample* e o *bias*.

Considere $x_i^{(j)}$ a solução ótima do problema (P^τ) para a árvore de cenário $\tau_i^{(j)}$, x^* a solução ótima do problema (P^τ) para a árvore de referência, $F(x_i^{(j)}; \tau_i^{(j)})$ o valor objetivo para cada árvore de cenário, $F(x^*; \tilde{\tau})$ o valor objetivo para árvore de referência e $F(x_i^{(j)}; \tilde{\tau})$ o valor verdadeiro então:

1. Se $F(x_i^{(j)}; \tau_i^{(j)}) \approx F(x_i^{(j')}; \tau_i^{(j')})$, para todo $j, j' \in \{1, \cdots, k\}$, temos estabilidade *in-sample* para as árvores com ζ_i cenários.
2. Se $F(x_i^{(j)}; \tilde{\tau}) \approx F(x_i^{(j')}; \tilde{\tau})$, para todo $j, j' \in \{1, \cdots, k\}$, temos estabilidade *out-of-sample* para as árvores com ζ_i cenários.
3. Se $F(x^*; \tilde{\tau}) \approx F(x_i^{(j)}; \tau_i^{(j)})$, com $i \in \{1, \cdots, p\}$, para todo $j \in \{1, \cdots, k\}$, não temos *bias* para as árvores com ζ_i cenários.

Com base em [Kaut e Wallace 2003] o modelo é resolvido sobre cada uma das árvores de cenários geradas. A média e o desvio padrão dos valores objetivo do problema estocástico (P^τ) para árvores de cenários, $\xi_i^{(j)}$, de tamanhos diferentes, são determinados. Com esses valores, fazemos a verificação de estabilidade *in-sample*, *out-of-sample* e *bias*. A seguir apresentamos, detalhadamente, o procedimento para essas análises:

1. Achar a solução ótima, $x_i^{(j)}$, do problema estocástico para cada árvore de cenários $\tau_i^{(j)}$.

2. Achar a solução ótima, x^*, do problema estocástico para árvore de referência $\tilde{\tau}$.
3. Determinar o valor objetivo, $F(x_i^{(j)};\tau_i^{(j)})$, do problema estocástico para cada árvore de cenários $\tau_i^{(j)}$.
4. Determinar o valor objetivo, $F(x^*;\tilde{\tau})$, do problema estocástico para árvore de referência $\tilde{\tau}$.
5. Determinar o valor objetivo verdadeiro, $F(x_i^{(j)};\tilde{\tau})$, do problema estocástico sobre a árvore de referência $\tilde{\tau}$, para cada solução ótima $x_i^{(j)}$ do problema sobre cada árvore de cenários $\tau_i^{(j)}$.
6. Determinar os p valores objetivo médio dos $F\left(x_i^{(j)};\tau_i^{(j)}\right)$, denotados por $\overline{\Gamma}_i$, para todo $i \in \{1,\cdots,p\}$.
7. Determinar os p valores objetivo médio dos $F\left(x_i^{(j)};\tilde{\tau}\right)$, denotados por $\overline{\Upsilon}_i$, para todo $i \in \{1,\cdots,p\}$.
8. Determinar os p desvios padrões dos $F\left(x_i^{(j)};\tau_i^{(j)}\right)$, denotados por σ_{Γ_i}, para todo $i \in \{1,\cdots,p\}$.
9. Determinar os p desvios padrões dos $F\left(x_i^{(j)};\tilde{\tau}\right)$, denotados por σ_{Υ_i}, para todo $i \in \{1,\cdots,p\}$.

5.4 Resultados computacionais

Os resultados da simulação tem como objetivo determinar soluções eficientes para o problema (P^τ), definido sobre uma rede. Os métodos de geração de cenários para otimização estocástica foram implementados em Python v2.7. Para resolver o problema (P^τ), utilizamos o pacote de otimização CPLEX® v12.8 em C++. Todos os testes foram executados sobre um Ubuntu 18.04.1 LTS em um equipamento Intel® Xeon(R), CPU Silver 4114 de 2.20GHz, 10 vCPUs, 64 GB de RAM e 100 GB de disco (virtual).

A topologia aleatória foi gerada, aleatoriamente, pelo modelo Barabási–Albert (BA) [Barabási e Albert 1999]. Utilizamos o mesmo grafo, gerado aleatoriamente, para todos os cenários considerados. O fator de esticamento adotado foi $\epsilon = 1,5$.

A rede de fluxos foi distribuída de acordo com duas configurações: 1) múltiplas (quaisquer) origens aleatórias e múltiplos destinos aleatórios, e 2) múltiplas origens aleatórias e um único (mesmo) destino aleatório. Em cada configuração, avaliamos cenários, com r = 10 fluxos.

A abordagem estocástica, além de capturar as flutuações nas qualidades dos enlaces, permite representar a correlação entre os eventos que influenciam a qualidade dos enlaces. Assim, estamos interessados em determinar a solução eficiente do modelo (P^τ) nos seguintes casos: (i) as qualidades dos enlaces são não-correlacionadas, (ii) as qualidades dos enlaces são correlacionadas do tipo 1 (Tabela 5.1), (iii) as qualidades dos enlaces são correlacionadas do tipo 2 (Tabela 5.1). Em todos os casos, as qualidades aleatórias são representadas por uma distribuição normal.

As informações para gerar as árvores de cenários pelo método SAA foram as seguintes: a distribuição utilizada foi a normal, a dimensão dos cenários foi $m/2$ e os elementos da matriz de correlação foram $\rho_{ij} = 1$, se $i = j$ e $\rho_{ij} = 0$, se $i \neq j$, quando não há correlação entre as qualidades dos enlaces e, quando há correlação nas qualidades dos enlaces utilizamos $\rho_{ij} = 1$, se $i = j$ e $\rho_{ij} = \rho_{ji} = p_{ij}$, se $i \neq j$, onde p_{ij} é um valor real do

Tabela 5.1: Distribuição de frequência cumulativa dos dois tipos de correlação considerados.

