

UNIVERSIDADE FEDERAL DE MINAS GERAIS  
ESCOLA DE ENGENHARIA  
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE  
PRODUÇÃO

Alexandre Fortes da Silva Reis

O Problema do Transporte Escolar Rural: Uma abordagem  
*Column-and-cut* para o Problema de Roteamento de Veículos  
Capacitado

Belo Horizonte  
07 de julho de 2015

**Alexandre Fortes da Silva Reis**

**O Problema do Transporte Escolar Rural: Uma abordagem  
*Column-and-cut* para o Problema de Roteamento de Veículos  
Capacitado**

Dissertação apresentada ao Curso de Pós-Graduação em Engenharia de Produção da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Engenharia de Produção.

Orientador: Ricardo Saraiva de Camargo

**Belo Horizonte  
07 de julho de 2015**

## Agradecimentos

Gostaria de agradecer ao meu orientador Prof. Ricardo Camargo, por toda dedicação, paciência, conversas e diversas alternativas propostas para o desenvolvimento deste trabalho.

Ao meu coorientador Prof. Samuel Vieira pela acolhida no programa de pós-graduação e oportunidades concedidas de participar desse grande projeto de pesquisa sobre o Transporte Escolar Rural.

Ao Prof. Gilberto de Miranda Júnior por todas as sugestões na evolução deste estudo.

A banca examinadora pelas sugestões, retornos e suporte ao desenvolvimento de conhecimento.

À minha família, meu alicerce, minha mãe Cláudia, minha noiva Fernanda, meu pai José e irmãs Anna Carolina e Anna Cláudia por me acompanharem durante este percurso, pelo enorme apoio e pela paciência nos momentos em que me ausentei para me dedicar a este trabalho. Às Famílias Fortes, Cot(t)a e Matos por todo incentivo.

Aos meus amigos, Gustavo, Edson, Júlio, Arthur, Felipe, Leila, Tom, Amanda, Vinícius Lage, Vinícius Fortes, Gabi, Téis, Viviane e todos os que fiz no mestrado, em especial Fátima, Luiza, Guilherme, Bráulio e Leandro.

Aos meus professores da UFOP: Gustavo Peixoto Silva pela iniciação à pesquisa científica, André Luís, Lásara e Magno Silvério por serem também incentivadores desse caminho.

À FCO pelo apoio financeiro que foi imprescindível para a dedicação ao mestrado.

## Resumo

Existe uma situação de afastamento e dificuldade ao acesso a serviços públicos em geral, na qual a população rural está sujeita. Um dos principais, o acesso à educação, encontra-se a caminhar para um horizonte melhor através da disponibilização de transporte adequado aos alunos. Assim, este trabalho vem ao encontro do interesse do governo local, no sentido de auxiliá-lo na melhoria da disponibilização de um serviço de transporte escolar de melhor qualidade a um custo mais baixo. O trabalho aborda o assunto do transporte escolar rural, classificado como Problema de Roteamento de Veículos Capacitado, propondo algoritmos do tipo *Column-and-cut* para sua resolução de forma mais rápida e com limites aceitáveis. O problema mestre deriva da ideia de particionamento de conjunto de Balinski e Quandt (1964) que é combinado com três subproblemas. Estes subproblemas são os responsáveis por gerar as colunas, o método das *q-routes* proposto por Christofides, Mingozzi e Toth (1981a) e o *Problema do Caminho Mínimo com Restrição de Recursos, não-elementar* de Desrochers (1986) e *elementar* de Feillet et al. (2004). Uma vez que a formulação está relaxada, utilizou-se duas famílias de cortes para melhorar os limites obtidos, a primeira, proposta por Jepsen et al. (2008), que são as *Desigualdades sobre Subconjunto de Linhas* e a segunda, proposta por Contardo, Desaulniers e Lessard (2015) que são as *Restrições Fortes de Conexão*. Os modelos são avaliados através da aplicação em instâncias da literatura.

**Palavras-chaves:** Transporte escolar rural, Roteamento de veículos capacitado, Geração de colunas e cortes.

## *Abstract*

There is a situation of remoteness and difficulty of access to public services in general, in which the Brazilian rural population is submitted. One of the main, the access to education, is moving towards a horizon better by providing adequate transportation to students. This work is in line with the local government's interest in order to assist in improving the provision of a school transport service better quality at a lower cost. This paper addresses the issue of rural school transportation, classified as Capacitated Vehicle Routing Problem, proposing some Column-and-cut algorithms to its resolution faster and with acceptable limits. The chosen master problem is derived from the set partitioning proposed by Balinski e Quandt (1964) which is combined with three different subproblems. These ones are responsible for generate the columns, they are the q-routes method from Christofides, Mingozzi e Toth (1981a) and the Shortest Path Problem with Resources Constraints, non-elementar from Desrochers (1986) and the elementar from Feillet et al. (2004). Once the formulation is relaxed, two families of cuts were used to improve the obtained bounds, the first one is the Subset-row Inequalities from Jepsen et al. (2008) and the second one the Strong Degree Constraints from Contardo, Desaulniers e Lessard (2015). The models are evaluated using some of literature benchmarking instances.

**keyword:** Rural Scholar Transportation, Capacitated Vehicle Routing, Column-and-cut generation.

# Lista de Figuras

2.1	Veículos inadequados ao transporte escolar. . . . .	10
2.2	Veículo inadequado e estrada má conservada . . . . .	10
2.3	Iteratividade do algoritmo de Geração de Colunas. Elaboração própria. . . . .	17
2.4	Adição de colunas ao <i>PMR</i> . Adaptado de Spoorendonk (2008) . . . . .	17
2.5	Adição de Cortes ao <i>Problema</i> . Elaboração própria. . . . .	19
3.1	Ilustração do conceito de economia. Adaptado de Lysgaard (1997) . . . . .	21
3.2	Elaboração das rotas usando Clarke e Wright (1964). Elaboração própria. . . . .	22
3.3	Ilustração da recuperação de <i>q-routes</i> na matriz <i>R</i> . Elaboração própria. . . . .	24
3.4	Ilustração do algoritmo <i>SPPRC</i> . Elaboração própria. . . . .	30
3.5	Ilustração do algoritmo <i>ESPPRC</i> . Elaboração própria. . . . .	32
3.6	Esquema dos Algoritmos 5 e 6. Elaboração própria. . . . .	39
3.7	Esquema do Algoritmo 7. Elaboração própria. . . . .	40
3.8	Adição de colunas e cortes ao <i>PMR</i> . Adaptado de Spoorendonk (2008) . . . . .	42
4.1	<i>GAP</i> e Tempo de processamento para os testes do modelo <i>q-routes</i> . . . . .	47
4.2	<i>GAP</i> e Tempo de processamento para os testes do modelo <i>q-routes-SSR</i> . . . . .	49
4.3	<i>GAP</i> e Tempo de processamento para os testes do modelo <i>q-routes-SDC</i> . . . . .	51
4.4	<i>GAP</i> e Tempo de processamento para os testes do modelo <i>ng-SPPRC</i> . . . . .	53
4.5	<i>GAP</i> e Tempo de processamento para os testes do modelo <i>ng-SPPRC-SSR</i> . . . . .	55
4.6	<i>GAP</i> e Tempo de processamento para os testes do modelo <i>ng-SPPRC-SDC</i> . . . . .	57
4.7	<i>GAP</i> e Tempo de processamento para os testes do modelo <i>ESPPRC</i> . . . . .	59
4.8	<i>GAP</i> e Tempo de processamento para os testes do modelo <i>ESPPRC-SSR</i> . . . . .	61
4.9	Rotas ótimas para problemas da literatura. Fonte Augerat (1995). . . . .	64

# Lista de Tabelas

2.1	Classificações do <i>PRV</i> Gendreau e Potvin (1998) . . . . .	6
4.1	Instâncias e respectivas capacidades . . . . .	43
4.2	Resultados para o modelo <i>q-routes</i> . . . . .	46
4.3	Resultados para o modelo <i>q-routes-SSR</i> . . . . .	48
4.4	Resultados para o modelo <i>q-routes-SDC</i> . . . . .	50
4.5	Resultados para o modelo <i>ng-SPPRC</i> . . . . .	52
4.6	Resultados para o modelo <i>ng-SPPRC-SSR</i> . . . . .	54
4.7	Resultados para o modelo <i>ng-SPPRC-SDC</i> . . . . .	56
4.8	Resultados para o modelo <i>ESPPRC</i> . . . . .	58
4.9	Resultados para o modelo <i>ESPPRC-SSR</i> . . . . .	60
4.10	Resumo dos Resultados . . . . .	62
4.11	Instâncias que obtiverem <i>LI</i> somente usando o subproblema das <i>q-routes</i> . . . . .	63

# Lista de Algoritmos

1	Etapa de Inicialização para o método das <i>q-routes</i> . . . . .	25
2	Etapa de Preenchimento para o método das <i>q-routes</i> . . . . .	25
3	Etapa de Recuperação para o método das <i>q-routes</i> . . . . .	26
4	Problema do Caminho Mínimo ( <i>Elementar</i> ) com Restrição de Recursos . . . . .	28
5	<i>Esquema Column-and-Cut elementar</i> . . . . .	38
6	<i>Esquema Column-and-Cut não-elementar</i> . . . . .	40
7	<i>Esquema Column-and-Cut gradativamente elementar</i> . . . . .	41



# Sumário

<b>1</b>	<b>Introdução</b>	<b>2</b>
1.1	Apresentação do Problema . . . . .	2
1.2	Justificativa e relevância da pesquisa . . . . .	3
1.3	Objetivos . . . . .	4
1.3.1	Geral . . . . .	4
1.3.2	Específicos . . . . .	4
1.4	Estrutura do trabalho . . . . .	4
<b>2</b>	<b>Exame de Literatura</b>	<b>5</b>
2.1	O Problema de Roteamento de Veículos . . . . .	5
2.1.1	Variações do <i>PRV</i> . . . . .	6
2.2	O Problema de Roteamento de Transporte Escolar . . . . .	7
2.3	O Problema de Roteamento de Transporte Escolar Rural . . . . .	8
2.3.1	O Problema de Roteamento de Transporte Escolar Rural no Brasil . . . . .	9
2.4	Formulações para o <i>PRV</i> Capacitado . . . . .	10
2.4.1	Formulação do <i>PRV</i> por <i>Dois índices</i> . . . . .	11
2.4.2	Formulação do <i>PRV</i> de Miller, Tucker e Zemlin (MTZ) . . . . .	11
2.4.3	Formulação do <i>PRV</i> por <i>Uma Mercadoria</i> . . . . .	12
2.4.4	Formulação do <i>PRV</i> por <i>Duas Mercadorias</i> . . . . .	13
2.4.5	Formulação do <i>PRV</i> por <i>Várias Mercadorias</i> . . . . .	14
2.4.6	Formulação do <i>PRV</i> por arcos capacitados . . . . .	15
2.5	Método de Resolução do Problema . . . . .	15
2.5.1	Método de Geração de Colunas . . . . .	16
2.5.2	Métodos de Separação . . . . .	18
<b>3</b>	<b>Formulação do Problema</b>	<b>20</b>
3.1	Formulação do <i>PRV</i> por Particionamento de Conjunto . . . . .	20
3.2	Soluções Iniciais . . . . .	21
3.2.1	Heurística de Roteamento Econômico de Clarke-Wright . . . . .	21
3.3	Subproblemas de Precificação . . . . .	23
3.3.1	Subproblema de Precificação Simples ( <i>q-routes</i> ) . . . . .	23
3.3.2	O Problema de Caminho Mínimo com Restrição de Recursos . . . . .	27
3.3.2.1	O Caminho Mínimo <i>Elementar</i> com Restrição de Recursos . . . . .	27
3.3.2.2	Algoritmo para o <i>SPPRC Elementar e Não-elementar</i> . . . . .	27
3.3.2.3	Dominância e extensão nos algoritmos <i>SPPRC</i> . . . . .	28
3.3.2.4	Introdução do conceito <i>ng-SPPRC</i> . . . . .	33
3.4	Planos de Cortes . . . . .	33
3.4.1	Restrições sobre Subconjunto de Linhas . . . . .	34
3.4.2	Restrições Fortes de Conexão . . . . .	36
3.5	Estratégia de aceleração . . . . .	37
3.5.1	Relaxação Incremental de Espaço . . . . .	37

3.6	Esquemas para obtenção do limite inferior <i>elementar</i> . . . . .	38
3.6.1	Esquema <i>puramente elementar</i> . . . . .	38
3.6.2	Esquemas <i>gradativamente elementar</i> . . . . .	39
<b>4</b>	<b>Experimentos e resultados computacionais</b>	<b>43</b>
4.1	Instâncias utilizadas e primeiros experimentos . . . . .	43
4.2	Resultados . . . . .	45
4.2.1	Resultados utilizando o subproblema <i>q-routes</i> . . . . .	46
4.2.2	Resultados utilizando o subproblema <i>ng-SPPRC</i> . . . . .	52
4.2.3	Resultados utilizando o subproblema <i>ESPPRC</i> . . . . .	58
4.3	Análise dos resultados . . . . .	62
<b>5</b>	<b>Conclusões e Propostas Futuras</b>	<b>65</b>
5.1	Conclusões . . . . .	65
5.2	Propostas Futuras . . . . .	66
	<b>Referências Bibliográficas</b>	<b>67</b>

# Capítulo 1

## Introdução

*“Não pode existir triunfo sem perdas. Não há vitória sem sofrimento. Não há liberdade sem sacrifício.”*

Do filme *O Senhor dos Anéis: O Retorno do Rei*

### 1.1 Apresentação do Problema

O problema de Roteamento de Veículos (*PRV*) é um dos mais estudados na área da otimização combinatória e foi introduzido por Dantzig e Ramser (1959). O *PRV* tem como objetivo definir um conjunto de rotas entre um depósito e um determinado número de clientes de demanda conhecida. O problema busca então, minimizar a distância percorrida, o tempo ou até mesmo o número de veículos. As restrições básicas do problema consistem em: (a) Cada nó ou vértice é visitado uma única vez por um único veículo; (b) Cada rota é iniciada num depósito e finalizada no mesmo depósito; (c) Todas as demandas ou ofertas de todos os clientes devem ser satisfeitas. Apesar do seu enunciado relativamente simples, apresenta elevada complexidade computacional, o que o torna muito interessante e aplicável à dezenas de aplicações cotidianas, alguns exemplos podem ser citados, como entrega e coleta de correspondências, transporte de funcionários de uma empresa, entrega de mercadorias em horários pré-estabelecidos, além do Transporte Escolar, o qual este trabalho propõe-se a estudar.

O Problema de Roteamento de Transporte Escolar (*PRTE*) busca planejar um conjunto de rotas eficientemente para uma frota de ônibus escolares. Cada ônibus coleta estudantes de diversos pontos e os entrega às respectivas escolas, onde eles estão designados. Sendo derivado do *PRV*, deve-se respeitar determinadas restrições como a capacidade do ônibus, o tempo de deslocamento do aluno ao ponto e do ponto à escola, além da janela de tempo das escolas. O problema vem sendo estudado desde sua primeira publicação por Newton e Thomas (1969). A princípio, e em casos nacionais, as rotas eram construídas manualmente, a qualidade dos resultados era função da experiência do programador e sua construção demandava tempo e mão de obra qualificada. Desde a publicação desses autores outras abordagens foram adotadas, divergindo com relação à classificação, restrições e algoritmos usados. O número de trabalhos sobre o *PRTE* ainda é muito limitado, sendo alguns dos que se destacam o Desrosiers (1980) e Braca et al. (1997).

Contextualizando para o cenário nacional, O Problema de Roteamento de Transporte Escolar, tem encontrado uma vasta realidade de aplicação, ainda mais no caso do *Transporte Escolar Rural*. Isto ainda se deve, principalmente, à precariedade em que este serviço é realizado, por meio de vias sem condições de rodagem e veículos em péssimo estado de conservação.

Carvalho et al. (2010) listam diversos fatores que precisam de atenção para a melhoria do *Transporte Escolar Rural*, como demografia, distância até a escola, condições muitíssimo precárias de circulação das rodovias, péssimo estado de conservação dos veículos ou transporte (canoas, *pick-ups*) e segurança adequados, além de trabalhar com capacidade excedida, sendo notável a falta de planejamento e gestão capacitada do processo.

De acordo com Instituto Brasileiro de Geografia e Estatística (2014), isto se deve ao fato de que historicamente, as comunidades rurais tem sido ignoradas pelas ações e políticas sociais, o que permitiu aos residentes da zona rural manterem uma situação de baixa qualidade de vida e tendo de enfrentar muitos obstáculos para obter acesso ao seu bem-estar. Isso se deve também ao fato de que algumas áreas rurais são tão isoladas e com a população tão pequena, que existem poucas oportunidades de trabalho.

Assim, devido a segregação sofrida pelos meios rurais, o Transporte Escolar Rural surge como uma importante ferramenta de integração social e espacial, sendo um dos elementos que possibilita o deslocamento para cidades próximas e conseqüentemente, permitindo o acesso a uma gama maior de serviços. Segundo Bof e Sampaio (2006), as escolas localizadas no meio rural atendem a um valor próximo de 18% da população em idade escolar do ensino fundamental brasileiro. Esses dados mostram também que o número de escolas rurais é demasiadamente pequeno, o que intensificou os deslocamentos para as redes educacionais urbanas. Neste sentido, o transporte escolar rural é o principal meio de deslocamento desses alunos.

O programa *Caminho da Escola* (FNDE, 2014) foi criado pela *Resolução n° 3, de 28 de março de 2007*, do FNDE, e tem como beneficiários todos os alunos da educação básica das redes públicas dos estados e dos municípios residentes em áreas rurais. Em 2009, o governo federal promulgou o decreto No 6.768, de 11/02/2009, que visa disciplinar o programa e que lhe atribui cinco objetivos, que são: (a) renovar a frota de veículos escolares; (b) garantir segurança e qualidade ao transporte dos estudantes, por meio de padronização e inspeção de veículos; (c) contribuir para a redução da evasão escolar em observância ao Plano Nacional de Educação; (d) garantir o acesso e a permanência na escola dos estudantes matriculados na educação básica da zona rural; (e) obter a redução dos preços dos veículos e propiciar o aumento da transparência nas aquisições.

Desta forma, é notável que a disponibilização de meios de acesso a educação de qualidade é a melhor forma de garantir desenvolvimento à regiões rurais e a geração de oportunidades e conhecimento necessários para a obtenção de melhorias nas vidas dos seus moradores.

## 1.2 Justificativa e relevância da pesquisa

Atualmente, a redução dos custos de transporte para as organizações é uma de suas maiores preocupações (LIMA, 2006). Entre os custos logísticos que influenciam no custo final dos produtos, o de transporte é o de maior peso, afetando diretamente o lucro das organizações. Dessa forma, um bom planejamento de rotas é essencial para o sucesso ou fracasso das mesmas. Quando se trata de uma organização pública, isso toma uma proporção ainda maior, dada a responsabilidade de usar os recursos não só de maneira eficaz, mas de maneira eficiente.

Tanto quanto nas organizações industriais, o transporte escolar também busca a redução dos custos através da otimização de suas rotas. Além disso, os responsáveis preocupam-se em oferecer um serviço de melhor qualidade aos alunos que dependem desse serviço. Um caso especial do transporte escolar envolve a zona rural, a qual é fortemente afetada por uma série de fatores.

Existem diversas formas de se resolver o *PRVC* e este trabalho se aterá às técnicas de *Geração de Colunas e Cortes*. Albert (2006) afirmou que enquanto *planos de corte* podem ser resolvidos rapidamente, este processo pode ser não confiável. Os algoritmos de *Branch-and-bound* são bastante confiáveis, mas costumeiramente mais lentos. Por último, tem-se os métodos de *Geração de Colunas*, aplicados quando o número de variáveis possíveis é muito grande, mas não existe a necessidade de explicitar todas elas, somente as mais atrativas. Dessa forma, esse trabalho apresentará uma abordagem *Column-and-cut* com aplicações em instâncias da literatura em busca de melhores limites *elementares* para o uso futuro da técnica de *branch-and-bound*.

## 1.3 Objetivos

### 1.3.1 Geral

Buscar uma nova combinação que ofereça melhores limites para o *Problema de Roteamento de Veículos Capacitado (PRVC)* utilizando diferentes métodos de *Geração de Colunas* e de *Cortes*, para possível aplicação em um problema real de grande impacto na sociedade.

### 1.3.2 Específicos

Buscar os melhores limites para o *PRVC* utilizando a formulação por *particionamento de conjunto* e os subproblemas:

- Problema de Precificação Simples (*q-routes*);
- Problema do Caminho Mínimo com Restrições de Recursos;
- Problema do Caminho Mínimo Elementar com Restrições de Recursos.

Combinados com os cortes:

- Restrições Fortes de Conexão;
- Restrições sobre Subconjunto de Linhas.