ρ_{ij}	0	0,1	0,3	0,5	0,7	0,9	1,0
Tipo 1	52%	64%	80%	92%	95%	95%	100%
Tipo 2	0%	10%	39%	77%	91%	94%	100%

intervalo $[0,1]$, gerado aleatoriamente, como apresentado na Tabela 5.1. Além disso, a matrix correlação ρ é uma symmetric positive semi-definite, ou seja, ρ tem que satisfazer a condição de Decomposição de Cholesky [Higham 1990]. Na prática, caso a matriz de correlação não satisfaça essa condição, já existem algoritmos que acham uma correlação aproximada da correlação específica [Higham 2000, Lurie e Goldberg 1998].

Para gerar a árvore de referência utilizamos: a distribuição normal, a dimensão de cada cenário, ξ_ℓ, foi $m/2$, os elementos da matriz de covariância foram calculados utilizando as matrizes de correlação definida anteriormente.

Consideramos $p = 4$, onde $\zeta = [\,10\ 20\ 40\ 60]$. Além disso, geramos $k = 25$ árvores de cenários de tamanho, $\zeta_i \in \zeta$, com $i \in \{1,2,3,4\}$: $\tau_i^{(j)} = \left(q_u^{\tau_i^{(j)}}\right)$, sendo que cada vetor $q_u^{\tau_i^{(j)}} = \begin{bmatrix} q_u^{\xi_1} \\ q_u^{\xi_2} \\ \vdots \\ q_u^{\xi_{\zeta_i}} \end{bmatrix}$ com $i \in \{1,2,3,4\}$, $j \in \{1,\cdots,25\}$ e $u \in \{1,\cdots,\frac{m}{2}\}$, onde definimos as qualidades como segue:

$$q_{ij}^{\xi_\ell} = \begin{cases} 10 & \text{se } 0 \le q_u^{\xi_\ell} < 20 \\ 40 & \text{se } 20 \le q_u^{\xi_\ell} < 60 \\ 70 & \text{se } 60 \le q_u^{\xi_\ell} \le 80 \\ 90 & \text{se } 80 \le q_u^{\xi_\ell} \le 100 \end{cases},$$

Portanto, temos as árvores resultantes $\tilde{\tau}_i^{(j)} = \left(q_{ij}^{\xi_\ell}\right)$. Consideramos, ainda, que $q_{ij}^{\xi_\ell} = q_{ji}^{\xi_\ell}$, $\forall (i,j) \in E$ e $\forall \xi_\ell \in \tau$.

A Tabela 5.2 mostra os valores $F(x^*, \tilde{\tau})$ do modelo (P^τ) para cada caso considerado. Já, as Tabelas 5.4 e 5.3, apresentam a média e o desvio padrão dos valores objetivo das soluções eficientes e dos valores verdadeiros da função objetivo, para os diferentes tamanhos de árvores de cenários.

Tabela 5.2: Valores, $F(x^*;\tilde{\tau})$, de (P^τ) para a árvore de referência, em cada caso.

	Tipos de Correlação		
	Sem correl.	Correl. Tipo 1	Correl. Tipo 2
Múltiplas origens e um único destino	421,00	451,00	451,00
Múltiplas origens e múltiplos destinos	71,00	91,00	91,00

No caso das qualidades de enlaces não-correlacionadas, os resultados sugerem que não devemos usar árvores menores que 20 cenários. E, para qualidades correlacionadas, sugerem que devemos usar árvores maiores ou igual a 20, para segunda configuração de fluxos, e, árvores de tamanho maiores ou igual a 40, para primeira

Tabela 5.3: Testes de estabilidade para o modelo de otimização (P^{τ}). A tabela apresenta a média e o desvio padrão dos valores ótimos, para os tamanhos diferentes de árvores de cenários.

Origens múltiplas e destinos múltiplos						
Sem correlação						
Descrição do teste			# de cenários			
Tipo do teste	Função Objetivo	Valor	10	20	40	60
in-sample	$F(x_i^{(j)};\tau_i^{(j)})$	Média ($\overline{\Gamma}_i$)	72,00	71,00	71,00	71,00
		Desvio (σ_{Γ_i})	0,00	0,00	0,00	0,00
out-of-sample	$F(x_i^{(j)};\tilde{\tau})$	Média ($\overline{\Upsilon}_i$)	71,00	71,00	71,00	71,00
		Desvio (σ_{Υ_i})	0,00	0,00	0,00	0,00
Correlação tipo 1						
Descrição do teste			# de cenários			
Tipo do teste	Função Objetivo	Valor	10	20	40	60
in-sample	$F(x_i^{(j)};\tau_i^{(j)})$	Média ($\overline{\Gamma}_i$)	92,00	91,00	91,00	91,00
		Desvio (σ_{Γ_i})	0,00	0,00	0,00	0,00
out-of-sample	$F(x_i^{(j)};\tilde{\tau})$	Média ($\overline{\Upsilon}_i$)	105,00	96,20	91,00	91,00
		Desvio (σ_{Υ_i})	22,9129	14,7535	0,00	0,00
Correlação tipo 2						
Descrição do teste			# de cenários			
Tipo do teste	Função Objetivo	Valor	10	20	40	60
in-sample	$F(x_i^{(j)};\tau_i^{(j)})$	Média ($\overline{\Gamma}_i$)	92,00	91,00	91,00	91,00
		Desvio (σ_{Γ_i})	0,00	0,00	0,00	0,00
out-of-sample	$F(x_i^{(j)};\tilde{\tau})$	Média ($\overline{\Upsilon}_i$)	101,00	94,20	91,00	91,00
		Desvio (σ_{Υ_i})	20,4124	11,4455	0,00	0,00

configuração de fluxos (veja Tabelas 5.4 e 5.3). Os resultados, ainda, mostram que o valor objetivo atinge um valor bem menor quando a configuração dos fluxos é de múltiplas origens e múltiplos destinos, existindo ou não correlação entre as qualidades dos enlaces. Isso ocorre, pois, nessa configuração dos fluxos, os fluxos ficam mais espalhados na rede, acarretando gargalos menores.