## 1.4 Estrutura do trabalho

Este trabalho será apresentado em mais quatro capítulos. A revisão da literatura dos *Problemas de Roteamento de Veículos, Roteamento de Transporte Escolar e Roteamento de Transporte Escolar Rural*, além dos métodos de resolução do problema são elucidados no Capítulo 2. Este capítulo também apresenta algumas formulações já propostas para o *Problema de Roteamento de Veículos Capacitado* e o *Método de Resolução* do problema. No Capítulo 3 encontram-se a formulação adotada, os *subproblemas de precificação*, os *cortes* e os *esquemas* utilizados por este trabalho. O Capítulo 4 apresenta os resultados computacionais obtidos e a análise realizada sobre eles. Por fim, no Capítulo 5 são apresentadas as conclusões e propostas para trabalho futuros.

## Capítulo 2

# Exame de Literatura

### 2.1 O Problema de Roteamento de Veículos

Diversos problemas reais envolvem a otimização de rotas para um conjunto de veículos. Dessa forma, uma quantidade relevante de estudos baseados na definição inicial do problema feita há 50 anos por Dantzig e Ramser (1959), e vem trazendo diversas oportunidades de pesquisas e diferentes abordagens de solução. O roteamento de veículos representa um importante componente de vários sistemas de distribuição e transporte, e de forma básica, os *Problemas de Roteamento de Veículos (PRV)* são baseados em diversas peculiaridades, a exemplo, na coleta e na entrega, tanto de produtos quanto de pessoas, distribuição de alimentos, coleta de lixo, entrega de correspondência, entre outros.

Partindo dessa definição básica, o objetivo dos estudos nas área de roteamento é buscar soluções que atendam às demandas dos clientes buscando reduzir o custo final da operação ou até mesmo o tempo de operação e o deslocamento dos veículos. O *PRV*, apesar do seu enunciado relativamente simples, apresenta elevada complexidade computacional, o que o torna muito interessante para aplicação prática. As restrições básicas do problema consistem em: (a) Cada cliente é visitado uma única vez por um único veículo; (b) Cada rota é iniciada num depósito e finalizada no mesmo depósito e; (c) Todas as demandas ou ofertas de todos os clientes devem ser satisfeitas.

Entre diversos estudos do *PRV*, duas categorias podem ser identificadas. De um lado, os métodos heurísticos sendo utilizados para encontrar soluções de qualidade aceitável rapidamente, usualmente baseadas em técnicas de busca local (Rochat e Taillard (1995), Cordeau et al. (2001), Gendreau, Hertz e Laporte (1994), Laporte (1992)). Algumas vezes, uma abordagem híbrida é utilizada com programação restrita, e do outro lado, tem-se os métodos exatos focados em encontrar a solução ótima. baseados em *Programação Linear Inteira*, Faz-se notar: (i) As técnicas de *Branch-and-bound* (LAND; DOIG, 1960) que dependem da qualidade dos limites inferiores usados para limitar a árvore de busca. Entre diversos trabalhos, alguns que se destacam são o de Christofides, Mingozzi e Toth (1981a) definiram seus limites inferiores utilizando *Relaxação Lagrangeana* (GEOFFRION, 1974). Fisher (1994) utilizando uma generalização do método da *Árvore Geradora*. (ii) Os métodos de *Programação Dinâmica (PD)* também já foram amplamente aplicados buscando-se limites inferiores melhores. Christofides, Mingozzi e Toth (1981b) apresentaram três formulações do *PRVC* e introduziram o conceito de relaxação de espaço para *PD* de forma recursiva obtendo bons limites inferiores para a solução ótima. (iii) O método de *Branch-and-cut*, inicialmente proposto por Chvátal (1973) é baseado em técnicas de planos de corte utilizadas para fortalecer a relaxação linear do formulação inteira do *PRVC* através da adição de inequações válidas. (iv) O método de *Particionamento de Conjunto* foi introduzido por Balinski e Quandt (1964). Christofides, Mingozzi e Toth (1981a) também apresentaram uma formulação por particionamento de conjunto que geravam limites inferiores aceitáveis. Hadjiconstantinou, Christofides e Mingozzi (1995) descreveram um método para contabilização de uma

solução dual viável para a relaxação linear de uma formulação de particionamento de conjunto, sendo esta relaxação baseada em  $q$ -routes e *Problemas de Caminho Mínimo*. (iv) Os métodos de *Geração de Colunas* (Desrochers, Desrosiers e Solomon (1992), Feillet et al. (2004), Chabrier (2006), Dumas, Desrosiers e Soumis (1991), Riera-Ledesma e Salazar-González (2013)) são parte desses métodos exatos e quando a relaxação linear de cada nó de uma árvore de *Branch-and-bound* é resolvido pela geração de colunas, é chamado de *Branch-and-price*. Uma possível variação desses métodos também é o método de *Branch-price-and-cut* ou *Branch-cut-and-price* que englobam as abordagens de *GC* e a introdução de *planos de corte* juntamente com o crescimento da *árvore de busca*. Tanto os os cortes quanto as novas *variáveis* podem ser adicionadas somente no *nó raiz da árvore de buscas* ou em cada uma de suas ramificações, de acordo com o esquema adotado pelo autor. Fukasawa et al. (2006) utilizaram a tradicional *Relaxação Lagrangeana* com o uso de  $q$ -routes para calcular os limites com associação as restrições de grau de conexão e capacidade. Com isso, os autores conseguiram estabelecer uma interseção entre um polítopo convexo e sua projeção limitada pelos cortes adicionados, diminuindo e direcionando a busca. Ropke e Cordeau (2009) usaram como *PM* a formulação de *Particionamento de Conjunto* e seu problema de precificação é baseado no *Problema do Caminho Mínimo com Restrição de Recursos* (DESROCHERS, 1986). Dado que o subproblema não oferece somente rotas *elementares*, eles adicionaram inequações que tornavam inviáveis estas rotas e outras baseadas no atendimento a um subconjunto de clientes e que impediam a ocorrência de bifurcações deixando ou chegando a um cliente. Pessoa, Uchoa e Poggi de Aragão (2009) implementaram um algoritmo baseado na decomposição de Dantzig e Wolfe (1960) capaz de resolver instâncias com até 100 clientes. As colunas foram associadas a  $q$ -rotas e desigualdades de capacidade que não aumentavam a complexidade do problema foram adicionadas ao *PM*. Gauvin, Desaulniers e Gendreau (2014) propuseram um método que usou o *PD* para a geração de colunas combinada com eliminação de ciclos, *Busca Tabu* (GLOVER, 1989) e rotulamento bidirecional. Além disso, adicionaram restrições de capacidade e as propostas por Chvátal (1973).

### 2.1.1 Variações do *PRV*

Uma vez que o *PRV* é derivado do Problema do Caixeiro-Viajante (*PCV*) e o *PCV* é *NP-Difícil* (LENSTRA; KAN, 1981), logo o *PRV* também é *NP-Difícil*. O *PCV* é considerado um caso especial, onde se tem somente uma rota ( $m = 1$ ) e capacidade infinita ( $C = \infty$ ), constituindo assim, na procura pelo caminho mais curto, ou de menor custo, entre vários vértices que devem ser obrigatoriamente visitados. Na prática, o *PRV* é considerado mais difícil de resolver do que o *PCV* de mesmo tamanho e pode ser subdividido em várias classes. Essas classificações não possuem um padrão e cada autor apresenta especificações diferentes. Gendreau e Potvin (1998) classificam em quatro grupos conforme a Tabela 2.1.

Tabela 2.1: Classificações do *PRV* Gendreau e Potvin (1998)

	Muitos para Muitos	Um para Muitos
Capacitado	Transporte de Passageiros	Entrega de comida
Não capacitado	Serviço de correios	Serviços de reparos

A razão de agrupar os problemas nessas duas classes se justifica pelo fato de que nem todos os serviços de transporte requerem transportar mercadorias ou ainda porque as mercadorias possuem peso ou tamanho irrelevantes. No primeiro caso, do emprego de veículos capacitados pode-se citar o transporte de passageiros, onde existe um limite máximo para o tamanho da carga. O segundo caso pode ser exemplificado com o transporte de cartas ou quando não se trata de transporte de mercadorias, mas de serviço a ser prestado, como um chaveiro ou um electricista. Na literatura existem três classificações costumeiramente adotadas e que se ramificam em diversas formulações. Elas são: (i) Problema de Roteamento de Veículos Capacitado, (ii) Problema de

Roteamento de Veículos com Janela de Tempo e (iii) o Problema de Roteamento de Veículos com Coleta e Entrega.

O Problema de Roteamento de Veículos Capacitado (*PRVC*) é a generalização do *PRV* na qual um conjunto de veículos, idênticos e localizados num depósito central, devem ser roteados de forma ótima, ou seja, minimizando os custos das rotas, e atender a demanda conhecida de um conjunto de clientes, respeitando as restrições de capacidade dos veículos (Gendreau e Potvin (1998), Baldacci, Hadjiconstantinou e Mingozzi (2004)). O Problema de Roteamento de Veículos com Janela de Tempo (*PRVJT*) envolve a complexibilidade de atender determinado cliente dentro de uma janela de tempo específica. Se o veículo atinge o cliente antes da abertura da janela, ele deverá esperar até que esta seja aberta, o não atendimento do cliente dentro da janela é considerado uma inviabilidade (Desrochers, Desrosiers e Solomon (1992), Desrochers et al. (1988)). Por fim, O Problema de Roteamento de Veículos com Coleta e Entrega (*PRVCE*) busca atender com o mesmo veículo as demandas de cada consumidor, que podem ser de coleta e/ou entrega (Savelsbergh e Sol (1995), Dumas, Desrosiers e Soumis (1991)). De forma geral, todos estes problemas buscam minimizar os custos de conexão entre os clientes visitados por uma rota sem infringir suas restrições, seja de capacidade, janela de tempo ou coleta e entrega.

## 2.2 O Problema de Roteamento de Transporte Escolar

O setor de transporte é um setor muito abrangente e estudado devido a sua importância econômica. Ele está diretamente relacionado, então, a diversos outros fatores, como condições da infraestrutura, da frota e das operações. Segundo Ballou (2006), de toda carga transportada no país, o modal rodoviário movimentava 61,1%. A ênfase nas rodovias eleva o custo de transporte de cargas, fazendo com que esta se torne a atividade logística que absorve a maior parte dos custos das empresas.

Um dos ramos de atividade relacionado ao transporte, é o transporte escolar. Este especificamente se refere ao transporte de estudantes de determinado ponto próximo a sua residência à escola onde realiza suas atividades. O Problema de Roteamento de Transporte Escolar (*PRTE*) teve sua primeira publicação por Newton e Thomas (1969) num trabalho que em que geravam uma rota inicial de capacidade infinita e a dividiam em rotas menores. Neste trabalho já buscaram trabalhar com informações como a demanda de estudantes por cada um dos pontos de parada, a capacidade do ônibus e entrega dos mesmos à escola que eles são designados, além de respeitar as restrições de janela de tempo, distância máxima percorrida pelo aluno, tempo de viagem, entre outras. A princípio, as rotas eram construídas manualmente, a qualidade dos resultados era função da experiência do programador e sua construção demandava tempo de mão de obra qualificada. Dessa forma, estudos e investimentos nesse setor de transporte surgem como forte aliado na integração social e espacial, principalmente no meio rural, foco da pesquisa. Contudo, a oferta desse serviço nem sempre é realizada de forma correta e adequada às necessidades dos alunos da zona rural. Chen et al. (1990) fazem notar que a disparidade entre os ambientes rural e urbano é evidenciada devido a vários aspectos tais como: baixa densidade populacional, maiores distâncias percorridas por rotas, ausência de ruas de mão única, poucos estudantes por ponto de embarque ou desembarque, menos pontos por rota, em alguns casos, os veículos passam a noite na casa do próprio condutor e existem poucas rotas alternativas disponíveis para sua construção.

Diversas abordagens já foram adotadas para o *PRTE*, divergindo com relação à classificação, restrições e algoritmos usados. A estrutura de solução de problemas com escola única é similar a obtida através do tradicional Problema de Roteamento de Veículo. No *PRV* tradicional, como já mencionado, a rota tem início no depósito, visita os clientes e retorna ao ponto inicial. Entretanto, no *PRTE*, o depósito (garagem), normalmente, possui localização diferente do ponto final, a escola. No entanto algumas abordagens ainda têm sido pouco exploradas como é o caso do transbordo e da frota heterôgenea (PARK; KIM, 2010).



Visto que o o *PRTE* é derivado do *PRV* diversos pontos e restrições podem ser abordados, como exemplo, se são múltiplas escolas, se a rota é gerada com base na localização da escola ou na residência dos estudantes, se o ambiente é rural ou urbano, qual o período de coleta dos alunos nos pontos e entrega em suas residências, tipo de frota, se há pontos de transbordo, capacidade do veículo, tempo máximo de viagem, janelas de tempo da escola, distância máxima percorrida e número máximo de rotas.

Ao longos dos anos, diversos trabalhos tentaram resolver o *PRTE* por métodos exatos ou heurísticos. Pacheco et al. (2013) utilizaram otimização multiobjetivo para resolver o *PRTE* em área rural. Partindo da ideia da minimização de objetivos conflitantes, entre a distância percorrida e a duração das viagens, os autores buscaram uma fronteira eficiente para permitir ao tomador de decisão a possibilidade de realizar um *trade-off* entre os objetivos. Pacheco e Martí (2006) também apresentaram uma bordagem heurística, utilizando a estratégia de empregar uma estrutura de *path-relinking* com a *Busca Tabu* e obter soluções não-dominadas. Neste trabalho buscaram minimizar o tamanho da frota e o tempo em que um estudante podia permanecer em trânsito.

Mandujano, Giesen e Ferrer (2012) propuseram um modelo que buscava otimizar a localização das escolas e as rotas para transporte dos alunos. O método era baseado em programação linear mista e era resolvido sequencialmente, sendo que o primeiro tratava da localização das escolas e o segundo no roteamento dos ônibus. Spada, Bierlaire e Liebling (2005) aplicaram as metaheurísticas *Simulated Annealing* e *Busca Tabu* para melhorar a solução inicial gerada por heurística de inserção. Thangiah e Nygard (1992) utilizaram algoritmos genéticos para minizar a frota de veículos e a distância viajada pelos mesmos e pelos alunos. A metaheurística foi aplicada através do sistema denominado GENROUTER para traçar a rota de dois distritos escolares.

Fügenschuh (2011) propôs uma reformulação para o problema de programação dos ônibus e da abertura da janela de tempo da escola usando *particionamento de conjunto*. Em um outro trabalho, Fügenschuh (2009) propôs uma formulação de programação linear inteira para integrar e coordenar o atendimento do transporte com os horários de pico e janelas de tempo das escolas.

Cortés, Matamala e Contardo (2010) utilizaram uma formulação baseada na decomposição de Benders (1962) conjugada com *branch-and-bound* na qual era permitido a transferência de cargas entre veículos em alguns pontos determinados e os cortes eram gerados pelo subproblema dual.

Riera-Ledesma e Salazar-González (2012) combinaram a seleção da parada e da rota a ser realizada pelo veículo, dada uma frota heterogênea. Realizaram isso através de uma formulação baseada no *Fluxo de Uma Mercadoria* com a adição de restrições sobre as variáveis de fluxo. Obtiveram resultados para instâncias simétricas e assimétricas de até 125 clientes.

Li e Fu (2002) trabalharam com otimização multiobjetivo minimizando objetivos conflitantes, tempo de viagem e total de ônibus utilizados, e propuseram um algoritmo que utilizando dados de instâncias reais conseguiram economizar até 29% o tempo total de viagem.

## 2.3 O Problema de Roteamento de Transporte Escolar Rural

Diversas adversidades ainda são encontradas pela população rural dos países em desenvolvimento ao se buscar serviços básicos como educação e saúde. Voltando-se para a educação, conforme levantado por Carvalho e Yamashita (2011), os estudantes enfrentam um enorme conjunto de dificuldades para se manterem na escola, entre elas estão por exemplo, a alimentação, material didático e vestuário. No entanto, o transporte é a principal. Diante desse cenário, a oferta de ensino público gratuito, muitas vezes, não é suficiente para que os estudantes tenham acesso à escola ou mesmo para que permaneçam nela. Sendo o acesso às escolas bem difícil, as consequências são visíveis na qualidade de vida e bem estar da população local, mesmo que o acesso à esses direitos garantidos em constituição, não sejam atendidos. Com a enorme dificuldade de se manterem na escola, os jovens ingressam no mercado de trabalho mais cedo, pois abandonam

as escolas em média após três anos de estudos, ao contrário das crianças da zona urbana que desfrutaram de um período bem mais longo de estudos. Outro fator que resulta na ausência ou ineficiência do sistema é o custo elevado de prover uma rede de transporte para uma população pouco densa ou muito dispersa o que, muitas vezes, torna o serviço de difícil execução fazendo com que a região fique ainda mais isolada e pobre. Poucos estudos e revisões têm sido conduzidas no Brasil com relação ao transporte de alunos na zona rural ou seu acesso a unidades de ensino. A nível nacional, a extinta Empresa de Planejamento de Transportes (GEIPOT), conduziu um dos primeiros estudos na área em 1995 e Carvalho e Yamashita (2011), Carvalho et al. (2010), Vasconcellos (1997) e Mandujano, Giesen e Ferrer (2012) deram sequência realizando estudos sobre o problema do transporte rural e acessibilidade dos moradores aos serviços públicos básicos.

O número de trabalhos desenvolvidos nesta área ainda é pequeno dada a riqueza do problema, onde várias possibilidades podem ser investigadas. Isso ocorre porque o Brasil é um país extenso e possui uma variedade de culturas, gerando problemas específicos para cada região. No entanto, como será visto posteriormente, algumas dessas particularidades tornam difícil sua resolução e conseqüentemente a criação de um modelo geral que possa atender várias regiões ao mesmo tempo. Com isso, a demanda de técnicos para atuar nessa área se torna ainda maior.

São muitos os desafios a serem enfrentados, não somente na área técnica, pela riqueza do problema, mas pelo impacto que o transporte tem nas condições de vida da população rural. Dessa forma o problema se torna não somente técnico, mas social, exigindo dedicação e empenho do governo a fim de garantir condições de equidade, segurança, conforto e melhores perspectivas de vida para os residentes da área rural.

### **2.3.1 O Problema de Roteamento de Transporte Escolar Rural no Brasil**

O cenário nacional apresenta diversas especificidades, devido a forma como a zona rural ao longo dos anos teve a atenção necessária negligenciada. Dessa forma, faz-se notar alguns dos principais itens que impedem o pleno funcionamento do sistema de educação na zona rural, como a péssima condição das estradas de rodagem, veículos deteriorados ou inadequados, grandes distâncias percorridas, programação das rotas e disponibilidade do transporte nos horários adequados.

Carvalho et al. (2010) apontam diversos fatores que precisam de atenção para a melhoria do Transporte Escolar Rural são eles: demografia (17% e 83% em escolas estaduais e municipais, respectivamente); distância até a escola (30% desses estudantes percorrem mais de 50 km até a escola, e há casos extremos de mais de 140 km), incluindo caminhadas de longa duração; mais de 90% das vias percorridas estão em condições muitíssimo precárias de circulação e a manutenção é praticamente inexistente (Figura 2.1); os veículos se encontram em péssimo estado de conservação e em grande maioria são velhos (casos de mais de 70 anos de atividade) ou inadequados demais (canoas, *pau-de-araras*) (Figura 2.2), sem segurança adequada (cinto, encosto de cabeça, saída de emergência, etc.) e trabalhando com capacidade excedida (Estudos mostram que a taxa média de ocupação é de 1,2%, o que significa que os ônibus transportam 20% a mais de estudantes que o permitido); por fim, a total falta de planejamento e administração do serviço.

Além da área de transportes, a zona rural também enfrenta dificuldades nas áreas de saúde e educação. Isso ocorre principalmente devido ao seu isolamento geográfico, baixa densidade populacional e às condições econômicas que unidas, dificultam e limitam os serviços de transporte, essencial para seu deslocamento. Grandes distâncias a serem percorridas são um fator particularmente crítico tratando-se de transporte escolar rural nacional. Nessas áreas as crianças muitas vezes, necessitam caminhar longas distâncias até o ponto de embarque além de distâncias adicionais até a escola, seja da área urbana ou não. Os valores máximos de caminhada encontrados na literatura foram de 2 a 3 quilômetros entre a residência e o ponto na estrada onde passa o veículo (VASCONCELLOS, 1997). Além da caminhada recomenda-se que o tempo máximo de deslocamento do aluno de sua residência até a escola seja 45 minutos (ARANTES, 1986). No entanto, é comum os estudantes da zona rural enfrentarem um tempo de viagem muitas vezes mais longo, por estradas mal conservadas.



(a) Veículo bastante antigo utilizado para o transporte.



(b) Casos de situação precária de segurança e conservação.

Figura 2.1: Veículos inadequados ao transporte escolar.