Além disso, quando há correlação entre os valores das qualidades dos enlaces, as estabilidades são atingidas nas árvores com os mesmos tamanhos e com os mesmos valores da função objetivo, quando as configurações do roteamento dos fluxos são iguais. Todavia, observando a Figura 5.5, temos que os comprimentos dos caminhos, na maioria das vezes, são diferentes para cada tipo de correlação, mostrando que as rotas tomadas pelos fluxos, na maioria das vezes, são diferentes para cada tipo de correlação. Portanto, podemos concluir que a correlação afeta nas decisões das rotas a serem tomadas pelos fluxos.

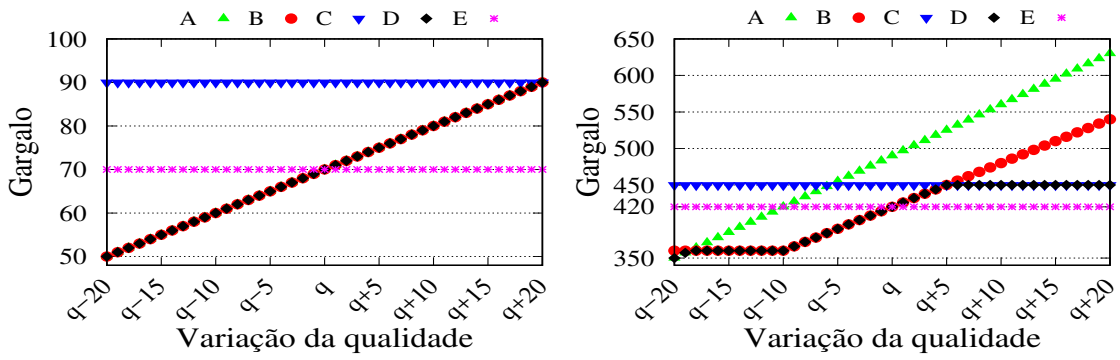
Podemos observar que quando as qualidades dos enlaces são não-correlacionadas, as estabilidades (*in-sample* e *out-of-sample*) ocorrem mais rápido do que quando há correlação. Além disso, quando não há correlação, o modelo, independente da configuração dos fluxos, consegue um valor objetivo eficiente bem melhor do que quando há correlação entre as qualidades, ou seja, o valor objetivo atinge um valor menor quando não há correlações entre as qualidades (veja Tabelas 5.4 e 5.3)

A Figura 5.4 mostra os valores dos gargalos do modelo (P^*), apresentados na

Tabela 5.4: Testes de estabilidade para o modelo de otimização (P^{τ}). A tabela apresenta a média e o desvio padrão dos valores ótimos, para os tamanhos diferentes de árvores de cenários.

Origens múltiplas e mesmo destino						
Sem correlação						
Descrição do teste			# de cenários			
Tipo do teste	Função Objetivo	Valor	10	20	40	60
in-sample	$F(x_i^{(j)};\tau_i^{(j)})$	Média ($\overline{\Gamma}_i$)	425,00	421,00	421,00	421,00
		Desvio (σ_{Γ_i})	0,00	0,00	0,00	0,00
out-of-sample	$F(x_i^{(j)};\tilde{\tau})$	Média ($\overline{\Upsilon}_i$)	421,00	421,00	421,00	421,00
		Desvio (σ_{Υ_i})	0,00	0,00	0,00	0,00
Correlação tipo 1						
Descrição do teste			# de cenários			
Tipo do teste	Função Objetivo	Valor	10	20	40	60
in-sample	$F(x_i^{(j)};\tau_i^{(j)})$	Média ($\overline{\Gamma}_i$)	455,00	451,00	451,00	451,00
		Desvio (σ_{Γ_i})	0,00	0,00	0,00	0,00
out-of-sample	$F(x_i^{(j)};\tilde{\tau})$	Média ($\overline{\Upsilon}_i$)	451,00	451,00	451,00	451,00
		Desvio (σ_{Υ_i})	0,00	0,00	0,00	0,00
Correlação tipo 2						
Descrição do teste			# de cenários			
Tipo do teste	Função Objetivo	Valor	10	20	40	60
in-sample	$F(x_i^{(j)};\tau_i^{(j)})$	Média ($\overline{\Gamma}_i$)	455,00	451,00	451,00	451,00
		Desvio (σ_{Γ_i})	0,00	0,00	0,00	0,00
out-of-sample	$F(x_i^{(j)};\tilde{\tau})$	Média ($\overline{\Upsilon}_i$)	451,00	451,00	451,00	451,00
		Desvio (σ_{Υ_i})	0,00	0,00	0,00	0,00

subseção 5.1.1 quando variamos todas as arestas de qualidade $q = 70$, e mostramos, também, o valor do gargalo de (P^{τ}) quando as qualidades dos enlaces não são correlacionadas, representado por uma reta paralela ao eixo das abscissas. Na legenda, onde usamos a notação A, B, C, D e E, lê-se $F_{q+i}(\bar{x}_{q-20})$, $F_{q+i}(\bar{x}_q)$, $F_{q+i}(\bar{x}_{q+20})$, $F(\bar{x}_{q+i})$ e o gargalo de (P^{τ}), respectivamente.



(a) *Múltiplas origens e múltiplos destinos* (b) *Múltiplas origens e único destino*

Figura 5.4: Comparação dos gargalos do modelo determinístico, variando todas as arestas de qualidade $q = 70$, com o gargalo do modelo estocástico.