(a) Caminhão *pau-de-arara* realizando o transporte dos alunos



(b) Veículo da frota renovada, mas em estrada sem conservação

Figura 2.2: Veículo inadequado e estrada má conservada

Outro fator que influencia na prestação do serviço é a existência de rodovias não pavimentadas e de difícil circulação, tornando a abordagem do problema mais delicada. Essa realidade influencia o tempo de viagem e a definição dos pontos de embarque, desembarque ou transbordo. Em algumas épocas do ano é comum o aumento da incidência de chuvas o que agrava ainda mais a situação. Com isso, o tempo de viagem pode aumentar e os pontos de parada podem ficar mais distantes das residências, aumentando o tempo de caminhada dos estudantes. Além disso, a extensão da área na qual os alunos estão dispersos é relativamente grande e difícil de cobrir, ocasionando um reduzido número de alunos por ponto de ônibus. Como consequência, aumenta-se o tempo de viagem e diminui o número de alunos atendidos por veículo, aumentando a demanda de veículos para realizar o serviço.

Diante desse cenário é clara a necessidade de medidas, tais como planejamento de rotas e controle das mesmas, que possam efetivamente melhorar as condições locais do transporte rural no Brasil, sendo este trabalho a tentativa de contribuição para a melhoria do processo decisório do serviço.

## 2.4 Formulações para o *PRV* Capacitado

Apesar de diversos problemas combinatoriais conhecidos, não existe uma definição totalmente aceita para o *PRV* dada a sua enorme diversidade de restrições encontradas na prática. Dessa forma, a maioria dos esforços de pesquisa foram direcionados a uma versão padrão do problema, chamada de *PRV Clássico*.

O *PRV Clássico* é definido como um grafo completo e não-direcionado  $G = (N, A)$  onde  $N = \{0, 1, \dots, n, n + 1\}$  é o conjunto de nós e  $A = \{(i, j) : i, j \in N, i \neq j\}$  é o conjunto de arcos. O vértice 0 representa o depósito, e  $n + 1$  sua cópia, onde estão localizadas  $k$  veículos idênticos de capacidade  $C$ . Com cada cliente,  $i \in \bar{N} = N \setminus \{0, n + 1\}$ , está associada determinada demanda não-negativa  $d_i \leq C$ . Uma matriz de custos  $c_{ij}$  é definida sobre  $A$ . Quando a matriz de custos é

simétrica, ou seja  $c_{ij} = c_{ji} \forall i, j \in N$ , é comum definir o problema como um grafo não-direcionado  $G = (N, E)$ ,  $E = \{(i, j) : i, j \in N, i < j\}$  é o conjunto de arestas utilizadas. O problema então, consiste em determinar um conjunto de  $k$  rotas que: (a) inicie no depósito e termine em sua cópia, de forma que (b) cada cliente seja visitado por um único veículo, (c) a demanda total da rota não exceda a capacidade  $C$ , e (d) o custo total da rota seja minimizado.

#### 2.4.1 Formulação do PRV por Dois índices

A formulação apresentada abaixo, foi retirada do trabalho de Baldacci, Toth e Vigo (2007) e originalmente foi proposta por Laporte, Nobert e Desrochers (1985) que permite somente rotas de um único cliente.

Seja  $\mathcal{S} = \{S : S \subseteq \bar{N}, |S| \geq 2\}$ . Para um dado subconjunto  $S \in \mathcal{S}$  denota-se por  $\bar{S}$  o conjunto complementar de nós  $N \setminus S$ . Seja  $r(S)$  o número mínimo de veículos de capacidade  $C$  necessários para satisfazer a demanda dos clientes em  $S \in \mathcal{S}$ . Também, seja  $d(S)$  a indicação da demanda total de um subconjunto  $S$ , tal que  $d(S) = \sum_{i \in S} d_i$  e  $x_{ij}$  uma variável inteira que pode tomar os valores  $\{0, 1\} \forall \{i, j\} \in A \setminus \{(0, j)\} : j \in \bar{N}$  e valores  $\{0, 1, 2\} \forall (0, j) \in A, j \in \bar{N}$ . Sendo que  $x_{ij} = 2$  quando a rota incluindo um único cliente na solução é selecionada.

$$\min \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (2.1)$$

s.t.:

$$\sum_{\substack{j \in N \\ i < j}} x_{ij} + \sum_{\substack{j \in N \\ i > j}} x_{ij} = 2 \quad \forall i \in N \setminus \{0\} \quad (2.2)$$

$$\sum_{i \in S} \sum_{\substack{j \in \bar{S} \\ i < j}} x_{ij} + \sum_{i \in \bar{S}} \sum_{\substack{j \in S \\ i < j}} x_{ij} \geq 2r(S) \quad \forall S \in \mathcal{S} \quad (2.3)$$

$$\sum_{j \in \bar{N}} x_{ij} = 2k \quad (2.4)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \setminus \{(0, j)\}, j \in \bar{N} \quad (2.5)$$

$$x_{ij} \in \{0, 1, 2\} \quad \forall (0, j) \in A, j \in \bar{N} \quad (2.6)$$

A restrição (2.2) é o grau de conexão para cada cliente. A restrição (2.3) é a restrição de capacidade, também conhecida como de eliminação de subrotas, sendo que  $r(S)$  entram e deixam um subconjunto  $S$  de clientes que não contém o depósito. Por fim, a restrição (2.4) define que  $k$  veículos devem deixar e retornar ao depósito. Devido a restrição (2.3) este problema é *NP-Difícil*, pois deve-se computar  $r(S)$ , ou seja, resolver um *Problema de Empacotamento*, onde  $r(S)$  é o número mínimo de *empacotamentos* de capacidade  $C$  necessários para armazenar todo a demanda no conjunto  $S$ . A função objetivo (2.1) contabiliza o custo de conexão entre os nós e as restrições (2.5) e (2.6) o domínio das variáveis.

#### 2.4.2 Formulação do PRV de Miller, Tucker e Zemlin (MTZ)

Esta formulação foi retirada do trabalho de Desrochers e Laporte (1991) e foi baseada na formulação para o *PCV* de Dantzig, Fulkerson e Johnson (1954) e a ela foi adicionada algumas restrições de eliminação de subrotas propostas por Miller, Tucker e Zemlin (1960).

$$\min \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (2.7)$$

s.t.:

$$\sum_{(i,j) \in A} x_{ij} = 1 \quad \forall i \in N \quad (2.8)$$

$$\sum_{(i,j) \in A} x_{ij} = 1 \quad \forall j \in N \quad (2.9)$$

$$u_i - u_j + n x_{ij} \leq (n - 1) \quad i, j \in N \setminus \{0\}, i \neq j \quad (2.10)$$

$$1 \leq u_i \leq n \quad \forall i \in N \setminus \{0\} \quad (2.11)$$

$$x_{ij} \in \{0,1\} \quad \forall (i,j) \in A \quad (2.12)$$

$$u_i \geq 0 \quad \forall i \in N \setminus \{0\} \quad (2.13)$$

A função objetivo (2.7) contabiliza o custo de conexão entre os nós. As restrições (2.8) e (2.9) são referentes ao grau de conexão. As restrições propostas por MTZ são as (2.10) e (2.11). A restrição (2.11) garante que existe somente uma solução correspondente a uma dada rota viável e (2.10) garante o fluxo e direção do caminho. Por fim, (2.12) e (2.13) definem o domínio das variáveis.

### 2.4.3 Formulação do *PRV* por *Uma Mercadoria*

A ideia da resolução por uma única mercadoria foi proposta por Gavish e Graves (1978) para a resolução do *PCV*. Após isso, ela foi adaptada para o *PRV* por Langevin, Soumis e Desrosiers (1990).

$$\min \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (2.14)$$

s.t.:

$$\sum_{(0,j) \in A} x_{0j} = k \quad (2.15)$$

$$\sum_{(i,0) \in A} x_{i0} = k \quad (2.16)$$

$$\sum_{(i,j) \in A} x_{ij} = 1 \quad \forall i \in N \setminus \{0\} \quad (2.17)$$

$$\sum_{(i,j) \in A} x_{ij} = 1 \quad \forall j \in \bar{N} \quad (2.18)$$

$$\sum_{(0,j) \in A} f_{0j} = \sum_{i \in N} d_i \quad (2.19)$$

$$\sum_{(i,j) \in A} f_{ij} - \sum_{(j,i) \in A} f_{ji} = d_j \quad \forall j \in \bar{N} \quad (2.20)$$

$$\sum_{(i,j) \in A} f_{ij} \leq C x_{ij} \quad (2.21)$$

$$x_{ij} \in \{0,1\} \quad \forall (i,j) \in A \quad (2.22)$$

$$f_{ij} \geq 0 \quad \forall (i,j) \in A \quad (2.23)$$

A função (2.14) representa a somatória dos custos de conexão entre os nós clientes. As restrições (2.15) e (2.16) definem que  $k$  veículos deixam o depósito e devem a ele retornar. Juntas, (2.17) e (2.18) garantem o grau de conexão dos nós usuários. A restrição (2.19) estabelece que o fluxo que deixa a origem é igual à demanda total dos clientes, (2.20) garante o equilíbrio do fluxo, ou seja, a parcela que deixa um nó é igual ao valor que chega menos a demanda do nó e (2.21) que define que o fluxo máximo que percorre uma aresta é no máximo igual à capacidade da rota, se a aresta está ativa. Por fim, (2.22) e (2.23) definem o domínio das variáveis  $x_{ij}$  e  $f_{ij}$ .

#### 2.4.4 Formulação do PRV por Duas Mercadorias

Esta formulação foi utilizada por Baldacci, Hadjiconstantinou e Mingozzi (2004) na resolução do *PRV Capacitado* baseada em uma rede com fluxo de duas mercadorias (*Two-Commodity Flow Formulation - TCFF*) proposta para o *PCV* por Finke, Claus e Gunn (1984). Nesta formulação, uma unidade da primeira mercadoria (*commodity*), chamada de  $A$ , deve ser entregue em cada nó e uma unidade da segunda mercadoria, chamada de  $B$ , deve ser coletada em cada nó. O caixeiro-viajante inicia a *rota* no depósito com  $n$  unidades da mercadoria  $A$  e 0 da mercadoria  $B$ . No próximo nó visitado, ela deixa uma unidade de  $A$  e pega uma unidade de  $B$ . Esta troca é realizada até que o caixeiro-viajante retorne ao depósito carregando  $n$  unidades de  $B$  e nenhuma unidade de  $A$ . Em cada arco da *rota* são sempre carregadas  $n$  unidades de mercadorias. O total da mercadoria  $A$  indica o número de nós deixados para ser visitados pelo caixeiro-viajante e o total de  $B$  indica o total de nós já visitados. Além disso, introduziram algumas desigualdades válidas que melhoraram os limites inferiores da formulação, se mostrando competitiva com a literatura.

A formulação usa duas variáveis de fluxo,  $f_{ij}$  e  $f_{ji}$  para representar uma aresta  $\{i,j\} \in A$  de uma solução factível do *PRV Capacitado* da qual o veículo carrega uma carga mista de  $C$  unidades. Se o veículo viaja de  $i$  para  $j$ , então o fluxo  $f_{ij}$  representa a carga de veículos e o fluxo  $f_{ji}$  o espaço vazio no veículo, ou seja,  $f_{ji} = C - f_{ij}$ . As variáveis de fluxo  $f_{ij} \forall i, j \in N, i \neq j$ , definem dois fluxos para qualquer *rota* de uma solução factível: Um caminho do nó 0 para o nó  $n + 1$  é dado pelas variáveis de fluxo representando a carga e o segundo caminho do nó  $n + 1$  para o nó 0 é dada pelo fluxo de variáveis representando o espaço vazio do veículo.

A formulação proposta para o *PRV Capacitado* é:

$$\min \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (2.24)$$

s.t.:

$$\sum_{j \in N} (f_{ij} - f_{ji}) = 2d_i \quad \forall i \in \bar{N} \quad (2.25)$$

$$\sum_{j \in \bar{N}} f_{0j} = \sum_{i \in \bar{N}} d_i \quad (2.26)$$

$$\sum_{j \in \bar{N}} f_{j0} = kC - \sum_{i \in \bar{N}} d_i \quad (2.27)$$

$$\sum_{j \in \bar{N}} f_{(n+1),j} = kC \quad (2.28)$$

$$f_{ij} + f_{ji} = Cx_{ij} \quad \forall \{i,j\} \in A \quad (2.29)$$

$$\sum_{j \in N: i < j} x_{ij} + \sum_{j \in N: i > j} x_{ji} = 2 \quad \forall i \in \bar{N} \quad (2.30)$$

$$f_{ij} \geq 0, f_{ji} \geq 0 \quad \forall \{i,j\} \in A \quad (2.31)$$

$$x_{ij} \in \{0,1\} \quad \forall \{i,j\} \in A \quad (2.32)$$

A função objetivo contabiliza o custo total de conexão. As equações (2.25)-(2.28) e as restrições de não-negatividade (2.31) definem um fluxo factível dos nós origem 0 e  $n + 1$  para nós em  $N$ . O fluxo de escoamento no nó origem 0 (2.26) é igual a demanda total dos clientes e o fluxo de afluência na fonte  $n + 1$  (2.28) corresponde a capacidade total da frota dos veículos. A restrição (2.25) garante que o fluxo que chega menos o que deixa cada nó usuário  $i \in \bar{N}$  é igual a  $2d_i$  e o fluxo que chega no nó 0 (2.27) corresponde a capacidade residual da frota dos veículos. A restrição (2.29) define as arestas de uma solução factível e a restrição (2.30) força qualquer solução viável a conter duas arestas para cada cliente. O domínio das restrições é definido por (2.31) e (2.32).

#### 2.4.5 Formulação do PRV por Várias Mercadorias

A seguinte formulação foi retirada do trabalho de Baldacci, Hadjiconstantinou e Mingozzi (2004). Ela combina as restrições de designação para modelagem da rota e restrições de *várias mercadorias* para modelar o fluxo dos bens, e além disso ela introduz explicitamente a orientação dos arcos. Retomando as notações já definidas para o PRVC, além de também fazer uso de uma variável de fluxo  $f_{ij}^l$ , que especifica o montante de demanda destinada a um cliente  $l \in \bar{N}$  que é transportada pelo arco  $(i,j)$ .

$$\min \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (2.33)$$

s.t.:

$$\sum_{(0,j) \in A} x_{0j} = k \quad (2.34)$$

$$\sum_{(i,0) \in A} x_{i0} = k \quad (2.35)$$

$$\sum_{(i,j) \in A} x_{ij} = 1 \quad \forall i \in \bar{N} \quad (2.36)$$

$$\sum_{(i,j) \in A} x_{ij} = 1 \quad \forall j \in \bar{N} \quad (2.37)$$

$$\sum_{\substack{(i,j) \in A \\ i \neq l}} f_{ij}^l - \sum_{\substack{(j,i) \in A \\ j \neq l}} f_{ji}^l = \begin{cases} d_l, j = l & \forall l \in \bar{N} \\ 0, j \neq l & \forall j, l \in \bar{N} \\ -d_l, j = 0 & \forall l \in \bar{N} \end{cases} \quad (2.38)$$

$$f_{ij}^l \leq d_l x_{ij} \quad \forall l \in \bar{N}, \forall i, j \in N, i \neq j \quad (2.39)$$

$$\sum_{j \in \bar{N}} \sum_{l \in \bar{N}} f_{ij}^l \leq C - d_i \quad \forall i \in N \quad (2.40)$$

$$x_{ij} \in \{0,1\} \quad \forall (i,j) \in A \quad (2.41)$$

$$f_{ij}^l \geq 0 \quad \forall l \in \bar{N}, \forall (i,j) \in A \quad (2.42)$$

As restrições (2.36) e (2.37) garantem que um cliente é visitado exatamente uma vez. Já (2.34) e (2.35) impõe que cada veículo seja utilizado uma única vez. A restrição (2.38) garante que a demanda de cada nó seja satisfeita. Por fim, (2.39) mostra que só existe fluxo por uma aresta se ela faz parte da solução e (2.40) garante que a capacidade do veículo seja respeitada. Os custos são contabilizados pela soma da conexão das arestas (2.33) e o domínio das variáveis por (2.41) e (2.42).

### 2.4.6 Formulação do *PRV* por arcos capacitados

Esta formulação foi adaptada do trabalho de Pessoa, Uchoa e Poggi de Aragão (2009) onde este se propõe a resolver o *PRV* utilizando uma frota heterogênea de veículos. Esta generalização é muito importante na prática, visto que parte dos problemas de roteamento lida com uma frota heterogênea. Define-se  $q \in Q = \{0 \dots C\}$ .

$$\min \sum_{q \in Q} \sum_{(i,j) \in A} c_{ij} x_{ij}^q \quad (2.43)$$

s.t.:

$$\sum_{\substack{q \in Q \\ q \geq d_j}} \sum_{(i,j) \in A} x_{ij}^q = 1 \quad \forall j \in \bar{N} \quad (2.44)$$

$$\sum_{\substack{q \in Q \\ q \geq d_i}} \sum_{(j,i) \in A} x_{ji}^q = 1 \quad \forall j \in \bar{N} \quad (2.45)$$

$$\sum_{q \in Q} \sum_{(0,j) \in A} x_{0j}^q = k \quad (2.46)$$

$$\sum_{(i,n+1) \in A} x_{i,n+1}^0 = k \quad (2.47)$$

$$\sum_{(i,j) \in A} x_{ij}^q = \sum_{(j,i) \in A} x_{ji}^{q-d_j} \quad \forall j \in \bar{N} \text{ e } \forall q \geq d_j \quad (2.48)$$

$$\sum_{q \in Q} \sum_{(0,j) \in A} q x_{0j}^q = \sum_{i \in N} d_i \quad (2.49)$$

$$x_{ij}^q \in \{0,1\} \quad \forall (i,j) \in A, q \in Q \quad (2.50)$$

A função objetivo dada por (2.43) contabiliza o custo total de todas as conexões de acordo com o veículo escolhido. As restrições (2.44) e (2.45) definem o grau de conexão do nó. O número de rotas que deixam e retornam ao depósito é estabelecido por (2.46) e (2.47). A restrição (2.48) define que se um arco de índice  $q$  chega ao nó  $j$ , um arco de índice  $q - d_j$  deve deixar o nó  $j$ . Por fim, (2.49) garante que toda a demanda dos nós clientes seja atendida e (2.50) o domínio das restrições. Os autores também propuseram decompor pelo método de Dantzig e Wolfe (1960), além de propor diversos cortes para fortalecer os limites da formulação decomposta.

## 2.5 Método de Resolução do Problema

Para a resolução do problema proposto foi escolhida a abordagem *Column-and-cut*, que consiste em utilizar as técnicas de *Geração de Colunas* (Seção 2.5.1) e *Separação* (Seção 2.5.2). O uso destes métodos em conjunto busca acelerar a busca pela solução inteira ótima. O problema está relaxado e o algoritmo é um procedimento onde os limites inferiores do valor ótimo são computados pela *geração de colunas* e os *planos de corte* são iterativamente adicionados ao *Problema Mestre Reduzido (PMR)*, de forma a melhorar os limites obtidos. Se a solução é inteira, então o ótimo do *PM* foi encontrado, caso contrário, adiciona-se algumas desigualdades válidas que são violadas pela solução atual em busca da aproximação da *casca convexa* correspondente a solução ótima inteira.

A ideia de combinar somente a *Geração de Colunas e de Cortes* para melhorar os limites inferiores foi muito pouco usada. Isto porque, as variáveis duais associadas aos planos de separação podem ter um efeito indesejado no subproblema (FUKASAWA et al., 2006). Ao fim dos anos 90, alguns autores como Barnhart et al. (1998) e Van den Akker, Hurkens e Savelsbergh (2000) identificaram que os *cortes* expressados em termos das variáveis do problema original poderiam ser dinamicamente separados, modificados e adicionados ao *Problema Mestre*. Assim, estes cortes não alteram a estrutura do *subproblema*, sendo esta abordagem chamada de *robusta*, conforme descrito por Pessoa, Uchoa e Poggi de Aragão (2009).



### 2.5.1 Método de Geração de Colunas

O método de *Geração de Colunas (GC)* está relacionado com o *Método de Decomposição de Dantzig e Wolfe (1960)* e é utilizado para computar limites inferiores quando se resolve o chamado *Problema Mestre Restrito (PMR)*. A principal vantagem do método de *GC* baseia-se no princípio de que a maioria das variáveis do problema será *não-básica*, assim só é vantajoso explicitar aquelas que sejam mais interessantes para o problema. Em resumo, a cada iteração do *Método SIMPLEX* procura-se *precificar* variáveis *não-básicas* para introduzi-las na base. Para isso, as variáveis já explicitadas tem seus valores relaxados, ou seja, ao invés de terem valores inteiros, elas podem assumir valores contínuos dentro dos limites determinados para cada uma delas. À partir da resolução do *PMR* é possível obter os *preços duais* para cada uma das restrições do problema. Estas informações são então utilizadas na *função objetivo* do subproblema escolhido. Caso o valor da função objetivo seja negativo (para o caso da minimização), uma variável com custo reduzido negativo é identificada e em seguida é adicionada ao *PMR* e este é resolvido novamente.