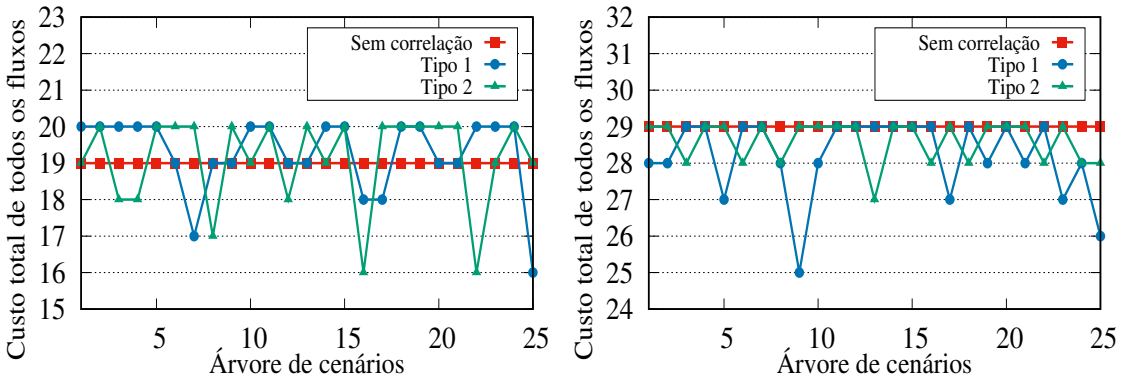
Como observado na Seção 5.1, do modelo determinístico, os valores do gargalo

$F(\bar{x}_{q+i})$ do modelo (P^*) oscilam muito, quando há uma variação regular nos valores das qualidades dos enlaces. Assim, não se pôde concluir qual seria a solução ótima que melhor representaria essas oscilações nas qualidades dos enlaces. O nosso objetivo, agora, é comparar esses gargalos com o gargalo do modelo estocástico (P^τ).

Para a configuração (1) de distribuição dos fluxos (origens e destinos múltiplos), observamos que o gargalo do modelo estocástico (P^τ) é igual, apenas, ao gargalo $F(\bar{x}_q)$ de (P^*), além disso, o gargalo $F(\bar{x}_{q+i})$ é melhor que o gargalo de (P^τ), quando os enlaces assumem valores de melhores qualidades, ou seja, quando $-20 \leq i \leq -1$. Para esse mesmo intervalo, o gargalo de (P^τ) é maior que os valores $F_{q+i}(\bar{x}_{q-20})$ e $F_{q+i}(\bar{x}_q)$. Nas demais variações da qualidade, quando $1 \leq i \leq 20$, o gargalo do modelo estocástico (P^τ) é melhor. Para o caso que utilizamos a rota ótima $\bar{x}_{q+20}$, os valores, $F_{q+i}(\bar{x}_{q+20})$, são todos piores do que o gargalo de (P^τ), como mostra a Figura 5.4(a).

Para configuração (2) de distribuição de fluxos (origens múltiplas e um único destino), os gargalos de (P^*), na maioria dos casos, assumem valores diferentes do gargalo do modelo estocástico (P^τ). Os gargalos $F(\bar{x}_{q+i})$ são menores que o gargalo de (P^τ) quando as qualidades assumem, também, valores de melhores qualidades $(-20 \leq i \leq -1)$ e os valores $F_{q+i}(\bar{x}_q)$ são menores que o gargalo de (P^τ), nesse mesmo intervalo e, os valores $F_{q+i}(\bar{x}_{q-20})$ são melhores do que o gargalo de (P^τ) quando $-20 \leq i \leq -9$. Os valores $F_{q+i}(\bar{x}_{q+20})$ são todos piores do que o gargalo de (P^τ), como mostra a Figura 5.4(b). Concluímos que, na maioria dos casos analisados, o gargalo do modelo estocástico (P^τ) é melhor do que o gargalo de (P^*), quando os valores das qualidades dos enlaces assumem valores de qualidades maiores, ou seja, quando as qualidades das conexões são piores.

Como comentamos, anteriormente, rotas longas aumentam, consideravelmente, o atraso de transmissão e a taxa de perda de pacotes. Assim, podemos escolher, dentre as 25 rotas eficientes, a rota eficiente da árvore que obteve o menor custo total de todos os fluxos (menor comprimento), por exemplo, para configuração de distribuição dos fluxos (1), múltiplas origens e múltiplos destinos, quando há correlação tipo 1, a menor rota ocorre na árvore 25 e, para o tipo 2, árvore 16 ou 22. Quando não há correlação, todas as rotas têm o mesmo comprimento total dos caminhos (veja Figura 5.5(a)).



(a) *Múltiplas origens e múltiplos destinos* (b) *Múltiplas origens e único destino*

Figura 5.5: Avaliação dos custos total dos resultados para as árvores de cenários de tamanho 60.

Os custos totais dos fluxos para a configuração (2) de distribuição dos fluxos, quando há correlação tipo 1, a menor rota ocorreu na árvore 9 e, para o tipo 2, na árvore 13. Quando não há correlação, todas as rotas têm o mesmo comprimento total dos caminhos (veja Figura 5.5(b)).

CAPÍTULO 6

Conclusão

Em algumas infraestruturas de rede, como em determinados backhauls de redes de acesso e em algumas redes em malha sem fio, são formadas, predominantemente, por enlaces (arestas) com boa qualidade, ou seja, as qualidades são homogêneas. Outras podem possuir enlaces com qualidades heterogêneas, por exemplo, as redes veicular e de sensores sem fio. Temos, ainda, as redes sem fio de múltiplos saltos, onde a qualidade de cada enlace tende a flutuar regularmente, de maneira significativa e em curtas escalas de tempo. Além disso, pode existir correlação entre os eventos que influenciam as qualidades dos enlaces. Este trabalho resultou na criação de três modelos de problemas de roteamento de fluxos em redes, levando em consideração essas características das redes.