Estes métodos tem sido amplamente aplicados a diversos problemas como o *Problema de Atribuição Generalizada* (Barnhart et al. (1998), Savelsbergh (1997)), *Problema de Programação da Tripulação* (Desrochers e Soumis (1989), Lavoie, Minoux e Odier (1988), Gamache et al. (1999)), *Problema de Programação de Máquinas Sequenciais e Paralelas* (Chen e Powell (1999), Van den Akker, Hurkens e Savelsbergh (2000), Demiriz, Bennett e Shawe-Taylor (2002)) e o *Problema de Roteamento de Veículos* em suas diversas abordagens (Desrochers, Desrosiers e Solomon (1992), Desrochers et al. (1988), Feillet et al. (2004)), entre muitos outros, como pode ser encontrado em Desaulniers, Desrosiers e Solomon (2005).

Dado o problema de otimização linear abaixo (2.51)-(2.53), este será chamado de *Problema Mestre*.

$$\min \sum_{j \in J} c_j \lambda_j \quad (2.51)$$

s.t.:

$$\sum_{j \in J} a_j \lambda_j \geq b \quad (\pi) \quad (2.52)$$

$$\lambda_j \geq 0 \quad \forall j \in J \quad (2.53)$$

Dado um vetor  $\pi$  de variáveis duais, deseja-se encontrar determinado  $j \in J$  que minimize  $\bar{c}_j := c_j - \pi^t a_j$ . Fica claro então, que explicitar todas estas variáveis pertencentes a um conjunto  $J$  de cardinalidade muito grande é muito caro computacionalmente. Dessa forma, trabalha-se com um conjunto menor  $J' \in J$  (*Problema Mestre Restrito - PMR*) de colunas. Assim, serão avaliados os custos reduzidos  $\bar{c}_j$  somente das variáveis enumeradas. Quando as colunas  $a_j, \forall j \in J$ , são dadas como elementos de um conjunto  $\mathcal{A}$ , e o coeficiente  $c_j$  pode ser computado de  $a_j$  por uma função de  $c$ , o subproblema adotado, terá a forma a seguir.

$$\bar{c}^* := \min\{c(a) - \pi^t a \mid a \in \mathcal{A}\} \quad (2.54)$$

Se  $\bar{c}^* \geq 0$  não existem mais custos reduzidos  $\bar{c}_j \forall j \in J$  negativos, e a solução  $\lambda$  para o *PMR* é ótima. Caso contrário, adiciona-se ao *PMR* a coluna, ou colunas, proveniente da solução do *subproblema de precificação* buscando a convergência. Cada coluna  $a \in \mathcal{A}$  é gerada uma única vez até que nenhuma variável que não esteja explícita no *PMR* tenha custo reduzido negativo. Quando o conjunto de colunas  $\mathcal{A}$  é finito, o procedimento de geração de colunas é exato.

Quando um limite superior  $\kappa \geq \sum_{j \in J} \lambda_j$  garante para a solução ótima do problema mestre que não terá um limite superior  $\bar{z}$  em  $z_{MP}^*$  a cada iteração, mas também um limite inferior que não pode reduzir  $\bar{z}$  mais do que  $\kappa$  vezes o menos custo reduzido  $\bar{c}^*$ :

$$\bar{z} + \kappa \bar{c}^* \leq z_{MP}^* \leq \bar{z} \quad (2.55)$$

Assim, pode-se verificar a qualidade da solução a qualquer momento. Na otimalidade de (2.51),  $\bar{c}^* = 0$  para as variáveis básicas e  $\bar{z} = z_{MP}^*$ .

A iteratividade do algoritmo de *GC* é apresentada na Figura 2.3 e o comportamento da estrutura do *PMR* com a adição de colunas na Figura 2.4.

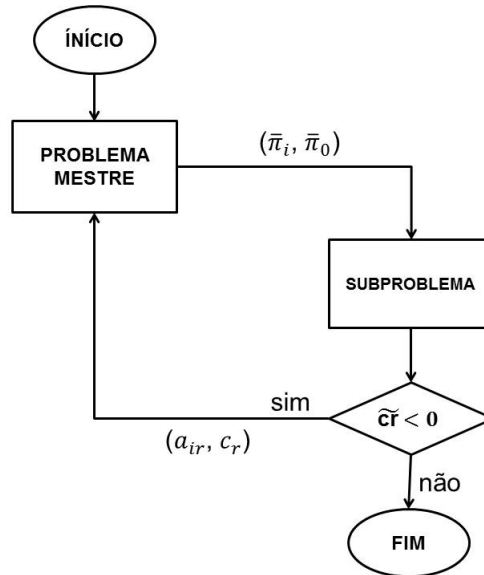


Figura 2.3: Iteratividade do algoritmo de Geração de Colunas. Elaboração própria.

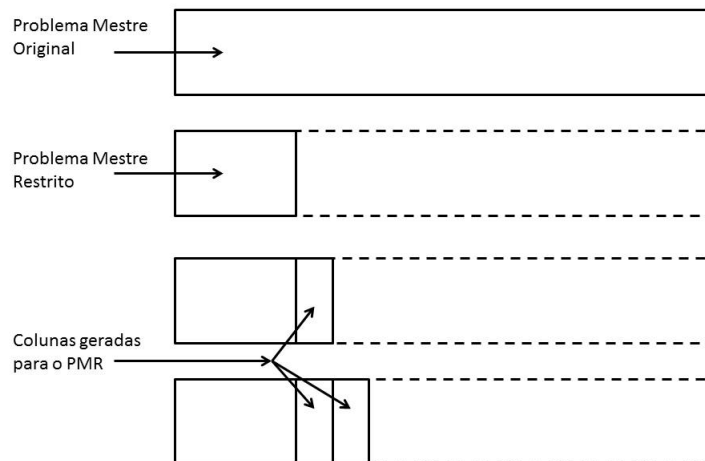


Figura 2.4: Adição de colunas ao *PMR*. Adaptado de Spoorendonk (2008)

## 2.5.2 Métodos de Separação

A ideia da inserção de *Planos de Corte*, também chamados de *Separação* ou *Geração de Cortes*, foi introduzida por Gomory (1958). O método de *Separação* é um método exato, para *Programação Linear*, que busca iterativamente refinar um conjunto viável por meio de inequações lineares, chamadas *cortes*. Tais procedimentos são utilizados para encontrar soluções inteiras para problemas de *Programação Linear Inteira (PPLI)* de forma a melhorar os limites e atingir a resposta ótima mais rápido, bem como para resolver problemas gerais de otimização convexa.

A proposta de Gomory (1958) era resolver um *PPLI* pela inserção de alguns *planos de corte*, assim reduzindo a solução de problemas inteiros à sucessão de problemas lineares. Estes cortes, também chamados de *cortes fracionários*, impõe a cada iteração do método *SIMPLEX* uma nova restrição baseada nos coeficientes das variáveis de folga cujo lado direito da respectiva linha não tem valor inteiro. Abaixo será apresentada não a formulação original proposta por Gomory (1958), mas uma revisada por Balas et al. (1996) e que obteve resultados coerentes em sua aplicação.

Seja o problema de programação linear inteira mista (MIP) abaixo:

$$\min \quad cx \quad (2.56)$$

s.t.:

$$Ax = b \quad (2.57)$$

$$x \geq 0 \quad (2.58)$$

$$x_i \in \{0,1\} \quad \forall i \in 1,\dots,p \quad (2.59)$$

Onde as primeiras  $p$  variáveis são binárias e as demais ( $i = p + 1, p + 2, \dots, n$ ) são contínuas. Denota-se por  $F_0, F_1 \subseteq \{1, \dots, p\}$  o conjunto de variáveis que foram fixadas em 0 ou 1, respectivamente, na árvore de busca. De forma geral, um corte gerado para um nó  $(F_0, F_1)$  da árvore de busca é válido somente para aqueles nós onde as variáveis  $F_0 \cup F_1$  permanecem com seus valores fixos. Quando os cortes podem ser gerados eles reduzem a necessidade de armazenar uma enorme quantidade de informação, além de que quando o corte é compartilhado ele melhora os limites para diversos nós da árvore. Um corte gerado para uma variável em  $(F_0, F_1)$  é válido para toda árvore de busca e computa os coeficientes para as variáveis  $x_j$  para  $j \in F_0 \cup F_1$ , sendo estes chamados de *Cortes de Elevação*, pois tornas o valor das variáveis mais próximos de valores inteiros.

Considerando então um PPL  $(F_0, F_1)$  em um nó genérico da árvore de busca e sua solução básica associada dada por:

$$x_i = \bar{a}_{i0} + \sum_{j \in J} \bar{a}_{ij}(-x_{ij}) \quad \forall i \in I. \quad (2.60)$$

$$x_k \geq 0 \quad \forall k \in I \cup J$$

$$x_k \leq 0 \quad \forall k \in F_0$$

$$x_k \geq 1 \quad \forall k \in F_1$$

Onde  $I$  e  $J$  são os conjuntos de variáveis *básicas* e *não-básicas* respectivamente. Assume-se sem perda de generalidade que  $F_0 \cup F_1 \subseteq J$ . Assim, em grande parte das implementações de *branch-and-cut* a correção das variáveis é realizada pela alteração dos limites inferior e superior e não pela eliminação das mesmas da formulação. Por padrão, adota-se a notação  $\lfloor \bar{a}_{ij} \rfloor$  e  $f_{ij}$  como sua parte inteira e a parte fracionária de  $\bar{a}_{ij}$ .

O trabalho de Spoorendonk (2008), explora o tópicos da construção e uso do método *Column-and-cut* para posterior combinação com o método de *branch-and-bound* (LAND; DOIG, 1960).

Mais detalhes podem ser encontrados na publicação de Desaulniers, Desrosiers e Spoorendonk (2011) que elucidam mais pontos do algoritmo *Branch-price-and-cut*, que combina *Geração de Colunas e Cortes* com *branch-and-bound*.

A inserção de diferentes famílias de cortes é feita através de um algoritmo que a cada iteração tem um conjunto de inequações geradas e adicionadas à formulação original ou problema mestre a ser novamente resolvido. Um fato interessante é que se pode decidir quantos cortes serão gerados a cada iteração. O trabalho de Balas, Ceria e Cornuéjols (1996) mostrou que a melhor aplicação para cortes em variáveis binárias é quando são realizados para todas elas que estão fracionárias, obtendo grande melhoria no valor da função objetivo. O esquema da adição de cortes ao problema original ou mestre é apresentado na Figura 2.5.

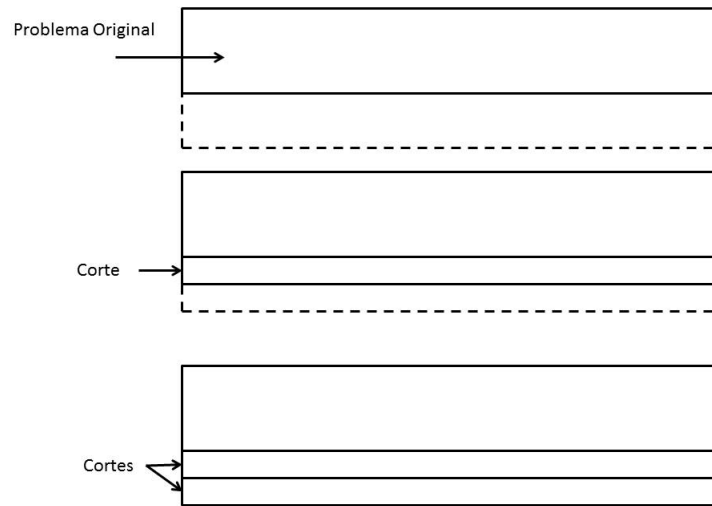


Figura 2.5: Adição de Cortes ao *Problema*. Elaboração própria.

## Capítulo 3

# Formulação do Problema

O *Problema de Roteamento de Veículos* tem como objetivo principal projetar um conjunto de  $k$  rotas para os veículos disponíveis de forma que estes visitem  $n$  clientes, tal que cada uma das rotas inicie e termine o trajeto no depósito, além de respeitar também a restrição de capacidade definida pelo problema. Nesta seção serão apresentadas a formulação utilizada pelo presente trabalho, além da estrutura adotada para o algoritmo *Column-and-cut*, sendo os *subproblemas* e os *cortes* usados, apresentados nas Seções 3.3 e 3.4, respectivamente.

### 3.1 Formulação do *PRV* por Particionamento de Conjunto

A ideia da utilização da formulação de *Particionamento de Conjunto* (*set partitioning*) junto com a *Geração de Colunas* (Seção 2.5.1) é trabalhar com um conjunto restrito de variáveis e ao fim do processo iterativo se aproximar ou até mesmo garantir a otimalidade, sem precisar explicitar todo o conjunto de rotas viáveis. Assim, a formulação proposta para o *PRV* por *Particionamento de Conjunto*, foi primeiramente estudada por Balinski e Quandt (1964) e posteriormente por diversos autores como Laporte (1992), Desrochers, Desrosiers e Solomon (1992), Letchford e Salazar-González (2006), Feillet et al. (2004), Baldacci, Hadjiconstantinou e Mingozzi (2004), Baldacci, Mingozzi e Roberti (2011), Contardo, Desaulniers e Lessard (2015). Como este trabalho utiliza uma *frota homogênea* de veículos, ele pode ser expresso pela formulação (3.1)-(3.4).

$$\min \sum_{r \in R} c_r \lambda_r \quad (3.1)$$

s.t.:

$$\sum_{r \in R} a_{ir} \lambda_r = 1 \quad \forall i \in \bar{N} \quad (\pi_i) \quad (3.2)$$

$$\sum_{r \in R} \lambda_r = k \quad (\pi_0) \quad (3.3)$$

$$\lambda_r \in \{0,1\} \quad \forall r \in R \quad (3.4)$$

Neste modelo de *particionamento de conjunto*, a variável  $\lambda_r$  representa a seleção ou não da rota  $r \in R$  para a solução. O conjunto de restrições representado por (3.2) garante que todos os clientes sejam atendidos, sendo elas conhecidas como *definidoras de faceta* ou *clique*. O parâmetro  $a_{ir}$  é binário e sinaliza se um cliente  $i$  participa ou não de uma dada rota  $r$ . Por fim, dado que o *PRV* necessita que  $k$  rotas sejam selecionadas, a restrição (3.4), chamada de *convexificação*, fica encarregada de garantir isto.

Este modelo decomposto contém um imenso número de variáveis possíveis. Seja  $|\bar{N}|$  o número de nós clientes, o número de rotas atingirá  $|\bar{N}|!$  em seu pior caso, onde todas são geradas. A

exemplo, para um caso muito simples de 10 nós, teriam-se 3 628 800 caminhos possíveis. Assim, para evitar esse problema, utiliza-se das técnicas de *Geração de Colunas* (Seção 2.5.1), onde as variáveis são adicionadas iterativamente.

Resolvendo o *PRV* utilizando *particionamento de conjunto*, formulação (3.1)-(3.4), e *geração de colunas* requer a admissão de colunas viáveis, além da relaxação da restrição de domínio (3.4), agora permitindo que  $0 \leq \lambda_r \leq 1, \forall r \in R$ . Essa viabilidade da rota será formalizada pelo *subproblema* de geração de rotas escolhido. Um primeiro ponto é a obtenção das variáveis duais ( $\pi_i$  e  $\pi_0$ ) provenientes das restrições (3.2) e (3.3), respectivamente. Os valores destas variáveis que ditarão a mudança nos custos do subproblema e gerarão rotas diferentes a cada iteração.

Para este trabalho, foram adotados então três *subproblemas*, conforme indicado em Toth e Vigo (2001) e Desaulniers, Desrosiers e Solomon (2005), para estudo sobre os limites obtidos, sendo eles todos baseados em *Programação Dinâmica*. O primeiro é o método de precificação simples, mais conhecido como *q-routes*, o segundo o *Problema do Caminho Mínimo com Restrição de Recursos* e o último o *Problema do Caminho Mínimo Elementar com Restrição de Recursos*. Todos eles serão devidamente apresentados nas Seções (3.3.1), (3.3.2) e (3.3.2.1), respectivamente, sendo que eles utilizam a mesma notação apresentada para os problemas da Seção (2.4).

## 3.2 Soluções Iniciais

O método de *Geração de Colunas* tem como característica não explicitar todo o conjunto de variáveis possíveis. Dessa forma para que ele possa iniciar seu processo iterativo é necessário que um conjunto de soluções viáveis iniciais seja provido. Desaulniers, Desrosiers e Solomon (2005) afirmam que a qualidade da solução inicial pode ser um fator decisivo no desempenho do método, auxiliando na convergência para a solução ótima, baseando-se nisso, foi utilizada uma *heurística* própria para geração de rotas, que tem seu funcionamento baseado na ideia de economia.

### 3.2.1 Heurística de Roteamento Econômico de Clarke-Wright

Um ponto muito importante é que para que a inicialização do algoritmo, é que uma solução viável inicial deve ser oferecida ao *PMR*. Para este trabalho foi utilizado a heurística proposta por Clarke e Wright (1964). Esse método foi criado com o intuito de fornecer soluções para o *PRV* baseando-se na ideia da economia ao se agrupar rotas. Assim, o algoritmo respeita todas as restrições do *PRVC*: (a) as rotas iniciam e terminam no depósito, (b) cada cliente é visitado uma única vez e suas necessidades atendidas, (c) a capacidade da rota não é violada.

Inicialmente, constrói-se uma tabela simétrica, onde o custo de ir de um nó  $i$  para outro nó  $j$  é o mesmo de ir de  $j$  para  $i$ . Um ponto notável é que, como o método é uma heurística, ele não garante a solução ótima do problema, mas pode se aproximar bastante desta. O conceito básico da economia (*saving*) expressa o custo obtido por se juntar duas ou mais rotas em uma única, conforme mostrado na Figura 3.1, onde o nó 0 representa o depósito.

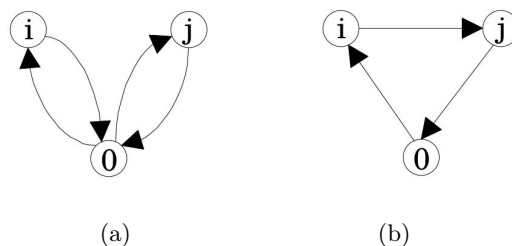


Figura 3.1: Ilustração do conceito de economia. Adaptado de Lysgaard (1997)

Conforme mostrado na Figura 3.1a clientes  $i$  e  $j$  são visitados em rotas distintas. Uma

alternativa seria visitar os dois clientes na mesma rota, por exemplo, na ordem  $i - j$ , como ilustrado na Figura 3.1b. Isto porque os custos de transporte são dados pela economia feita em se realizar a rota conforme Figura 3.1b ao invés de duas rotas distintas como na Figura 3.1a. Para explicitar então o quanto a junção de duas rotas oferece, usa-se a notação do custo de traslado entre  $i$  e  $j$  como  $c_{ij}$  e o custo total de transporte como  $T_a$  para a Figura 3.1a tem-se:

$$T_a = c_{0i} + c_{i,n+1} + c_{0j} + c_{j,n+1}$$

Equivalentemente, para a Figura 3.1b, tem-se:

$$T_b = c_{0i} + c_{ij} + c_{j,n+1}$$

Combinadas então as duas rotas, tem-se:

$$S_{ij} = T_a - T_b = c_{0i} + c_{i,n+1} + c_{0j} + c_{j,n+1} - (c_{0i} + c_{ij} + c_{j,n+1}) = c_{i,n+1} + c_{0j} - c_{ij} \quad (3.5)$$

Institivamente é notável que grandes valores de  $S_{ij}$ , Eq. (3.5), são atrativos, indicando que é interessante visitar os clientes  $i$  e  $j$  na mesma rota, sendo  $j$  visitado imediatamente após  $i$ . Existem duas versões do algoritmo, a *sequencial* e a *paralela*. Na *sequencial*, constrói-se exatamente uma rota por vez, enquanto que na *paralela*, mais de uma rota pode ser construída por vez.

Este trabalho, irá se ater somente a explicação do método *sequencial*, pois foi o utilizado. O primeiro passo então é calcular todas as economias, par a par, usando a Equação (3.5) e ordená-las decrescentemente. Junto com esse passo, pode-se gerar somente pares  $i - j$  que respeitem o limite de capacidade da rota. Em sequência, do topo desta lista ordenada de pares de clientes  $i - j$ , um par é considerado por vez. Escolhido o par, verifica-se se ele pode ser adicionado ou não a alguma rota já existente sem exceder a capacidade. Dado que a rota tem como primeiro nó visitado, após deixar o depósito, um nó  $k$  e o último, antes de atingir a cópia  $n + 1$ , um outro nó  $l$  (Figura 3.2a). Se  $i = l$ , introduz-se  $j$  entre  $l$  e  $n + 1$  (Figura 3.2b), se  $i = k$ ,  $j$  é introduzido entre 0 e  $k$  (Figura 3.2c). Existem também as possibilidades de  $j = l$  ou  $j = k$ . Para o primeiro caso (Figura 3.2d), introduz-se  $i$  antes de  $n + 1$  e na segunda alternativa,  $i$  é introduzido após 0 (Figura 3.2e).

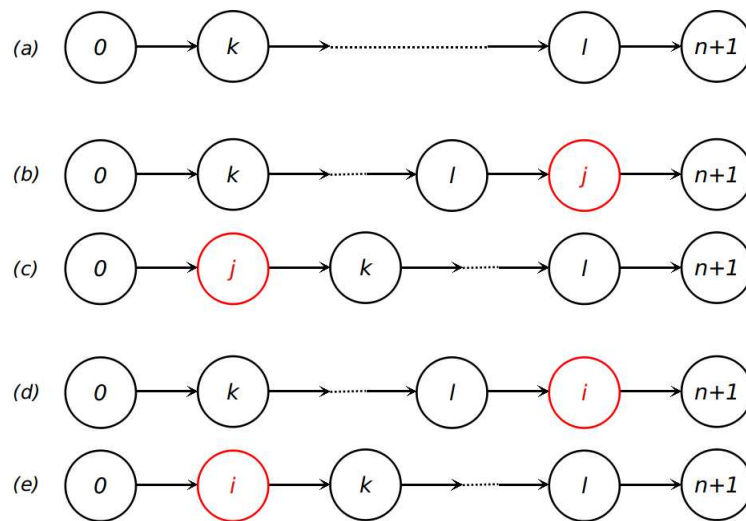


Figura 3.2: Elaboração das rotas usando Clarke e Wright (1964). Elaboração própria.

Este procedimento é repetido até que nenhum cliente possa mais ser adicionado à rota. Após

isso, exclui-se todas as *economias* dos nós que participam desta nota, retorna-se ao topo da lista tomando um novo par  $i - j$  e a elaborar uma nova rota é iniciada. Este procedimento é repetido até que não existam mais *economias* na lista. Todas estas rotas, com seus respectivos custos, são adicionadas ao *PMR* e caso um único cliente  $j$  não participe de nenhuma rota, uma coluna contendo somente ele e o custo  $c(0 - j - n + 1)$  também é adicionada. Um bom exemplo numérico pode ser encontrado em Lysgaard (1997).

### 3.3 Subproblemas de Precificação

O *Subproblema de Precificação* é executado sempre que o *PMR* é resolvido na otimalidade e ainda existam variáveis com potencial de contribuir para a melhora da solução, ou seja, o custo reduzido continua negativo. Um ponto importante na escolha dos subproblemas é quantidade de colunas que ele é capaz de entregar ao *PMR*. Fica claro então que a quantidade e qualidade das colunas que ditarão a velocidade de convergência do *PMR* para a solução  $\lambda^*$  ótima.

Neste trabalho foram testadas, além dos algoritmos de *PD*, as formulações exatas do *Problema do Caixeiro Viajante com Coleta de Prêmios (PCV-CP)* (BALAS, 1989) e a do *Problema do Caminho Mínimo Elementar com Restrição de Recursos (FPCMERC)* (FEILLET et al., 2004). Ambas só ajudaram a atingir a elementariedade para instâncias muito pequenas, 10 a 15 nós, não convergindo para instâncias consideradas grandes pela literatura, acima de 50 nós. A primeira, *PCV-CP*, mostrou maior velocidade de convergência do que a *FPCMERC*, porém em ambos os casos, a cada iteração que o subproblema era chamado, somente uma única coluna ótima era entregue ao *PMR*, sendo a razão da não convergência para a resposta ótima e nem para o os limites conhecidos das instâncias utilizadas. Assim, a principal vantagem da *Programação Dinâmica* sobre os métodos exatos citados é a entrega de uma quantidade maior de colunas ao *PMR*, o que torna mais rápida a explicitação da casca convexa do problema e a aproximação da resposta ótima.

Os custos utilizados para ditar a criação de rotas nos subproblemas não são os custos das rotas em si, mas sim o custo reduzido de uma rota, ou seja, o quão atrativa ele é para o problema. Este custo é dado pela Equação 3.6

$$\bar{c}_{ij} = \begin{cases} c_{ij} - \bar{\pi}_i, & \text{se } i \neq 0 \\ c_{ij}, & \text{caso contrário.} \end{cases} \quad (3.6)$$

#### 3.3.1 Subproblema de Precificação Simples (*q-routes*)

Um subproblema para o *PRVC* pode ser escrito em termos de *q-routes*, como proposto por Christofides, Mingozzi e Toth (1981a). Uma *q-route* é uma caminhada que inicia no nó depósito 0, visita uma sequência de clientes, onde a demanda máxima é menor ou igual a  $C$  e retorna a cópia do depósito  $n + 1$ . Um ponto interessante é que cada *q-route* define um conjunto de arcos indexados  $x_{ij}^d$ . Sendo que a sequência é iniciada com um arco  $(0, j)^D$ , onde  $j$  é o primeiro cliente visitado após o veículo deixar o depósito e  $D \leq C$  é a soma da demanda de todos os nós visitados. Caso um nó  $i$  seja visitado  $t$  vezes, então sua demanda  $d_i$  é contabilizada  $t$  vezes. Cada arco  $(i, j)^d$  é seguido por um arco  $(j, l)^{d-d_j}$ , onde  $l$  é o cliente visitado imediatamente após  $j$ . A sequência finaliza necessariamente com o arco  $(i, n + 1)^0$ , onde  $i$  é o último cliente visitado antes de retornar ao depósito  $n + 1$ . Um exemplo, se  $D = C = 7$  e as demandas dos vértices clientes são  $d_1 = d_3 = 1$ ,  $d_2 = 3$  e  $d_4 = 2$ . Uma *q-route não-elementar* que visitasse os vértices 1,2,3 na sequência 2,1,3,1,3, seria representada pelos seguintes arcos:  $(0,2)^7, (2,1)^4, (1,3)^3, (3,1)^2, (1,3)^1, (3,5)^0$  (em azul na Figura 3.3). Já uma *q-route elementar* que visitasse os vértices 1,2,3,4 na sequência 3,4,2,1, seria representada pelos seguintes arcos:  $(0,3)^7, (3,4)^6, (4,2)^4, (2,1)^1, (1,5)^0$  (em vermelho na Figura 3.3).



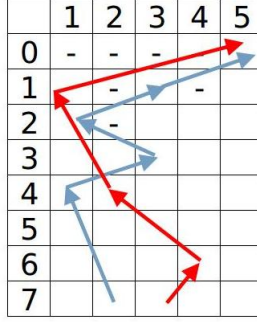


Figura 3.3: Ilustração da recuperação de  $q$ -routes na matriz  $R$ . Elaboração própria.

O custo reduzido de uma variável  $\lambda_r$  do  $PMR$  pode ser definido como a soma dos custos reduzidos dos arcos  $(i,j)^d$  que compõem a respectiva  $q$ -route e são calculados utilizando a equação de recursão (3.7). Assim, o subproblema consiste em encontrar  $q$ -routes que levam a variáveis com custo reduzido mínimo, sendo  $NP$ -Difícil, mas que pode ser resolvido em tempo pseudo-polinomial de  $O(nC)$ , conforme descrito por Pessoa, Uchoa e Poggi de Aragão (2009). Na matriz  $R$  existem  $(n+1) \times (C+1)$  entradas e o custo de cada arco é acessado uma única vez. Para cada arco acessado, o procedimento gasta  $O(1)$  para atualizar a matriz, então o tempo de preenchimento total é  $O(nC)$ .

Seja a matriz  $R$ , tal que a posição  $R(d,i)$  representa o *custo reduzido mínimo parcial* da  $q$ -route que inicia no vértice  $i \in \bar{N}$  com demanda total igual a  $d$  e termina exatamente em  $n+1$ . Os custos dos arcos que deixam o depósito  $(c_{0,j})$  não são considerados em  $R$ . Cada entrada  $R(d,i)$  contém um *rótulo* que consiste de cinco informações: o próprio nó  $i$ , o custo reduzido da  $q$ -route,  $j$  que consiste do próximo nó visitado,  $d$  que indica o índice que se chega ao nó  $j$  e  $U \subseteq \bar{N}$ , que armazena os nós que participaram do caminho  $i$  até  $n+1$ , denotados por  $i(R(d,i))$ ,  $\bar{c}r(R(d,i))$ ,  $j(R(d,i))$ ,  $d(R(d,i))$  e  $U(R(d,i))$ , respectivamente. O subproblema de  $q$ -routes se dá em três etapas: (a) Inicialização, (b) Preenchimento e (c) Recuperação (Ilustradas pelos Algoritmos 1, 2 e 3, respectivamente).

Na fase de *Inicialização*, todas as entradas  $R(d_i,i)$  são inicializadas com rótulos correspondentes a um único arco  $(i, n+1)^0$  com destino ao depósito,  $\forall i \in \bar{N}$ . Todas as entradas  $R(d,i)$  tal que  $d < d_i$  também são inicializadas com rótulos contendo custos infinitos. Isso porque só se pode visitar um cliente com capacidade disponível  $d \geq d_i$ , caso contrário a visita não é viável. A fase de *Preenchimento* usará o seguinte procedimento de recursividade dinâmica:

$$\bar{c}r(R(d,i)) = \min_{(i,j)^d \in \delta(i)^+} \{\bar{c}_{ij} + \bar{c}r(R(d-d_i,j))\} \quad (3.7)$$

Por fim, tem-se para cada posição da matriz o custo reduzido mínimo da  $q$ -routes com demanda exata  $d$ , que inicia em cada cliente  $i \in \bar{N}$ , para  $d = d_i, \dots, C$ . Assim, na etapa de *Recuperação*, para cada cliente  $i \in \bar{N}$  escolhe-se a  $q$ -route de demanda  $d^*$  de menor custo reduzido  $(c_{0,i} + \bar{c}r(R(d^*,i)))$  para se inserir em (3.1)-(3.4). A formação de ciclos é comum neste método. Para evitar isso, diversas técnicas podem ser aplicadas, na Seção 3.5 é apresentada com detalhes a utilizada por este trabalho.

---

**Algoritmo 1:** Etapa de Inicialização para o método das  $q$ -routes

---

```
1 Grafo completo  $G = (N, A)$ , nó origem 0 e nó destino  $n + 1$ 
2 para todo  $d = 0, \dots, C$  faça
3     para todo  $i \in \bar{N} \cup \{n + 1\}$  faça
4          $i(R(d, i)) \leftarrow i$ 
5         se  $d = d_i$  então
6             se  $i \neq n + 1$  então
7                  $\bar{c}r(R(d, i)) \leftarrow \bar{c}_{i, n+1}$ 
8             senão
9                  $\bar{c}r(R(d, i)) \leftarrow 0$ 
10            fim se
11             $j(R(d, i)) \leftarrow n + 1$ 
12             $d(R(d, i)) \leftarrow d - d_i$ 
13        senão se  $d < d_i$  então
14             $\bar{c}r(R(d, i)) \leftarrow +\infty$ 
15             $j(R(d, i)) \leftarrow -1$ 
16             $d(R(d, i)) \leftarrow -1$ 
17        senão se  $d > d_i$  e  $i = n + 1$  então
18            fim se
19    fim para todo
20 fim para todo
```

---

**Algoritmo 2:** Etapa de Preenchimento para o método das  $q$ -routes

---

```
1 para todo  $d = 1, \dots, C$  faça
2     para todo  $i \in \bar{N}$  faça
3         se  $d > d_i$  então
4              $\tilde{c} \leftarrow +\infty$ 
5              $\tilde{j} \leftarrow -1$ 
6              $\tilde{d} \leftarrow -1$ 
7             para todo  $(i, j) \in A$  faça
8                  $\bar{r}c \leftarrow \bar{c}_{ij} + \bar{c}r(R(d - d_i, j))$ 
9                 se  $\bar{r}c < \tilde{c}$  então
10                     $\tilde{c} \leftarrow \bar{r}c$ 
11                     $\tilde{j} \leftarrow j$ 
12                     $\tilde{d} \leftarrow d - d_i$ 
13            fim se
14        fim para todo
15         $\bar{c}r(R(d, i)) \leftarrow \tilde{c}$ 
16         $j(R(d, i)) \leftarrow \tilde{j}$ 
17         $d(R(d, i)) \leftarrow \tilde{d}$ 
18    fim se
19    fim para todo
20 fim para todo
```

---

---

**Algoritmo 3:** Etapa de Recuperação para o método das  $q$ -routes

---

```
1 para todo  $i \in \bar{N}$  faça
2     vetor  $a_{ir} \leftarrow 1$ 
3      $\tilde{c} \leftarrow +\infty$ 
4      $\tilde{d}^* \leftarrow -1$ 
5     para todo  $d = d_i, \dots, C$  faça
6         se  $\overline{cr}(R(d,i)) < \tilde{c}$  então
7              $\tilde{c} \leftarrow \overline{cr}(R(d,i))$ 
8              $\tilde{d}^* \leftarrow d$ 
9         fim se
10    fim para todo
11     $\tilde{j} \leftarrow j(R(\tilde{d}^*, i))$ 
12     $\tilde{d} \leftarrow d(R(\tilde{d}^*, i))$ 
13    vetor  $a_{jr} \leftarrow 1$ 
14     $route_{cost} \leftarrow c_{0,i} + c_{ij}$ 
15    enquanto  $\tilde{j} \neq n + 1$  faça
16         $route_{cost} \leftarrow c_{\tilde{j},j}^{(R(\tilde{d}, \tilde{j}))}$ 
17         $nj \leftarrow j(R(\tilde{d}, \tilde{j}))$ 
18         $nd \leftarrow d(R(\tilde{d}, \tilde{j}))$ 
19         $\tilde{j} \leftarrow nj$ 
20         $\tilde{d} \leftarrow nd$ 
21    vetor  $a_{\tilde{j}r} \leftarrow a_{jr} + 1$ 
22    fim enquanto
23    adicione a coluna  $a_r$  de custo  $route_{cost}$  ao  $PMR$ 
24 fim para todo
```

---

### 3.3.2 O Problema de Caminho Mínimo com Restrição de Recursos

O *Problema do Caminho Mínimo com Restrição de Recursos (SPPRC - Shortest Path Problem with Resources Constraints)* é uma variação do *Problema do Caminho Mínimo*. Ele foi introduzido por Desrochers (1986) em sua tese de pós-doutorado como um subproblema para o Problema de Programação de Tripulação de Ônibus (*Bus Driver Scheduling Problem*). Ele consiste em encontrar o menor caminho (mais barato, menor distância ou menor tempo) entre todos os caminhos que iniciam em um nó origem e terminam em um nó destino, satisfazendo uma série de restrições definidas sobre um conjunto de recursos. Um determinado recurso, corresponde a uma quantidade, tal como tempo, carga, duração de um intervalo de jornada, que varia ao longo do caminho. O *SPPRC* pode ser classificado de acordo com (a) a forma como os recursos são acumulados, levando a diferentes definições de caminhos viáveis, (b) o objetivo, (c) a rede em que ele é descrito e (d) a existência de restrições de estruturas adicionais que excluem determinado tipo de caminho (elementar ou não-elementar). Este último caso o chamado *Problema do Caminho Mínimo Elementar com Restrição de Recursos* é apresentado na seção 3.3.2.1. Para a resolução do *SPPRC*, *elementar* ou não, é utilizado um algoritmo de *Programação Dinâmica* baseado nos conceitos de rotulamento e dominância e será apresentado na Seção 3.3.2.2.

#### 3.3.2.1 O Caminho Mínimo *Elementar* com Restrição de Recursos

O *Problema do Caminho Mínimo Elementar com Restrições de Recursos (Elementary Shortest Path Problem with Resource Constraints - ESPPRC)* é uma variação do *Problema do Caminho Mínimo com Restrição de Recursos*, também denominado *não-elementar*. A diferença entre eles é que o *ESPPRC* não aceita ciclos, ou seja, se um nó  $i \in \bar{N}$  é visitado por um caminho, isto só é feito uma única vez. A principal vantagem deste subproblema sobre os demais é que a estrutura dele sempre oferece colunas viáveis ao problema mestre sem a necessidade de usar algum outro método que evite ou proíba ciclos.

#### 3.3.2.2 Algoritmo para o *SPPRC Elementar e Não-elementar*

O algoritmo utilizado foi uma adaptação do proposto por Feillet et al. (2004), sendo que para o caso do *SPPRC*, a *elementariedade* dos caminhos formados está *relaxada*, ou seja, um nó só não é visitado, se a adição dele ao caminho formar um rótulo inviável. A inviabilidade de um rótulo *não-elementar* só acontecerá pela extrapolação da capacidade  $C$  de uma rota. Este algoritmo é uma extensão do *Ford-Bellman*, levando a restrição de recursos em conta. Cada nó recebe rótulos ao longo do algoritmo, que representam caminhos parciais e indicam o custo e o consumo de recursos destes caminhos. Os nós são iterativamente tratados até que nenhum novo rótulo seja criado. Dessa forma, quando um nó é tratado seus rótulos são estendidos a todos os nós sucessores. Para limitar a proliferação de rótulos, regras de dominância são definidas. A definição de algumas notações se faz necessária. Cada caminho  $\rho_{0i}$  da origem 0 até o nó  $i$  está associado a um estado  $(T_i^1, \dots, T_i^\Gamma)$  correspondente a quantidade de cada um dos  $\Gamma$  recursos utilizados pelo caminho e também o custo  $C(T_i^1, \dots, T_i^\Gamma)$ . Simplificando, o rótulo  $(R_i, C_i)$  é definido para cada caminho  $\rho_{0i}$ , onde  $R_i = (T_i^1, \dots, T_i^\Gamma)$  e  $C_i = C(T_i^1, \dots, T_i^\Gamma)$ . Abaixo são definidas algumas notações necessárias.

- $P_i$ : Lista dos rótulos (caminhos) *não-dominados* até o nó  $i$ .
- $\rho_i$ : Rótulo (caminho) que sai do nó 0 e chega ao nó  $i$ .
- $Succ(i)$ : Conjunto de nós sucessores do nó  $i$ .
- $W$ : Lista de nós esperando para serem tratados.
- $Extend(\rho_i, j)$ : Função que retorna o rótulo resultante da extensão do rótulo  $\rho_i \in P_i$  ao nó  $j$  quando a extensão é viável e nada, caso contrário.

- $F_{ij}$ : Conjunto de rótulos estendidos de  $\rho_i$  a  $j$ .
- $EFF(P_i)$ : Procedimento que mantém somente rótulos não dominados na lista  $P_i$
- $V_\rho^j$ : Vetor que sinaliza se o nó  $j$  já foi visitado pelo caminho  $\rho$  ou que a extensão a ele é inviável.
- $T_i^\gamma$ : representa o consumo de cada um dos recursos  $\gamma = 1, \dots, \Gamma$  até o nó  $i$ .

---

**Algoritmo 4:** Problema do Caminho Mínimo (*Elementar*) com Restrição de Recursos

---

```

1 INICIALIZAÇÃO
2  $P_0 \leftarrow \{(0, \dots, 0)\}$ 
3 para todo  $i \in \bar{N}$ 
4     faça  $P_i \leftarrow \{\emptyset\}$ 
5  $W \leftarrow W \cup \{0\}$ 
6 repita
7     EXPLORAÇÃO DOS SUCESSORES DE UM NÓ
8     escolha  $i \in W$ 
9     para todo  $j \in Succ(i)$ 
10        faça  $F_{ij} \leftarrow \{\emptyset\}$ 
11        para todo  $\rho_i = (T_i^1, \dots, T_i^\Gamma, s_i, V_i^1, \dots, V_i^n, C_i) \in P_i$ 
12            faça se  $V_i^j = 0$ 
13                então  $F_{ij} \leftarrow F_{ij} \cup \{Extend(\rho_i, j)\}$ 
14        fim para todo
15         $P_j \leftarrow EFF(F_{ij} \cup P_j)$ 
16        se  $P_j$  mudou
17            então  $W \leftarrow W \cup \{j\}$ 
18    fim para todo
19    REDUÇÃO DE  $W$ 
20     $W \leftarrow W \setminus \{i\}$ 
21 até que  $W = \{\emptyset\}$ 

```

---

As linhas 1 a 5 realizam a inicialização do algoritmo, onde o nó *origem* passa a ser primeiro a ser tratado e todos os nós  $i \in \bar{N}$  tem seus rótulos inicializados como vazios. Entre as linhas 6 e 21, tem-se o processo de iteração, até que não exista mais nós a serem rotulados. Assim, a cada iteração um nó é escolhido e para cada um dos seus sucessores verifica-se a viabilidade de extensão e se, estas não forem dominadas, estes novos rótulos serão explorados nas próximas iterações. É interessante de se notar que sem o teste de dominância da linha 15 do Algoritmo 4 todos caminhos viáveis seriam enumerados, ou seja, teria-se uma abordagem de força bruta, o que não é computacionalmente interessante. Os caminhos não promissores são eliminados através dos *critérios de dominância* adotados, conforme apresentado na Seção (3.3.2.3). Para ambos os casos, *elementar ou não*, todas as rótulos gerados para o nó  $n + 1$  foram adicionados ao *PMR*.