Os dois primeiros modelos são problemas de programação inteira biobjetivo. Visam rotear um conjunto de fluxos por caminhos, minimizando o gargalo da rede e o comprimento dos caminhos. No primeiro modelo (P), as qualidades das arestas não foram levadas em consideração, ou seja, as qualidades são homogêneas. O gargalo é a quantidade máxima de fluxos que atravessam as arestas. No segundo modelo ($\overline{P}$), as qualidades das arestas e os pesos dos fluxos compõem a função gargalo desse problema. Para isso ser possível, consideramos a função gargalo como uma soma ponderada desses dois parâmetros.

Os trabalhos anteriores mais próximos à esses problemas são os de [Mello et al. 2016] e de [Gálvez e Ruiz 2013], que trazem abordagens heurísticas para resolver esses tipos de problemas. Aqui, desenvolvemos um algoritmo exato e polinomial, com base no método ϵ-constraint, para obtermos um conjunto mínimo completo de soluções Pareto-ótimas.

Nos resultados computacionais, uma ampla avaliação de desempenho foi realizada. Vários fatores foram avaliados, como topologia, número de fluxos e distribuição de fluxos, e aqueles que tiveram maior impacto foram identificados em cada contexto. Por exemplo, o tipo de topologia tem um efeito significativo no tamanho do conjunto mínimo completo de soluções Pareto-ótimas. Além disso, para instâncias com fluxos de origens e destinos variados, o número de soluções Pareto-ótimas no conjunto mínimo completo é baixa. Isso consiste em um resultado importante, pois facilita a tomada de decisão em torno de qual solução escolher para fazer o roteamento dos fluxos.

Entretanto, na configuração em que os fluxos têm um único destino (e várias origens), na primeira solução Pareto-ótima, gerada pelo algoritmo, há alta carga concentrada sobre as arestas que incidem o nó de destino (único). Isso implicou numa maior cardinalidade do conjunto mínimo de soluções Pareto-ótimas.

Para o problema (P), quando consideramos os custos dos fluxos, para atravessarem as arestas, todos iguais, comparando a última solução Pareto-ótima com a primeira solução Pareto-ótima, geradas pelo algoritmo, percebemos que em cada instância

no mínimo 70% dos fluxos não sofrem alterações na quantidade de saltos. Isso mostra que a injustiça no comprimento do caminho causada pelo balanceamento de carga, tem pouco efeito em geral. Podemos validar melhor isso com o índice de justiça, que foi superior a 92%, para as topologias consideradas.

Nos resultados computacionais mostramos, também, que a primeira abordagem exata para resolver o problema (P) não garante a geração de um conjunto mínimo completo de soluções Pareto-ótimas para o problema ($\overline{P}$). Concluiu-se que há a necessidade de considerarmos os pesos dos fluxos e as qualidades das aresta, para resolver o problema ($\overline{P}$).

Nossa abordagem exata pode ser considerada cara em alguns cenários, por exemplo, em [Mello et al. 2016] e [Gálvez e Ruiz 2013]. O dinamismo do tráfego de fluxos e a demanda para decisões rápidas questionada pela abordagem heurística. Todavia, essa abordagem exata é uma ferramenta valiosa para avaliar se as soluções geradas pelas abordagens heurísticas são soluções equivalentes aos valores gerados pela abordagem exata ou aproximações consideráveis. A abordagem exata poderia servir, também, para combinar com a abordagem heurística para obter uma abordagem híbrida.

O último modelo apresentado neste trabalho, (P^{τ}), tratou de um problema estocástico mono-objetivo onde as qualidades das arestas são variáveis aleatórias discretas. O objetivo foi minimizar o gargalo da rede e, como restrição, controlar o comprimento total dos caminhos dos fluxos roteados.

Nos resultados computacionais, mostramos a limitação da abordagem de otimização determinística ao lidar com problemas de roteamento em redes sem fio de múltiplos saltos, devido à incerteza inerente aos enlaces de comunicação desse tipo de tecnologia. Mostramos, também, como a abordagem estocástica encontra uma solução eficiente ao incorporar no modelo de otimização o conhecimento sobre como as qualidades dos enlaces flutuam. Adicionalmente, evidenciamos como o modelo estocástico é capaz de representar eventuais correlações que possam existir entre as flutuações das qualidades dos enlaces de comunicação.

Para trabalhos futuros os problemas biobjetivos, apresentados neste trabalho, poderão ser investigados o uso do algoritmo exato aplicado a cenários reais de redes sem fio de múltiplos saltos, tais como redes em malha ou redes de sensores. Esses cenários poderão ser descritos através de um emulador de redes (e.g., Mininet-WiFi), o qual permitirá capturar métricas clássicas como vazão e atraso. Além do roteamento, o emulador permitirá representar características de enlaces sem fio IEEE 802.11 e imitar aplicações reais baseadas em protocolos de transporte da Internet (e.g., TCP ou UDP). Assim, será possível avaliar de maneira mais ampla as múltiplas soluções não dominadas que são geradas pelo nosso algoritmo exato.

A capacidade de representação sofisticada de técnicas para resolver o modelo estocástico implica em um modelo de alto custo computacional. Por essa razão, como trabalho futuro, pretendemos investigar técnicas de redução de cenários com o intuito de resolver o problema abordado para redes maiores e de representações reais. Também temos a intenção de implementar a solução em um simulador e/ou emulador para avaliar outras métricas, como vazão, atraso e perda de pacotes.

Referências Bibliográficas

[Abbas e Bellahcene 2006] ABBAS, M.; BELLAHCENE, F. Cutting plane method for multiple objective stochastic integer linear programming. *European Journal of operational research*, Elsevier, v. 168, n. 3, p. 967–984, 2006.

[Abbas e Moulai 1999] ABBAS, M.; MOULAI, M. Solving multiple objective integer linear programming. *Ricerca Operativa*, FrancoAngeli Editore, 1999.