### 3.3.2.3 Dominância e extensão nos algoritmos *SPPRC*

Para se obter soluções ótimas para o *SPPRC*, existe a necessidade de se considerar somente os rótulos (caminhos) *não-dominados* de 0 até determinado nó cliente  $i$ . Um rótulo *não-dominado* é um caminho que visita um subconjunto de clientes  $j \in \bar{N}$  e  $P = (0, j_1, j_2, \dots, j_{k-1}, j_k = i)$ , de forma que, se pelo menos um dos recursos ou atributos associados a este caminho, quando comparados pela política de dominância, for melhor do que algum dos atuais rótulos *não-dominados*, este novo rótulo também será considerado *não-dominado*. Além disto, uma das falhas deste algoritmo é a dificuldade em convergir quando se tem um número reduzido de recursos e além disso cada um destes recursos não tem seus limites de consumo muito estreitos (CONTARDO;

DESAULNIERS; LESSARD, 2015). Neste trabalho isso acontece nos casos onde a carga  $q$  acumulada até o nó  $i$  é muito menor do que a capacidade  $C$  da rota, isso faz com que a *fronteira Pareto* de cada nó cresça muito, tornando inviável o tratamento de todos os rótulos gerados.

Os critérios de dominância e extensão foram baseado nos trabalhos de Dumas, Desrosiers e Soumis (1991), Feillet et al. (2004), Desaulniers, Desrosiers e Solomon (2005) e Ropke e Cordeau (2009). Estes autores utilizaram diversos elementos para auxiliar no critério de dominância, como custo, carga acumulada, proximidade com a abertura da janela de tempo para entrega, composição da rota (clientes que dela participam), composição dos cortes, entre outros. Assim cada rótulo  $\rho_{0i}$  contém as seguintes informações, usadas no critério de dominância:

- $\rho(i)$ :  $i$ , último nó visitado pelo caminho  $\rho_{0i}$ ;
- $\bar{c}(\rho)$ : custo reduzido associado ao rótulo  $\rho_{0i}$ ;
- $s_\rho$ : total de nós que são inviáveis para extensão pelo caminho  $\rho_{0i}$ ;
- $q_\rho$ : carga acumulada pelo caminho  $\rho_{0i}$ ;
- $V_\rho^k$ : vetor binário que indica quais nós  $k \in \bar{N} \cup \{n+1\}$  que são inviáveis partindo de  $\rho(i)$ , inclusive  $i$ ;
- $U_\rho$ : conjunto de nós clientes que participam do caminho  $\rho_{0i}$ , inclusive  $i$ .

A regra de dominância trabalha como segue. Seja  $\rho'_{0i}$  e  $\rho^*_{0i}$  dois caminhos distintos de 0 a  $i$  cada um associado com os rótulos  $(R'_i, C'_i)$  e  $(R^*_i, C^*_i)$ , respectivamente.  $\rho'_{0i}$  dominará  $\rho^*_{0i}$  ( $\rho'_{0i} \preceq \rho^*_{0i}$ ) se e somente se  $C'_i \leq C^*_i$  e  $T_i'^\gamma \leq T_i^{*\gamma}$  para todo recurso  $\gamma = 1, \dots, \Gamma$  e  $(R'_i, C'_i) \neq (R^*_i, C^*_i)$ , ou seja, se e somente se todas as condições abaixo forem satisfeitas:

- i.  $\bar{c}(\rho')_i \leq \bar{c}(\rho^*)_i$ ;
- ii.  $s'_\rho \leq s^*_\rho$ ;
- iii.  $q'_\rho \leq q^*_\rho$ ;
- iv.  $V_\rho^{k'} \leq V_\rho^{k^*}, \forall k = 1, \dots, n$ .

Existem ainda outros dois casos a se explorar a dominância. O primeiro em que, para algum dos recursos analisados,  $\rho'_{0i}$  é melhor do que  $\rho^*_{0i}$ , assim considera-se que os rótulos são *não-dominados* entre si e ambos irão participar da *Fronteira Pareto* do nó  $i$ . O segundo, quando o rótulo  $\rho'_{0i}$  não consegue ser melhor do que  $\rho^*_{0i}$  em nenhum dos recursos, assim,  $\rho'_{0i}$  é descartado. Um fato interessante a ser notado é que quanto mais recursos adicionados, mais rápida será a convergência e mais *afiada* a política de dominância será (FEILLET et al., 2004).

Após a verificação de que o rótulo é *não-dominado*, ele será *extendido* (linha 13) para todos nós considerados viáveis. Estender um rótulo  $\rho_{0i}$ , ou caminho, através de um arco  $(\rho(i), j)$  resulta em rótulo  $\rho_{0j}$ , de predecessor  $\rho_{0i}$ , custo  $c(\rho)_i + \bar{c}_{\rho(i)j}$ , conjunto  $U_\rho^i \cup \{j\}$  e carga  $q_\rho^i + d_j$ . Todos aqueles nós  $k \in \bar{N}$  que se tornarem inviáveis para uma futura extensão serão sinalizados em  $s_\rho^j$  e  $V_\rho^j$ . Este novo valor para os recursos utilizados é então testado dentro dos seus limites, inferior e superior, e aceito somente se os respeita. Para este trabalho, o recurso utilizado é a carga transportada pela rota e seu limite superior  $\bar{q}_\rho = Q$ . Um arco  $(\rho(i), j)$  só é considerado viável, se a carga acumulada do caminho  $\rho_{0i}$  mais a carga  $d_j$  que será acrescentada ao se visitar  $j$  seja menor do que  $Q$ .

Para facilitar o entendimento sobre o rotulamento do algoritmo *SPPRC*, um exemplo será apresentado usando a Figura 3.4. Um rótulo é representado por  $\rho_{0i}^p = [q, i_0, i_1, i_2, i_3, s, rc, p]$ . As suas componentes são,  $q$  (carga acumulada),  $i_0$  a  $i_3$  sinalizam a presença do nó  $i$  no caminho,  $s$  o total de nós inviáveis de extensão,  $rc$  o custo reduzido e  $p$  o rótulo predecessor.

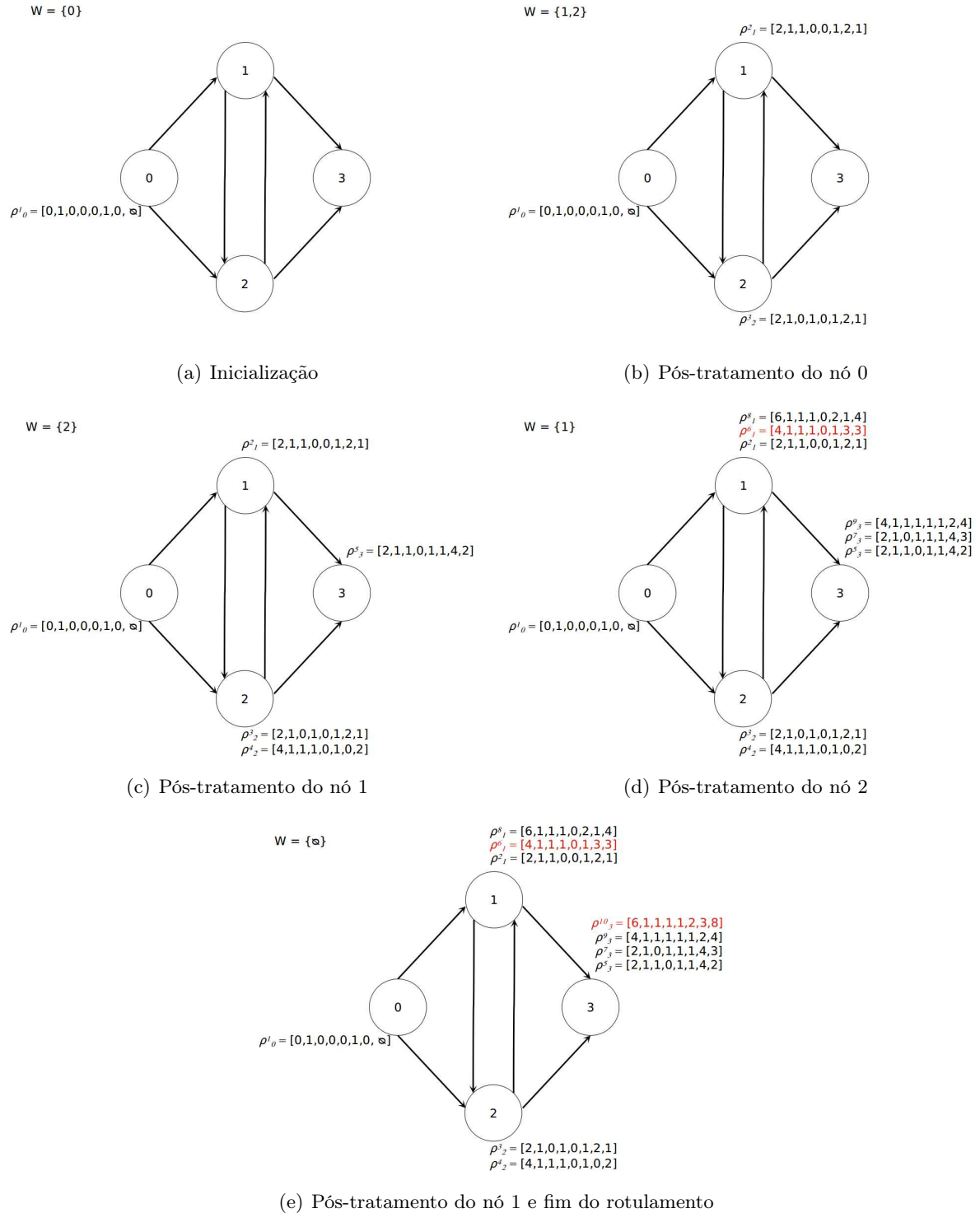


Figura 3.4: Ilustração do algoritmo *SPPRC*. Elaboração própria.

A rede tem 4 nós, sendo que 0 e 3 representam o depósito e sua cópia, respectivamente. O índice  $p$  no rótulo  $\rho_{0i}^p$  representa o número dado a ele, tornando-o único e de fácil recuperação. O índice  $i$ , sinaliza o último nó visitado pelo caminho. Os valores adotados para esta rede foram, capacidade  $Q = 6$ , demandas  $q_0 = q_3 = 0$  e  $q_1 = q_2 = 2$ , custos  $c_{01} = c_{02} = c_{13} = c_{23} = 2$ ,  $c_{12} = -2$ ,  $c_{21} = 1$ .

A Figura 3.4a mostra a *Inicialização*, onde o nó 0 recebe o rótulo  $\rho_0^1 = [0, 1, 0, 0, 0, 1, 1, \emptyset]$ . A lista  $W = \{\emptyset\}$ , de nós esperando para ser tratados, recebe  $W \leftarrow W \cup \{0\}$ .

O rótulo  $\rho_0^1$  é estendido para os nós sucessores 1 e 2, gerando os rótulos  $\rho_1^2$  e  $\rho_2^3$ . Assim,  $W \leftarrow W \setminus \{0\}$  e depois  $W \leftarrow W \cup \{1, 2\}$  (Figura 3.4b).

O nó 1 será tratado, sendo seu rótulo  $\rho_1^2$  estendido para os nós 2 e 3, gerando os novos rótulos  $\rho_2^4$  e  $\rho_3^5$ . A lista é atualizada  $W \leftarrow W \setminus \{1\}$  (Figura 3.4c).

O nó 2 tem seus rótulos  $\rho_2^3$  e  $\rho_2^4$  estendidos, tanto para o nó 1 quanto para o nó 3. Os rótulos gerados são  $\rho_1^6$  e  $\rho_1^8$ , para o nó 1 e,  $\rho_3^7$  e  $\rho_3^9$ , para o nó 3. O rótulo  $\rho_1^6$  é *dominado* pelo rótulo  $\rho_1^2$ , pois ele tem valores piores (maiores) para a carga, número de nós visitados e o custo, não oferecendo nenhuma vantagem em ser mantido para uma extensão futura. O rótulo  $\rho_1^8$ , terá sinalizado que  $s = 2$ , pois o limite da rota  $Q = 6$  foi atingido, sendo que não é mais possível visitar o nó 2 e a única extensão viável agora é para o depósito  $n + 1$ , pois  $q_{n+1} = 0$ . Os rótulos  $\rho_3^7$ ,  $\rho_1^8$  e  $\rho_3^9$  são *não-dominados* entre si e serão mantidos (Figura 3.4d).

Resta agora a extensão de  $\rho_1^8$ . Este rótulo só pode ser estendido ao nó 3, gerando o rótulo  $\rho_3^{10}$ . Este rótulo é dominado por  $\rho_3^9$ , pois tem sua carga, número de nós inviáveis e custo maiores que os encontrados em  $\rho_3^9$  (Figura 3.4e).

O rotulamento no algoritmo *ESPPRC* se dá de uma maneira similar ao *SPPRC*, com a principal alteração de que um cliente não pode ser visitado mais de uma vez durante a extensão dos rótulos. Neste exemplo um rótulo é representado por  $\rho_{0i}^p = [rc, i_0, i_1, i_2, i_3, q]$ . As suas componentes são,  $rc$  o custo,  $i_0$  a  $i_3$  sinalizam a presença do nó  $i$  no caminho e  $q$  a carga acumulada. Os elementos  $s$  e  $p$  foram omitidos por razões de tornar mais simples a explicação. A Figura 3.5 ilustra o processo de *rotulagem* e *extensão*.



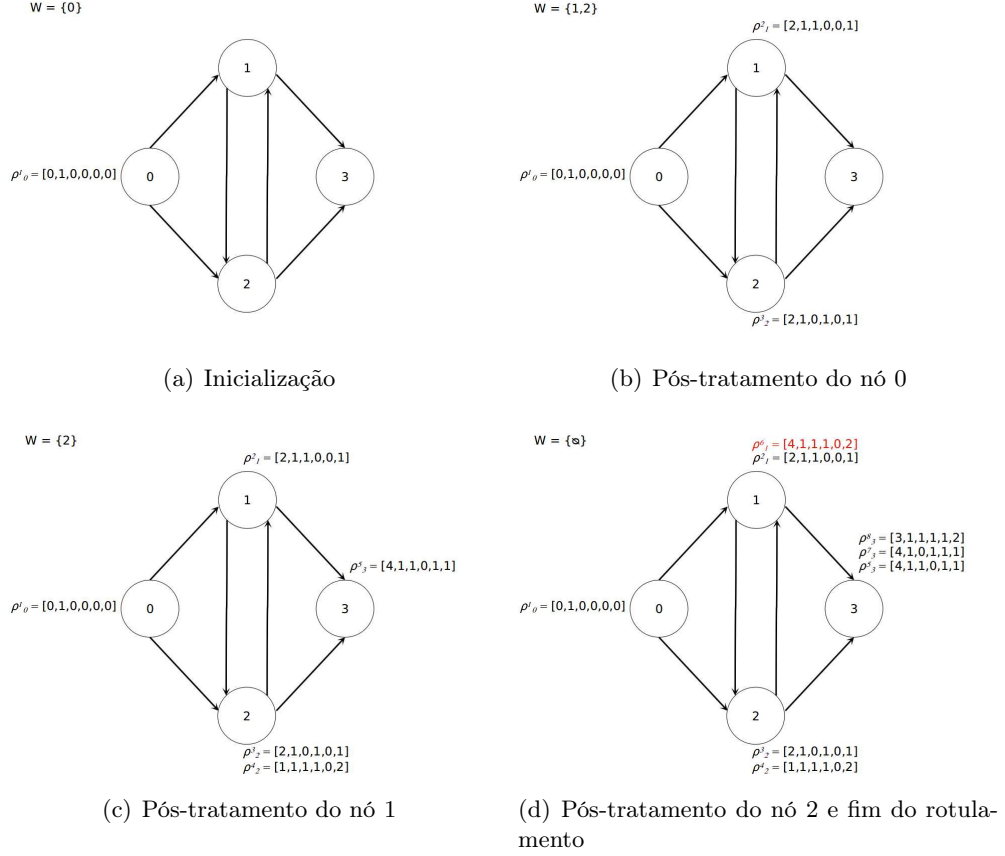


Figura 3.5: Ilustração do algoritmo *ESPPRC*. Elaboração própria.

A rede tem 4 nós, sendo que 0 e 3 representam o depósito e sua cópia, respectivamente. O índice  $p$  no rótulo  $\rho_{0i}^p$  representa o número dado a ele, tornando-o único e de fácil recuperação. O índice  $i$ , sinaliza o último nó visitado pelo caminho. Os valores adotados para esta rede foram, capacidade  $Q = 4$ , demandas  $q_0 = q_3 = 0$  e  $q_1 = q_2 = 1$ , custos  $c_{01} = c_{02} = c_{13} = c_{23} = 2$ ,  $c_{12} = -1$ ,  $c_{21} = 1$ .

A Figura 3.5a mostra a *Inicialização*, onde o nó 0 recebe o rótulo  $\rho_0^1 = [0,1,0,0,0,0]$ . A lista  $W = \{\emptyset\}$ , de nós esperando para ser tratados, recebe  $W \leftarrow W \cup \{0\}$ .

O rótulo  $\rho_0^1$  é estendido para os nós sucessores 1 e 2, gerando os rótulos  $\rho_1^2$  e  $\rho_2^3$ . Assim,  $W \leftarrow W \setminus \{0\}$  e depois  $W \leftarrow W \cup \{1,2\}$  (Figura 3.5b).

O nó 1 será tratado, sendo seu rótulo  $\rho_1^2$  estendido para os nós 2 e 3, gerando os novos rótulos  $\rho_2^4$  e  $\rho_3^5$ . O rótulo  $\rho_2^4$  não é dominado por  $\rho_2^3$ , pois tem o custo menor. A lista é atualizada  $W \leftarrow W \setminus \{1\}$  (Figura 3.5c).

Na Figura 3.5d, o nó 2 tem seu rótulo  $\rho_2^3$  estendido primeiramente para o nó 1, gerando o rótulo  $\rho_1^6$  que é dominado pelo rótulo  $\rho_1^2$ , já que tem maior número de nós visitados, maior custo e maior carga. Não oferecendo vantagem em uma futura extensão. Agora o rótulo  $\rho_2^3$  é estendido para 3 gerando o rótulo  $\rho_3^7$ , que não é dominado pelo rótulo  $\rho_3^5$ . O rótulo  $\rho_2^4$  não pode ser estendido ao nó 1, visto que este já foi visitado, restando assim ser estendido para o nó 3. O rótulo  $\rho_3^8$  é gerado desta extensão, mas é não é dominado por  $\rho_3^5$  e nem por  $\rho_3^7$ , pois tem seu custo menor do que de ambos.

### 3.3.2.4 Introdução do conceito *ng-SPPRC*

O conceito de caminho-*ng* foi introduzido por Baldacci, Mingozzi e Roberti (2011) e consiste em um caminho progressivo  $P = (0, i_1, i_2, \dots, i_{k-1}, i_k = i)$  não necessariamente elementar partindo do depósito 0 com carga  $q = 0$ , visitando os clientes  $V(P) = \{0, i_1, i_2, \dots, i_{k-1}, i_k\}$  respeitando a capacidade  $C$  da rota e finalizando no cliente  $i = i_k = \sigma(P)$  com demanda  $0 \leq q \leq C$ .  $A(P)$  denota o conjunto de arcos utilizados pelo caminho  $P$  e por  $c(P) = \sum_{(i,j) \in A(P)} \bar{c}_{ij}$  o custo do caminho  $P$ . Este caminho pode conter ciclos, de acordo com de vizinhanças  $V_i \subseteq \bar{N}, \forall i \in \bar{N}$ , sendo que  $i$  obrigatoriamente participa de sua vizinhança. Além disto, a vizinhança  $V_i$  contém os clientes mais próximos de  $i$ . Assim, um ciclo  $i = i_0 - i_1 - \dots - i_j - i_{j+1} = i$  não pode ser parte de um caminho-*ng* se  $i \in V_{i_k}$  para todo  $k = 1, 2, \dots, j$ . Equivalentemente, este ciclo é permitido se e somente se existe  $k \in \{1, 2, \dots, j\}$  tal que  $i \notin V_{i_k}$ .

Segundo Contardo, Cordeau e Gendron (2013), dado então um conjunto de vizinhanças  $V_i \subseteq \bar{N}, \forall i \in \bar{N}$ , o problema de encontrar no grafo  $G$  o caminho-*ng* mais curto entre 0 e  $n + 1$  que satisfaça a restrição de capacidade e janelas de tempo é chamado de *ng-SPPRC*. É notável que, se  $V_i = \{i\}$  para todo  $i \in \bar{N}$ , então o *ng-SPPRC* é um *SPPRC puro*. O caso oposto, onde  $V_i = \bar{N}$ , tem-se um Problema do Caminho Mínimo *Elementar* com Restrição de Recursos. Quando o *ng-SPPRC* não é uma generalização do *2-ciclos-SPPRC*, ou seja, existem  $i \in \bar{N}$ , tal que  $V_i \neq \bar{N}$ , então podem existir caminhos-*ng* que contenham *2-ciclos*.

Para a resolução do *PRVC*, pode-se usar o *ng-SPPRC* como subproblema onde o tamanho das vizinhanças  $V_i, \forall i \in \bar{N}$  é fixo. Conforme dito, se a cardinalidade da vizinhanças é menor do que a do conjunto de nós clientes ( $|V_i| < |\bar{N}|$ ), não existe garantia de *elementariedade* dos caminhos gerados. Entretanto, Baldacci, Mingozzi e Roberti (2011) observaram que para instâncias com até 100 clientes, conjuntos com cardinalidade entre 8 e 10 são suficientes para a obtenção de bons limites e tempos computacionais. Uma alteração necessária na dominância (Seção 3.3.2.3) do Algoritmo 4 precisa ser realizada. Lembrando que  $V_\rho^{k'} \leq V_\rho^{k*}, \forall k = 1, \dots, n$  o vetor binário que indica se um nó é inviável de uma futura extensão. Seja então um conjunto  $\Pi(\rho_{0i})$  que irá substituir o uso do vetor  $V_\rho^{k'}$  na dominância do algoritmo, mas não em sua função na estrutura de repetição (Ex.: linha 11). O rótulo  $\rho_{0i}$  passa a conter o elemento  $\Pi$  e este será inicializado e atualizado da seguinte maneira. Na *Inicialização* do rótulo  $\rho_0$  (linha 2) o conjunto  $\Pi(\rho_0)$  será igual a  $\{\emptyset\}$ . Nas demais extensões, o  $\Pi(\rho_{0i})$  só poderá ser estendido a  $j$ , se  $j \notin \Pi(\rho_{0i})$  e quando estendido criará um novo rótulo  $\rho'_{0j}$ , onde:

$$\Pi(\rho'_{0j}) = (\Pi(\rho_{0i}) \cap V_j) \cup \{j\}$$

A condição de dominância (iv) será substituída por:

$$v. \Pi(\rho'_{0j}) \subseteq \Pi(\rho_{0j}^*)$$

Devido a cardinalidade de  $\Pi$  ser menor ou igual a de  $V_i, \forall i \in \bar{N}$  a condição (v) menos restritiva do que (iv), Seção 3.3.2.3, levando geralmente a mais rótulos dominados e tempos computacionais mais rápidos. A ideia é comparar todos os clientes de uma única vez, se o novo elemento  $j$  não pertence a vizinhança  $V_i$ , ele não é considerado atrativo. Quando compara-se utilizando o vetor binário  $V_\rho^i$ , se pelo menos uma destas posições for melhor (número de visitas inferior), ele é considerado atrativo e assim gera mais rótulos *não-dominados*. Conforme mostrado acima, os clientes  $j$  não passíveis de extensão também são levados em conta no conjunto  $\Pi$ .

## 3.4 Planos de Cortes

O *Problema de Particionamento de Conjunto* é notoriamente conhecido por ter uma ampla gama de restrições para sua estrutura. Algumas foram listadas por Balas (1977) e que tinham diversos propósitos como, melhorar os limites, atingir mais rápido a solução ótima ou também facilitar a expressão da faceta convexa.

Há também duas famílias de cortes que se fazem notar, sendo os *Cortes de Gomory* (1958) e os de *Chvátal-Gomory* (CHVÁTAL, 1973). Ambos foram estudados para este trabalho e algumas conclusões podem ser tiradas.

A aplicação dos *Cortes de Gomory* se tornou inviável porque as variáveis do *PMR* são diferentes das usadas pelos subproblemas, não existindo forma linear de expressar a relação entre estas variáveis e o subproblema, assim seria interessante utilizá-lo, por exemplo, em um problema decomposto pela técnica de Dantzig e Wolfe (1960). Os *Cortes de Chvátal-Gomory* não mostraram uma aplicação eficaz. Isso porque a elaboração de cada um destes cortes leva em consideração linhas do quadro *SIMPLEX* final onde existam variáveis fracionárias, fazendo com que a folga dessas linhas seja menor e as variáveis *básicas* se aproximem de seus valores inteiros mais próximos. Porém para o problema de *Particionamento de Conjunto*, os cortes só podem ser criados usando as linhas das restrições (3.2), pois é através delas que serão retirados os preços duais  $\pi_i$  usados nos subproblemas. Caso um corte seja gerado utilizando as linhas de algum dos cortes previamente gerados, *Cortes de Chvátal-Gomory de posto*  $\geq 1$ , não existirá mais a relação direta e linear com os nós clientes. Um fato interessante desses cortes é que as variáveis acabam *escapando* para as linhas dos cortes. Assim, estes cortes são interessantes para *geração de colunas* quando os cortes são do tipo *posto* 1, ou seja, não são gerados sobre linhas de cortes anterior.

Desde que o *Problema Mestre* já foi definido como *Particionamento de Conjunto*, fica óbvio continuar nesta direção em busca de cortes válidos para esta estrutura. As duas famílias de cortes utilizadas são propostas recentes da literatura e são apresentadas nas Seções (3.4.1) e (3.4.2).

O algoritmo utilizado é do tipo *Column-and-cut* e supondo  $L$  cortes já tenham sido adicionados ao *PMR*, os custos reduzidos referentes a estes cortes devem ser contabilizados no *subproblema de precificação* escolhido. De forma geral ele tomará a forma apresentada em (3.8), conforme apresentado por Barnhart et al. (1998) e Desaulniers, Desrosiers e Spoorendonk (2011).

$$\bar{c}_{ij} = c_{ij} - \bar{\pi}_i - \sum_{l=1}^L a_l \bar{\phi}_l \quad (3.8)$$

Onde  $a_l$  é o coeficiente relacionado à variável dual  $\bar{\phi}_l$  que representa genericamente às variáveis duais das restrições adicionadas ao *PMR*. A adição dos cortes e uma explicação mais densa sobre a contabilização no *custo reduzido* oferecido por cada um deles para os *subproblemas* utilizados serão apresentadas nas Seções (3.4.1) e (3.4.2).

### 3.4.1 Restrições sobre Subconjunto de Linhas

Jepsen et al. (2008) se basearam na ideia de que um conjunto de restrições para o *Problema do Empacotamento* (BALAS, 1977) é um subconjunto de cortes válidos para o problema de *Particionamento de Conjunto* (3.1)-(3.4).

Assim, parte-se nessa direção em busca de cortes válidos para esta formulação. Os autores Jepsen et al. (2008) primeiramente utilizaram esta família de restrições em um algoritmo *Branch-cut-and-price* para o *PRVJT*. Os autores também provaram que esta família de cortes é derivada dos cortes de *Chvátal-Gomory* (CHVÁTAL, 1973), dessa forma é empregada como forma de diminuir a *folga* em determinadas linhas do *tableu SIMPLEX*, assim, "*empurrando*" as variáveis que vierem a ocupar estas linhas a terem valores mais próximos de um valor inteiro. Estas restrições são chamadas de *Restrições sobre Subconjunto de Linhas (SSR - Subset-Row Inequalities)* e estão diretamente ligadas às linhas e não às colunas do problema, o que permite identificar o coeficiente de cada uma das colunas para cada inequação gerada. É notável então que o coeficiente  $[\sum_{i \in \mathbb{S}_l} a_{ir}]$  é maior ou igual a 1 quando  $\kappa$  ou mais clientes em  $\mathbb{S}$  são visitados em  $r$ .

$$\sum_{r \in R} \left[ \sum_{i \in \mathbb{S}_l} a_{ir} \right] \lambda_r \leq \left\lfloor \frac{|\mathbb{S}_l|}{\kappa} \right\rfloor, \forall \mathbb{S}_l \subseteq \bar{N} \quad (\beta_l) \quad (3.9)$$

Existem duas formas para realizar a separação destas restrições. Em um primeiro caso, são listadas um determinado número de restrições que violam a atual solução, para valores fixos de  $|\mathbb{S}|$  e  $\kappa$ . No outro lado, tem-se a resolução de um *MIP* que é *NP-Completo* e dará um único corte que oferecerá a maior violação para a atual solução  $\lambda^*$ . A formulação (3.10)-(3.12) gera este corte, onde  $R^+$  representa o conjunto das variáveis *básicas* não-nulas.

$$\max \sum_{r \in R^+} \left[ \frac{1}{\kappa} \sum_{i \in \bar{N}} a_{ir} x_i \right] \lambda_r^* - \left\lfloor \frac{|\mathbb{S}|}{\kappa} \right\rfloor \quad (3.10)$$

s.t.:

$$\sum_{i \in \bar{N}} x_i = |\mathbb{S}| \quad (3.11)$$

$$x_i \in \{0,1\} \quad \forall i \in \bar{N} \quad (3.12)$$

A função objetivo (3.10) oferece a máxima violação obtida por este corte e para isso utiliza como parametro o valor ótimo  $\lambda^*$  da solução corrente. A restrição (3.11) garante que  $|\mathbb{S}|$  linhas sejam selecionadas e a (3.12) explicita o domínio da variável  $x_i$  do problema. Um ponto interessante aqui é que computacionalmente, de acordo com o *solver* utilizado, venha a existir a necessidade da *linearização* do termo  $\left\lfloor \frac{1}{\kappa} \sum_{i \in \bar{N}} a_{ir} x_i \right\rfloor$ . Para isso, faz-se  $z_r = \left\lfloor \frac{1}{\kappa} \sum_{i \in \bar{N}} a_{ir} x_i \right\rfloor$ ,  $z_r \in \mathbb{Z}^+$ ,  $\forall r \in R^+$  e cria-se duas restrições,  $z_r \leq \frac{1}{\kappa} \sum_{i \in \bar{N}} a_{ir} x_i$  e  $z_r \geq \frac{1}{\kappa} \sum_{i \in \bar{N}} a_{ir} x_i - 1 + \delta$ . O valor de  $\delta$  será grande o suficiente para que impeça  $z_i$  de assumir um valor inteiro menor do que o que seria encontrado pelo arredondamento. A título de exemplo, se o arredondamento fosse exatamente igual a 6, o valor dada a linearização poderia assumir  $5 = 6 - 1$ . Com a adição de um  $\delta$  qualquer, por exemplo  $\delta = 0,01$ , teria-se  $5,01 = 6 - 1 + 0,01$ , visto que  $z_i$  é do tipo inteira, a variável assumiria o valor 6.

Após a adição de algumas *SSR* ao modelo, o *custo reduzido* destas restrições também deve ser considerado no subproblema. Seja  $\beta_l \leq 0, \forall \mathbb{S}_l \subseteq \bar{N}$ , a variável dual associada a cada uma destas restrições. Quando um rótulo  $\rho_{0i}$  é estendido nos métodos *ng-SPPRC* (Seção 3.3.2) e *ESPPRC* (Seção 3.3.2.1) ou uma nova posição da matriz do método *q-routes* (Seção 3.3.1), o custo reduzido será computado da seguinte forma:

$$\bar{c} = \begin{cases} \bar{c}(i) + c_{ij} - \bar{\pi}_i - \sum_{i \in \mathbb{S}} \bar{\beta}_l \left\lfloor \frac{|\mathbb{S}_l \cap U|}{\kappa} \right\rfloor, & \text{se } \mathbb{S}_l \cap U \neq \{\emptyset\} \text{ e se } j = n + 1 \\ \bar{c}(i) + c_{ij} - \bar{\pi}_i, & \text{caso contrário} \end{cases} \quad (3.13)$$

Onde  $\bar{c}(i)$  é o *custo reduzido* acumulado até o último nó  $i$  visitado e  $U$  é o conjunto que contém os nós participantes do caminho  $0, \dots, i$ , respectivamente representados como  $\bar{c}(\rho)_i$  e  $U_\rho^i$  para o algoritmo *ng-SPPRC* (Seção 3.3.2),  $\bar{c}(R(d,i))$  e  $U(R(d,i))$  para o método de *q-routes* (Seção 3.3.1). Conforme Jepsen et al. (2008) definem que o custo relacionado aos cortes deve ser debitado somente no último arco  $(i,j) = (i, n + 1)$  formado pelo caminho. Para os algoritmos do *ng-SPPRC* e *ESPPRC*, isto é fácil de se implementar, já que eles são *progressivos*, ou seja, a criação dos caminhos segue a ordem  $0, i_1, i_2, \dots, i, n + 1$ . Quando estamos no método das *q-routes*, os caminhos são *regressivos*, ou seja, criados de  $n + 1$  até um nó  $i$ . Como cada posição da matriz representa o custo reduzido acumulado deixando  $i$  com capacidade  $d$  e chegando ao depósito  $n + 1$ , é necessário que a cada nova posição  $R(d,i)$  preenchida, primeiro seja creditado o custo reduzido dos cortes *SSR* da posição destino  $R(d - d_i, j)$ . Depois, faz-se a união  $U(d - d_i, j) \cup \{i\}$

e subtrai-se novamente os custos relacionados ao corte relevando o novo cliente  $i$  da rota.

Após a adição dos cortes  $SSR$ , o item  $(i.)$  ( $\bar{c}(\rho)'_i \leq \bar{c}(\rho)^*_i$ ) da regra de dominância para o algoritmo  $ng\text{-SPPRC}$  deve ser atualizado. Primeiro, é necessário a definição de um novo recurso, denominado por  $\mathcal{T} = |\mathbb{S} \cap V(\rho)| \bmod \kappa$ , que representa o número de visitas feitas ao subconjunto  $|\mathbb{S}|$  desde que a última penalidade foi paga por visitar  $\kappa$  nós pertencentes a  $\mathbb{S}$ . Seja  $Z = \{z : \beta_z < 0 \wedge \mathcal{T}_z(\rho'_{0i}) > \mathcal{T}_z(\rho^*_{0i})\}$  um subconjunto de cortes  $SSR$ . O propósito dessa alteração é fazer com que o subconjunto  $\mathbb{S}$  seja visitado um número suficiente de vezes para que a penalidade relacionada a cada um dos cortes  $SSR$  seja cobrada. Assim, a nova condição de dominância tomará a forma:

$$\bar{c}(\rho)'_i + \sum_{z \in Z} \beta_z \leq \bar{c}(\rho)^*_i \quad (3.14)$$

Um fato interessante é que caso o subproblema utilizado seja similar a um *Problema do Caixeiro Viajante com Coleta de Prêmios*, o custo é contabilizado na função objetivo na forma  $\lfloor \frac{1}{k} \sum_{i \in \mathbb{S}_i} y_i \rfloor$  (DESAULNIERS; DESROSIERS; SPOORENDONK, 2011). A grande dificuldade aqui é que de acordo com o *solver* utilizado, o termo que será arredondado para baixo precisa ser linearizado, como explicado para formulação (3.10)-(3.12).

### 3.4.2 Restrições Fortes de Conexão

Contardo, Cordeau e Gendron (2013), trabalhando com o  $PRVJT$ , notaram que a imposição da *elementariedade* a um conjunto reduzido de clientes  $i$  era mais vantajoso do que impor a todo o conjunto  $\bar{N}$  de clientes. Os autores ainda afirmaram que caso colunas cíclicas sejam permitidas no modelo (3.1)-(3.4) será notável que ocorrerão mais ciclos em alguns clientes do que em outros. Esse comportamento é percebido principalmente em nós que tenham a demanda muito pequena, janelas de tempo mais amplas ou que estejam muito próximos de seus vizinhos, fazendo com que sejam visitados com uma frequência maior do que os demais clientes. Os ciclos ocorrerão menos naqueles clientes com demanda muito alta, janelas de tempo mais estreitas ou distantes dos respectivos vizinhos. Em busca de explicitar essa necessidade de que os clientes *mais tendenciosos* a participarem de ciclos, os autores introduziram as então denominadas *Restrições Fortes de Conexão* ( $SDC$  - *Strong Degree Constraints*). Elas são definidas por:

$$\sum_{r \in R} b_{ir} \lambda_r \geq 1, \forall i \in \mathcal{E} \ (\sigma_i) \quad (3.15)$$

Onde  $\mathcal{E} \subseteq \bar{N}$  é o conjunto de clientes que tem a *elementariedade* requerida,  $b_{ir} = \min\{1, a_{ir}\}, \forall i \in \mathcal{E}, \forall r \in R$  é um parametro binário que tem valor igual a 1 se o cliente  $i$  é visitado pelo menos uma vez. As restrições (3.15) irão impor a cada rota que contenha ciclo em um cliente  $i \in \mathcal{E}$  do problema (3.1)-(3.4), que esta variável não tome valor positivo na relaxação.

Após a adição de algumas  $SDC$  ao modelo, o *custo reduzido* destas restrições também deve ser considerado no subproblema. Seja  $\sigma_i \geq 0, \forall i \in \mathcal{E}$ , a variável dual associada a cada uma destas restrições. Quando um rótulo  $\rho_{0i}$  é estendido no método  $ng\text{-SPPRC}$  (Seção 3.3.2) ou uma nova posição da matriz do método  $q\text{-routes}$  (Seção 3.3.1), o custo reduzido será computado da seguinte forma:

$$\bar{c} = \begin{cases} \bar{c}(i) + c_{ij} - \bar{\pi}_i - \bar{\sigma}_j, & \text{se } j \in \mathcal{E} \setminus U \\ \bar{c}(i) + c_{ij} - \bar{\pi}_i, & \text{caso contrário} \end{cases} \quad (3.16)$$

Onde  $\bar{c}(i)$  é o *custo reduzido* acumulado até o último nó  $i$  visitado e  $U$  é o conjunto que contém os nós participantes do caminho  $0, \dots, i$ , respectivamente representados como  $\bar{c}(\rho)_i$  e  $U_\rho^i$  para o algoritmo  $ng\text{-SPPRC}$  (Seção 3.3.2) e  $\bar{c}r(R(d,i))$  e  $U(R(d,i))$  para o método de  $q\text{-routes}$  (Seção 3.3.1). É notável que o conjunto  $U$  é utilizado para determinar se o preço dual  $\sigma_j$  será debitado ou não, ou seja, somente quando o cliente  $j$  é visitado pela primeira vez ao longo do caminho.

Com a adição destes cortes, a regra de dominância para o algoritmo *ng-SPPRC* terá de ser atualizada. A condição (i.)  $(\bar{c}(\rho)'_i \leq \bar{c}(\rho)_i^*)$  toma a forma:

$$\bar{c}(\rho)'_i + \sum_{i \in \mathcal{E} \cap U(\rho'_{0i}) \setminus U(\rho_{0i}^*)} \sigma_i \leq \bar{c}(\rho)_i^* \quad (3.17)$$

Esta nova condição faz com que, para os nós cuja *elementariedade* não é muito restritiva, ou seja, associados a pequenos valores *duais* das *SDC*, estes tem menor representatividade na regra de dominância do que aqueles que tem valores *duais* maiores.

### 3.5 Estratégia de aceleração

O notável que grande parte do tempo gasto nos métodos de geração de colunas se dá na complexidade dos subproblemas. Como o trabalho está em busca de soluções viáveis para o *PRVC* é interessante que a cada vez que os algoritmos *ng-SPPRC* (Seção 3.3.2) e *q-routes* (Seção 3.3.1) são chamados eles gastem mais de seu tempo de processamento gerando estas colunas viáveis. Como eles não tem a *elementariedade* garantida, ambos podem consumir muito tempo computacional e identificar um demasiado número de caminhos *não-elementares*. Dado que o objetivo dos subproblemas é de encontrar rotas *elementares* de custo reduzido negativo, tais caminhos podem ser encontrados mesmo que os subproblemas não a garantam para todas as rotas geradas.

#### 3.5.1 Relaxação Incremental de Espaço

Um problema enfrentado pelos algoritmos *ng-SPPRC* e *q-routes* é que a *elementariedade* não é garantida. Dessa forma, para induzir a formação de rotas *elementares*, além dos preços duais, pode-se utilizar de outras ferramentas. Podem ser citadas a eliminação de *2-ciclos* proposta por Houck et al. (1978), onde ciclos do tipo  $i \rightarrow j \rightarrow i, \forall i, j \in \bar{N}$  são evitados. Existe também a eliminação de *k-ciclos*, proposta por Irnich e Villeneuve (2006), que se baseia na não visitação entre clientes de um conjunto seletor, segundo algumas condições impostas pelo problema. Mais detalhes podem ser vistos em Desaulniers, Desrosiers e Solomon (2005) e em Larsen (1999).