[Abdel-Rahman et al. 2016] ABDEL-RAHMAN, M. J. et al. Dimensioning virtualized wireless access networks from a common pool of resources. In: IEEE. *Consumer Communications & Networking Conference (CCNC), 2016 13th IEEE Annual*. [S.l.], 2016. p. 1042–1047.

[Ahuja, Magnanti e Orlin 1993] AHUJA, R. K.; MAGNANTI, T. L.; ORLIN, J. B. Network flows: theory, algorithms, and applications. Prentice hall, 1993.

[Alvelos e Carvalho 2003] ALVELOS, F.; CARVALHO, J. V. D. Comparing branch-and-price algorithms for the unsplittable multicommodity flow problem. In: *International Network Optimization Conference*. [S.l.: s.n.], 2003. p. 7–12.

[Alves e Costa 2012] ALVES, M. J.; COSTA, J. P. Programação linear inteira e inteira-mista multiobjetivo: Conceitos fundamentais e métodos. In: *Congresso Latino-Iberoamericano de Investigacíon Operativa e Simpósio Brasileiro de Pesquisa Operacional. Rio de Janeiro*. [S.l.: s.n.], 2012.

[Barabási e Albert 1999] BARABÁSI, A. L.; ALBERT, R. Emergence of scaling in random networks. *science*, American Association for the Advancement of Science, v. 286, n. 5439, p. 509–512, 1999.

[Beale 1955] BEALE, E. M. On minimizing a convex function subject to linear inequalities. *Journal of the Royal Statistical Society. Series B (Methodological)*, JSTOR, p. 173–184, 1955.

[Berman, Einav e Handler 1990] BERMAN, O.; EINAV, D.; HANDLER, G. The constrained bottleneck problem in networks. *Operations Research*, INFORMS, v. 38, n. 1, p. 178–181, 1990.

[Bertsimas 1992] BERTSIMAS, D. J. A vehicle routing problem with stochastic demand. *Operations Research*, INFORMS, v. 40, n. 3, p. 574–585, 1992.

[Bianchi et al. 2009] BIANCHI, L. et al. A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing*, Springer, v. 8, n. 2, p. 239–287, 2009.

[Birge e Louveaux 2011] BIRGE, J. R.; LOUVEAUX, F. *Introduction to stochastic programming*. [S.l.]: Springer Science & Business Media, 2011.

[Biswas e Morris 2005] BISWAS, S.; MORRIS, R. ExOR: Opportunistic Multi-hop Routing for Wireless Networks. *SIGCOMM Comput. Commun. Rev.*, v. 35, n. 4, p. 133–144, Aug 2005.

[Borges 2013] BORGES, C. As rodovias brasileiras e o salto necessário para o primeiro mundo. *Palestra apresentada por Cesar Borges (Ministro dos Transportes). In: Fórum de Infraestrutura e Logística Belo Horizonte*, v. 7, 2013.

[Bornstein et al. 2012] BORNSTEIN, C. T. et al. Multiobjective combinatorial optimization problems with a cost and several bottleneck objective functions: an algorithm with reoptimization. *Computers & Operations Research*, Elsevier, v. 39, n. 9, p. 1969–1976, 2012.

[Chalmet, Lemonidis e Elzinga 1986] CHALMET, L. G.; LEMONIDIS, L.; ELZINGA, D. An algorithm for the bi-criterion integer programming problem. *European Journal of Operational Research*, Elsevier, v. 25, n. 2, p. 292–300, 1986.

[Chankong e Haimes 2008] CHANKONG, V.; HAIMES, Y. Y. *Multiobjective decision making: theory and methodology*. [S.l.]: Courier Dover Publications, 2008.

[Clímaco, Ferreira e Captivo 1997] CLÍMACO, J.; FERREIRA, C.; CAPTIVO, M. E. Multicriteria integer programming: An overview of the different algorithmic approaches. In: *Multicriteria analysis*. [S.l.]: Springer, 1997. p. 248–258.

[Clímaco e Martins 1982] CLÍMACO, J. C. N.; MARTINS, E. Q. V. A bicriterion shortest path algorithm. *European Journal of Operational Research*, Elsevier, v. 11, n. 4, p. 399–404, 1982.

[Clímaco e Pascoal 2012] CLÍMACO, J. C. N.; PASCOAL, M. M. B. Multicriteria path and tree problems: discussion on exact algorithms and applications. *International Transactions in Operational Research*, Wiley Online Library, v. 19, n. 1-2, p. 63–98, 2012.

[Clímaco, Antunes e Alves 2003] CLÍMACO, J. N.; ANTUNES, C. H.; ALVES, M. J. G. *Programação linear multiobjectivo: do modelo de programação linear clássico à consideração explícita de várias funções objectivo*. [S.l.]: Imprensa da Universidade de Coimbra/Coimbra University Press, 2003.

[Cohon 2013] COHON, J. L. *Multiobjective programming and planning*. [S.l.]: Courier Corporation, 2013.

[Cohon, Church e Sheer 1979] COHON, J. L.; CHURCH, R. L.; SHEER, D. P. Generating multiobjective trade-offs: An algorithm for bicriterion problems. *Water Resources Research*, Wiley Online Library, v. 15, n. 5, p. 1001–1010, 1979.

[Cohon e Marks 1973] COHON, J. L.; MARKS, D. H. Multiobjective screening models and water resource investment. *Water Resources Research*, Wiley Online Library, v. 9, n. 4, p. 826–836, 1973.

[Dantzig 2004] DANTZIG, G. B. Linear programming under uncertainty. *Management Science*, INFORMS, v. 50, n. 12_supplement, p. 1764–1769, 2004.

[Ehrgott 2006] EHRGOTT, M. A discussion of scalarization techniques for multiple objective integer programming. *Annals of Operations Research*, Springer, v. 147, n. 1, p. 343–360, 2006.

[Ehrgott e Gandibleux 2000] EHRGOTT, M.; GANDIBLEUX, X. A survey and annotated bibliography of multiobjective combinatorial optimization. *Or Spectrum*, Springer, v. 22, n. 4, p. 425–460, 2000.

[Ehrgott e Gandibleux 2003] EHRGOTT, M.; GANDIBLEUX, X. Multiobjective combinatorial optimization—theory, methodology, and applications. In: *Multiple criteria optimization: State of the art annotated bibliographic surveys*. [S.l.]: Springer, 2003. p. 369–444.

[Fernandes, Pinto e Cardoso 2019] FERNANDES, K. C. C.; PINTO, L. L.; CARDOSO, K. V. Flow routing aiming load balancing and path length in multi-hop networks with different link qualities. *Submetido para IEEE Communications Letters em abril de 2019*, 2019.

[Fernandes et al. 2019] FERNANDES, K. C. C. et al. Modelo de otimização estocástica para o problema de minimização de gargalo em redes sem fio com restrição no comprimento total dos caminhos dos fluxos. In: *Artigo a ser submetido para SBrT 2019 (XXXVII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais) em maio 2019*. [S.l.: s.n.], 2019.

[Ferreira e Wakabayashi 1996] FERREIRA, C. E.; WAKABAYASHI, Y. *Combinatória poliédrica e planos-de-corte faciais*. [S.l.]: UNICAMP-Instituto de Computacao, 1996.

[Gadegaard, Klose e Nielsen 2016] GADEGAARD, S. L.; KLOSE, A.; NIELSEN, L. R. A bi-objective approach to discrete cost-bottleneck location problems. *Annals of Operations Research*, Springer, p. 1–23, 2016.

[Gálvez e Ruiz 2013] GÁLVEZ, J. J.; RUIZ, P. M. Efficient rate allocation, routing and channel assignment in wireless mesh networks supporting dynamic traffic flows. *Ad Hoc Networks*, Elsevier, v. 11, n. 6, p. 1765–1781, 2013.

[Gendreau, Laporte e Séguin 1996] GENDREAU, M.; LAPORTE, G.; SÉGUIN, R. Stochastic vehicle routing. *European Journal of Operational Research*, Elsevier, v. 88, n. 1, p. 3–12, 1996.

[Haimes 1973] HAIMES, Y. Y. Integrated system identification and optimization. *Control and dynamic systems: Advances in theory and applications*, Academic Press, v. 10, p. 435–518, 1973.

[Haimes e Wismer 1971] HAIMES, Y. Y.; WISMER, D. A. Integrated system modeling and optimization via quasilinearization. *Journal of Optimization Theory and Applications*, Springer, v. 8, n. 2, p. 100–109, 1971.

[Hansen 1980] HANSEN, P. Bicriterion path problems. In: *Multiple criteria decision making theory and application*. [S.l.]: Springer, 1980. p. 109–127.

[Heeks 2010] HEEKS, R. Do information and communication technologies (icts) contribute to development? *Journal of international development*, Wiley Online Library, v. 22, n. 5, p. 625–640, 2010.

[Higham 1990] HIGHAM, N. J. *Analysis of the Cholesky decomposition of a semi-definite matrix*. [S.l.]: Oxford University Press, 1990.

[Higham 2000] HIGHAM, N. J. Computing the nearest correlation matrix. Citeseer, 2000.

[Hoffmann e Kruskal 1956] HOFFMANN, A.; KRUSKAL, J. Integral boundary points of convex polyhedra. *Linear inequalities and related systems*, p. 223–246, 1956.

[Høyland, Kaut e Wallace 2003] HØYLAND, K.; KAUT, M.; WALLACE, S. W. A heuristic for moment-matching scenario generation. *Computational optimization and applications*, Springer, v. 24, n. 2, p. 169–185, 2003.

[Jain, Chiu e Hawe 1984] JAIN, R.; CHIU, D.-M.; HAWE, W. R. *A quantitative measure of fairness and discrimination for resource allocation in shared computer system*. [S.l.]: Eastern Research Laboratory, Digital Equipment Corporation Hudson, MA, 1984.

[Kall, Wallace e Kall 1994] KALL, P.; WALLACE, S. W.; KALL, P. *Stochastic programming*. [S.l.]: Springer, 1994.

[Kaut e Wallace 2003] KAUT, M.; WALLACE, S. W. Evaluation of scenario-generation methods for stochastic programming. Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät II, Institut für Mathematik, 2003.

[King e Wallace 2012] KING, A. J.; WALLACE, S. W. *Modeling with stochastic programming*. [S.l.]: Springer Science & Business Media, 2012.

[Klein e Hannan 1982] KLEIN, D.; HANNAN, E. An algorithm for the multiple objective integer linear programming problem. *European Journal of Operational Research*, Elsevier, v. 9, n. 4, p. 378–385, 1982.

[Laporte, Louveaux e Mercure 1989] LAPORTE, G.; LOUVEAUX, F.; MERCURE, H. Models and exact solutions for a class of stochastic location-routing problems. *European Journal of Operational Research*, Elsevier, v. 39, n. 1, p. 71–78, 1989.

[Laporte e Osman 1995] LAPORTE, G.; OSMAN, I. H. Routing problems: A bibliography. *Annals of Operations Research*, Springer, v. 61, n. 1, p. 227–262, 1995.

[Liu et al. 2012] LIU, Q. et al. ISAR: Improved Situation-Aware Routing Method for Wireless Mesh Backbones. *IEEE Communications Letters*, v. 16, n. 9, p. 1404–1407, Sep 2012.

[Lokman e Köksalan 2013] LOKMAN, B.; KÖKSALAN, M. Finding all nondominated points of multi-objective integer programs. *Journal of Global Optimization*, Springer, v. 57, n. 2, p. 347–365, 2013.

[Lurie e Goldberg 1998] LURIE, P. M.; GOLDBERG, M. S. An approximate method for sampling correlated random variables from partially-specified distributions. *Management science*, INFORMS, v. 44, n. 2, p. 203–218, 1998.

[Marglin 1967] MARGLIN, S. A. *Public Investment Criteria*. [S.l.]: MIT Press, 1967.

[Martins 1984] MARTINS, E. Q. V. On a special class of bicriterion path problems. *European Journal of Operational Research*, Elsevier, v. 17, n. 1, p. 85–94, 1984.

[Maurras, Truemper e Akguel 1981] MAURRAS, J. F.; TRUEMPER, K.; AKGUEL, M. Polynomial algorithms for a class of linear programs. *Mathematical programming*, Springer, v. 21, n. 1, p. 121–136, 1981.

[Mello et al. 2016] MELLO, M. O. de et al. Improving load balancing, path length, and stability in low-cost wireless backhauls. *Ad Hoc Networks*, Elsevier, v. 48, p. 16–28, 2016.

[Meng et al. 2016] MENG, T. et al. Spatial Reusability-Aware Routing in Multi-Hop Wireless Networks. *IEEE Transactions on Computers*, v. 65, n. 1, p. 244–255, Jan 2016.

[Miller e Byers 1973] MILLER, W.; BYERS, D. Development and display of multiple-objective project impacts. *Water Resources Research*, Wiley Online Library, v. 9, n. 1, p. 11–20, 1973.

[Neto et al. 2011] NETO, C. A. d. S. C. et al. Gargalos e demandas da infraestrutura rodoviária e os investimentos do pac: mapeamento ipea de obras rodoviárias. Instituto de Pesquisa Econômica Aplicada (Ipea), 2011.

[Özlen e Azizoğlu 2009] ÖZLEN, M.; AZIZOĞLU, M. Multi-objective integer programming: a general approach for generating all non-dominated solutions. *European Journal of Operational Research*, Elsevier, v. 199, n. 1, p. 25–35, 2009.

[Özlen, Burton e MacRae 2014] ÖZLEN, M.; BURTON, B. A.; MACRAE, C. A. Multi-objective integer programming: An improved recursive algorithm. *Journal of Optimization Theory and Applications*, Springer, v. 160, n. 2, p. 470–482, 2014.

[Pflug e Pichler 2015] PFLUG, G. C.; PICHLER, A. Dynamic generation of scenario trees. *Computational Optimization and Applications*, Springer, v. 62, n. 3, p. 641–668, 2015.

[Pham e Perreau 2004] PHAM, P. P.; PERREAU, S. Increasing the network performance using multi-path routing mechanism with load balance. *Ad Hoc Networks*, v. 2, n. 4, p. 433–459, 2004.

[Pinto, Bornstein e Maculan 2009] PINTO, L. L.; BORNSTEIN, C. T.; MACULAN, N. The tricriterion shortest path problem with at least two bottleneck objective functions. *European Journal of Operational Research*, Elsevier, v. 198, n. 2, p. 387–391, 2009.

[Pinto e Fernandes 2016] PINTO, L. L.; FERNANDES, K. C. C. Um algoritmo exato para um problema de programação inteira biobjetivo. In: SOBRAPO. *Anais do XLVIII SBPO*. [S.l.], 2016. p. 484–494.

[Pinto et al. 2019] PINTO, L. L. et al. An Exact and Polynomial Approach for a Bi-Objective Integer Programming Problem Regarding Network Flow Routing. *Computers & Operations Research*, v. 106, p. 28–35, 2019.

[Pinto e Pascoal 2010] PINTO, L. L.; PASCOAL, M. M. On algorithms for the tricriteria shortest path problem with two bottleneck objective functions. *Computers & Operations Research*, Elsevier, v. 37, n. 10, p. 1774–1779, 2010.

[Ramachandran et al. 2007] RAMACHANDRAN, K. et al. Routing Stability in Static Wireless Mesh Networks. In: *Passive and Active Network Measurement*. [S.l.: s.n.], 2007. p. 73–82.

[Sahinidis 2004] SAHINIDIS, N. V. Optimization under uncertainty: state-of-the-art and opportunities. *Computers & Chemical Engineering*, Elsevier, v. 28, n. 6, p. 971–983, 2004.

[Schrijver 1998] SCHRIJVER, A. *Theory of linear and integer programming*. [S.l.]: John Wiley & Sons, 1998.

[Schrijver 2000] SCHRIJVER, A. *A course in combinatorial optimization*. [S.l.]: TU Delft, 2000.

[Shapiro 2003] SHAPIRO, A. Monte carlo sampling approach to stochastic programming. In: EDP SCIENCES. *ESAIM: Proceedings*. [S.l.], 2003. v. 13, p. 65–73.

[Shapiro e Mello 1998] SHAPIRO, A.; MELLO, T. Homem-de. A simulation-based approach to two-stage stochastic programming with recourse. *Mathematical Programming*, Springer, v. 81, n. 3, p. 301–325, 1998.

[Shapiro e Philpott 2007] SHAPIRO, A.; PHILPOTT, A. A tutorial on stochastic programming. *Manuscript. Available at www2. isye. gatech. edu/ashapiro/publications. html*, v. 17, 2007.

[Sylva e Crema 2004] SYLVA, J.; CREMA, A. A method for finding the set of non-dominated vectors for multiple objective integer linear programs. *European Journal of Operational Research*, Elsevier, v. 158, n. 1, p. 46–55, 2004.

[Tavares 2005] TAVARES, P. T. da S. Matrizes totalmente e quase totalmente unimodulares. *Dissertação (Mestrado) – Departamento de Matemática, Faculdade de Ciências e Tecnologia, Universidade de Coimbra*, 2005.

[Ulungu e Teghem 1994] ULUNGU, E. L.; TEGHEM, J. Multi-objective combinatorial optimization problems: A survey. *Journal of Multi-Criteria Decision Analysis*, Wiley Online Library, v. 3, n. 2, p. 83–104, 1994.