Uma estratégia que foi adaptada para auxiliar na formação das rotas elementares é a *Relaxação Incremental de Espaço* (*DSSR - Incremental State-space Relaxation*), proposta por Righini e Salani (2008). Ela é realizada considerando um novo subconjunto de clientes, mais precisamente, seja  $E \subseteq \bar{N}$  um subconjunto de clientes cuja a *elementariedade* é requerida. Quando um rótulo  $L$  é estendido, o conjunto de clientes que são inalcançáveis, para *ng-SPPRC* ou que participam da rota, para *q-routes*, deve ser atualizado, conforme mostrado em 3.18.

$$U(L') = \begin{cases} U(L) \cup \{j\}, & \text{se } j \in E \setminus U(L) \\ U(L), & \text{caso contrário.} \end{cases} \quad (3.18)$$

Onde  $U(L)$  é o conjunto que contém os nós inviáveis do rótulo  $0, \dots, i$  para o algoritmo *ng-SPPRC*, representado por  $\Pi(\rho_{0i}) \cup \{j\}$  (Seção 3.3.2). Para o método das *q-routes* (Seção 3.3.1),  $U(L)$  representa os participantes  $U(R(d, i))$ .

O interessante desse aspecto é que a regra de dominância para o algoritmo *ng-SPPRC* permanece inalterada, não modificando a complexidade do método. Consequentemente, ciclos podem ocorrer em qualquer cliente  $i \notin E$ . Para buscar a *elementariedade*,  $E$  é aumentado conforme necessário.

Tipicamente, o procedimento *DSSR* inicia com  $E = \{\emptyset\}$ . Se o caminho com menor custo reduzido contém ciclo em um cliente  $j$ , este cliente  $j$  é adicionado ao conjunto  $E$ , conforme apresentado por Contardo, Desaulniers e Lessard (2015). Desaulniers, Lessard e Hadjar (2008) notaram que para o procedimento de geração de colunas é interessante manter o conjunto  $E$  para todas as iterações.

### 3.6 Esquemas para obtenção do limite inferior *elementar*

Nesta seção são descritas as diferentes estratégias testadas neste artigo para alcançar melhores limites inferiores *elementares* ou não para o modelo (3.1)-(3.4) relaxado. Estas estratégias são baseadas nos subproblemas, cortes e estratégia de aceleração previamente apresentados.

#### 3.6.1 Esquema *puramente elementar*

Neste primeiro *esquema* foram geradas somente colunas *elementares* para serem adicionadas ao *PMR*, dessa forma, o subproblema usado foi o *ESPPRC*.

O Algoritmo 5 representa o funcionamento do método. O valor  $z_{SP}^E$  representa o valor da variável *elementar* gerada pelo subproblema com custo reduzido mais negativo.

Na primeira variação, nenhum corte da família *SSR* foi adicionado ao *PMR*, assim, o esquema fica reduzido às linhas 1,2 e 4 a 9. O propósito do uso destes cortes, visto que são derivados dos cortes de *Chvátal-Gomory* (CHVÁTAL, 1973), é *forçar* as variáveis relaxadas a se aproximarem de seus limites, ou seja, serem menos fracionárias.

---

**Algoritmo 5:** *Esquema Column-and-Cut elementar*

---

- 1 INICIALIZAÇÃO
  - 2 Construa para o *PMR* uma solução inicial viável
  - 3 **repita**
  - 4     **repita**
  - 5         Resolva o *PMR* com as *SSR*
  - 6         Resolva o *ESPPRC*
  - 7         **se**  $z_{SP} < 0$  **então**
  - 8             Adicione as colunas de custo reduzido negativo geradas ao *PMR*
  - 9         **até que**  $z_{SP}^E \geq 0$
  - 10        **se** a solução  $\lambda^*$  é violada por alguma *SSR* **faça**
  - 11            Adicione as restrições *SSR* violadas ao *PMR*
  - 12        **fim para todo**
  - 13 **até que** não existam mais *SSR* violadas
-

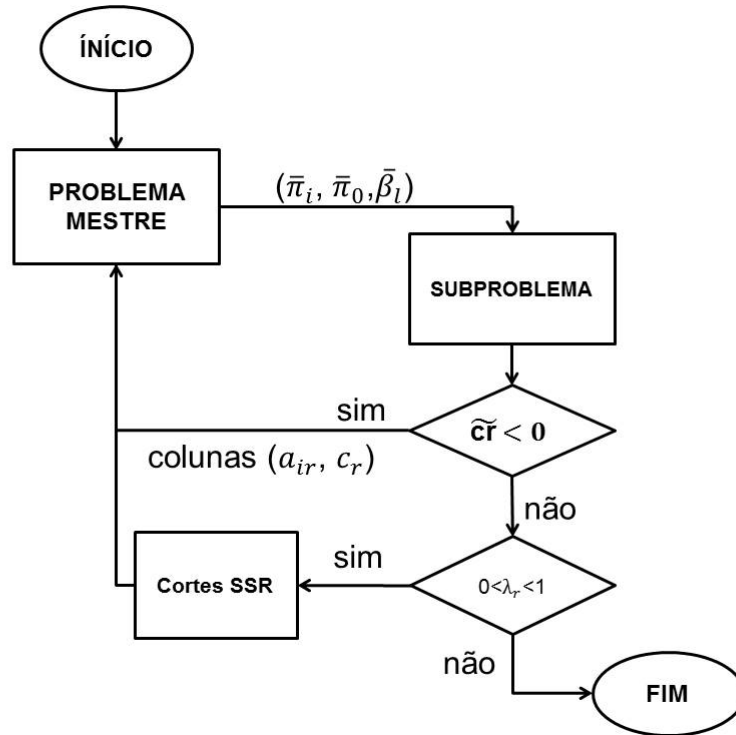


Figura 3.6: Esquema dos Algoritmos 5 e 6. Elaboração própria.

### 3.6.2 Esquemas *gradativamente elementar*

Os segundo e terceiro *esquemas* permitem *popular* o *PMR* também com colunas *inviáveis*, que contém ciclos. Para isso os *subproblemas* utilizados foram o *ng-SPPRC* e *q-routes*.

Os Algoritmos 6 e 7 representam o funcionamento dos métodos que permitem ciclos combinados com as restrições *SSR* e *SDC*, respectivamente. As vizinhanças usadas não tiveram seu tamanho alterado durante a execução, sendo fixadas em  $|V_i| = 10, \forall i \in \bar{N}$ , conforme orientado pelos autores Baldacci, Mingozzi e Roberti (2011). O valor  $z_{SP}$  representa o valor da variável gerada pelo *subproblema* com custo reduzido mais negativo, sendo esta variável *elementar* ou não.

Primeiramente, buscando visualizar a melhoria alcançada pelo uso das novas restrições, nenhum corte das famílias *SSR* ou *SDC* foi adicionado ao *PMR*. Assim o esquema permite ciclos na solução ótima  $\lambda^*$  e fica reduzido às linhas 1 a 4 e 6 a 12 no Algoritmo 6, onde somente a estratégia *DSSR* é usada e na linha 7 o *PMR* é resolvido sem as *SDC* ou *SSR*.

O segundo *esquema* é executado da mesma forma que o apresentado pela Figura 3.6. A diferença aqui, é que a solução ótima  $\lambda^*$  não tem garantida a *elementariedade*, pois não há utilização das *SDC* para eliminar os ciclos da solução final. A condição de parada do *laço* mais externo é a não existência de restrições *SSR* violadas pela solução ótima corrente.

O Algoritmo 7 apresenta o último *esquema* usado, representado na Figura 3.7. Neste caso são adicionadas somente as restrições *SDC*. Estas restrições, após a finalização do procedimento de *Geração de Colunas*, visa proibir a participação de colunas *não-elementares* da solução ótima  $\lambda^*$ . A cada iteração do *laço* mais interno (linhas 6 a 12), todas as colunas geradas pelos *subproblemas* são adicionadas ao *PMR*, até que o menor *custo reduzido* dentre as variáveis geradas tenha seu valor igual ou maior do que 0. Durante a segunda parte do algoritmo é iniciada uma varredura na solução atual do problema que procura por variáveis *básicas* que contenham ciclos. Para cada cliente  $i \in \bar{N}$  que tenha um ciclo na solução em alguma das variáveis *básicas*, ele terá uma *SDC* gerada e adicionada ao *PMR*.



---

**Algoritmo 6:** *Esquema Column-and-Cut não-elementar*

---

- 1 INICIALIZAÇÃO
  - 2 Considere as vizinhanças  $V_i, \forall i \in \bar{N}$ , se o subproblema as utiliza
  - 3 Considere um conjunto inicial  $E = \{\emptyset\}$ , dos clientes que terão a *elementariedade* requerida através da *DSSR*
  - 4 Construa para o *PMR* uma solução inicial viável
  - 5 **repita**
  - 6     **repita**
  - 7         Resolva o *PMR* com as *SSR*
  - 8         Resolva o *subproblema* com as vizinhanças  $V_i, \forall i \in \bar{N}$ , caso este as utilize.
  - 9         **se**  $z_{SP} < 0$  **então**
  - 10             Adicione as colunas de custo reduzido negativo geradas ao *PMR*
  - 11             Atualize  $E \leftarrow E \cup \{i\}$ , caso na coluna com menor custo reduzido negativo exista ciclo para um dado cliente  $i$
  - 12         **até que**  $z_{SP} \geq 0$
  - 13         **se** a solução  $\lambda^*$  é violada por alguma *SSR* **faça**
  - 14             Adicione as restrições *SSR* violadas ao *PMR*
  - 15 **até que** não existam mais *SSR* violadas
- 

A contribuição dos cortes *SDC* são levadas ao *subproblema* escolhido, conforme mostrado na Seção 3.4.2. Estes cortes ajudarão na formação de rotas mais *elementares*, já que tornará mais atrativo a visitação uma única vez de um nó que tenha uma *SDC*.

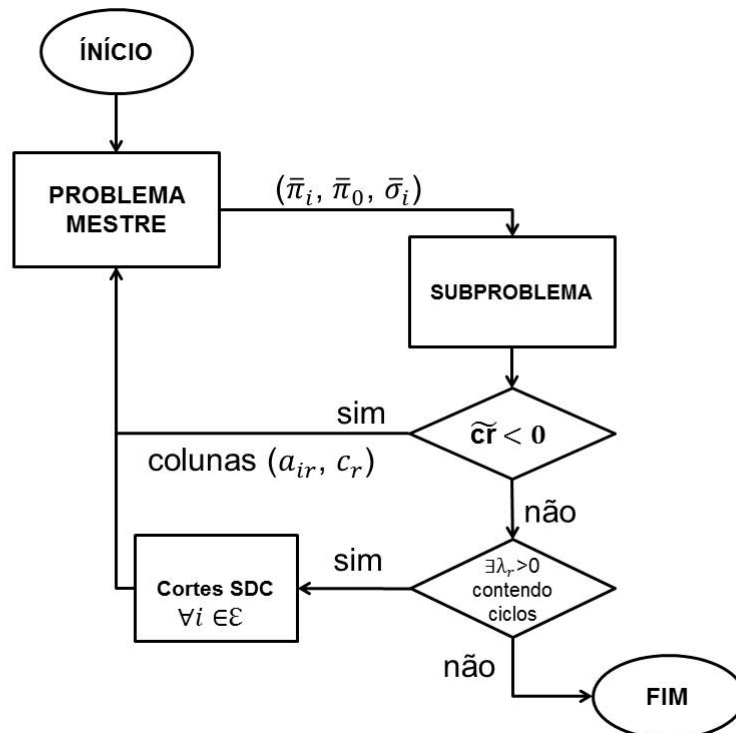


Figura 3.7: Esquema do Algoritmo 7. Elaboração própria.

---

**Algoritmo 7:** *Esquema Column-and-Cut gradativamente elementar*

---

- 1 INICIALIZAÇÃO
  - 2 Considere as vizinhanças  $V_i, \forall i \in \bar{N}$ , se o subproblema as utiliza
  - 3 Considere um conjunto inicial  $E = \{\emptyset\}$ , dos clientes que terão a *elementariedade* requerida através da *DSSR*
  - 4 Considere um conjunto inicial  $\mathcal{E} = \{\emptyset\}$ , dos clientes que terão a *elementariedade* requerida através das *SDC*
  - 5 Construa para o *PMR* uma solução inicial viável
  - 6 **repita**
  - 7     **repita**
  - 8         Resolva o *PMR* com as *SDC* para o dado conjunto  $\mathcal{E}$
  - 9         Resolva o *subproblema* com as vizinhanças  $V_i, \forall i \in \bar{N}$ , caso este as utilize.
  - 10        **se**  $z_{SP} < 0$  **então**
  - 11            Adicione as colunas de custo reduzido negativo geradas ao *PMR*
  - 12            Atualize  $E \leftarrow E \cup \{i\}$ , caso na coluna com menor custo reduzido negativo exista ciclo para um dado cliente  $i$
  - 13        **até que**  $z_{SP} \geq 0$
  - 14        **para toda** coluna  $\lambda_r^* > 0$  do *PMR* que contenha ciclos em um cliente  $i$  **faça**
  - 15            Atualize o conjunto dos clientes que requerem *elementariedade*,  $\mathcal{E} \leftarrow \mathcal{E} \cup \{i\}$
  - 16            Adicione ao *PMR* ou atualize as *SDC* referentes aos clientes  $i \in \mathcal{E}$
  - 17        **fim para todo**
  - 18 **até que** todos as colunas pertencentes à solução do *PMR* sejam elementares
-

A Figura 3.8 ilustra o processo iterativo de adição de *colunas* (variáveis) e *cortes* (restrições) ao problema. Para este trabalho, sempre que uma nova coluna era adicionada ao *PMR*, caso já existisse algum *corte* no problema, o *corte* também seria atualizado com as informações das novas colunas.

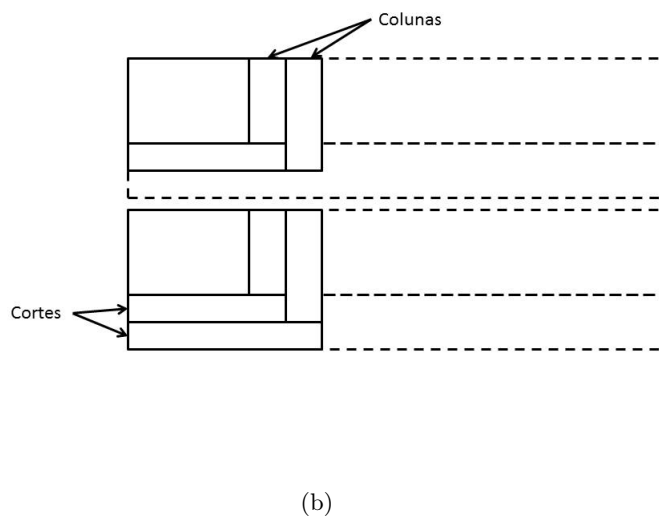
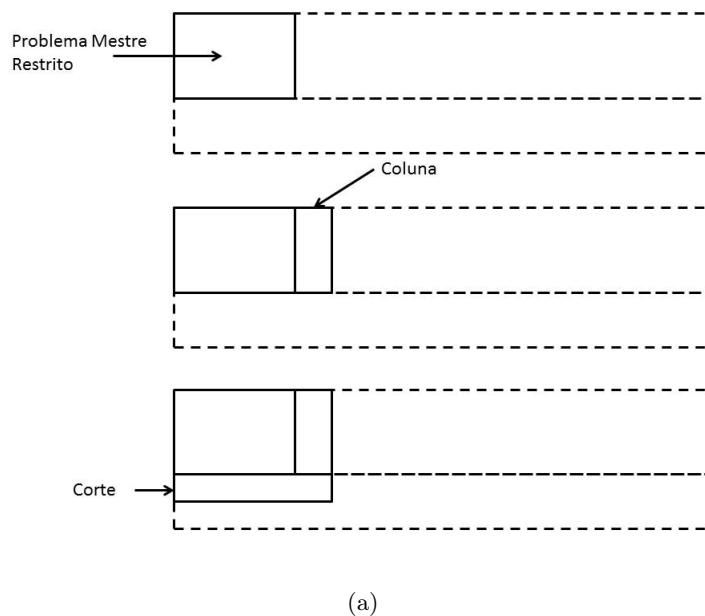


Figura 3.8: Adição de colunas e cortes ao *PMR*. Adaptado de Spoorendonk (2008)

## Capítulo 4

# Experimentos e resultados computacionais

Os modelos foram desenvolvidos usando a linguagem C/C++ junto com a tecnologia IBM CPLEX<sup>®</sup> 12.5 *Callable* em um notebook com sistema operacional Ubuntu Linux 14.04 LTS, processador Intel<sup>™</sup> Core<sup>®</sup> i5-4210U e 4GB de *RAM*. Os testes foram executados em uma máquina Dell<sup>™</sup> PowerEdge<sup>®</sup> T620 *workstation*, equipada com dois processadores Intel<sup>™</sup> Xeon<sup>®</sup> E5-2600v2 e 96GB de *RAM*.

### 4.1 Instâncias utilizadas e primeiros experimentos

As instâncias utilizadas foram algumas das disponíveis na literatura. As instâncias dos tipos **A**, **B** e **P** são de autoria de Augerat (1995). As **E** de Christofides e Eilon (1969) e **F** de Fisher (1994). No nome de cada instância é possível identificar o autor pela primeira letra, o número de nós clientes que é precedido pela letra  $n$  e o número mínimo de veículos para atender a demanda, precedido pela letra  $k$ . Elas foram escolhidas por terem seu valor ótimo ( $LS$ ) conhecido para o *PRVC*. As instâncias e suas respectivas capacidades são apresentadas na Tabela 4.1. Elas tem diferentes características, o número de nós clientes, total de veículos, a localização do depósito (central, *corner* ou aleatória) e se os clientes são *clusterizados* (agrupados) ou randômicos. Quanto mais *clusterizada* a instância, maior a tendência a formação de ciclos, visto que os clientes estão mais próximos entre si.

Tabela 4.1: Instâncias e respectivas capacidades

<b>Instância</b>	A	B	E-n101-k14	E-n101-k8	E-n22-k4	E-n23-k3	E-n30-k3
<b>Capacidade</b>	100	100	112	200	6000	4500	4500
<b>Instância</b>	E-n33-k4	E-n51-k5	E-n76-k10	E-n76-k7	E-n76-k8	F-n45-k4	P-n51-k10
<b>Capacidade</b>	8000	160	140	220	180	2010	80

Nos primeiros testes realizados para instâncias de maior porte, foi verificada a dificuldade de convergência para o *LI* pelos algoritmos *ng-SPPRC* e *ESPPRC*. Diferente do método das *q-routes* que tem seu tempo computacional definido, ambos os algoritmos, do *Problema do Caminho Mínimo com Restrição de Recursos*, irão convergir se, e somente se, são gerados todos os rótulos *não-dominados* para o nó destino  $n + 1$ . Isso ocorre mais rápido à medida em que se aumenta a quantidade de recursos usados no critério de dominância, como é comum na literatura, janelas de tempo e coleta e entrega. Vale novamente ressaltar que este trabalho utilizou somente a *capacidade* como recurso que ditava a viabilidade de um novo rótulo. Assim, este algoritmo poderia ser chamado de *Problema do Caminho Mínimo (Elementar) com Restrição*

de *Capacidade*. Como meio de compensar essa dificuldade de convergência, definiu-se que o algoritmo é executado por até 60 segundos. Se este limite é atingido são entregues ao *PMR* os rótulos (*colunas*) gerados para o nó  $n + 1$ .

Em busca de melhorar a política de dominância, uma alternativa proposta por Jepsen et al. (2008) foi cogitada. Quando as restrições *SSR* são utilizadas pode-se criar um novo recurso  $\mathcal{T} = |\mathbb{S} \cap U| \bmod \kappa$ , que representa o número de visitas realizadas a  $\mathbb{S}$  desde que a última penalidade foi paga por visitar  $\kappa$  nós em  $\mathbb{S}$ . Isso não foi feito por dois motivos, o primeiro iria alterar a complexidade do subproblema, deixando de ser *robusto* e com isso, poderia provocar um desequilíbrio quando se fosse comparado aos outros métodos usando as *SDC* ou ao *q-routes* e suas combinações, pois este subproblema não trabalha com dominância. Um subproblema *robusto* é um subproblema que não tem sua complexidade alterada, por exemplo a adição de um recurso na dominância, quando o *PMR* sofre alguma alteração.

A combinação dos subproblemas que permitem ciclos com as restrições *SSR* e *SDC* simultaneamente não foi possível. A questão é que sem a técnica de *branch-and-bound* (LAND; DOIG, 1960) para fixar as variáveis, não foi possível interromper a geração de colunas com a solução ótima elementar e simultaneamente sem nenhuma *SSR* violada. Isso porque os procedimentos acabavam ficando presos no *laço* mais interno gerando as mesmas colunas e sem atingir o custo reduzido positivo ou igual a 0.

A quantidade de colunas adicionadas ao *PMR* se dava da seguinte maneira, para os algoritmos *ESPPRC* e *ng-SPPRC* todas as colunas geradas para o nó  $n + 1$  eram adicionadas ao *PMR*. Já para o método das *q-routes*, para cada cliente  $i \in \bar{N}$  escolhe-se a demanda total  $d^*$  que minimiza o custo reduzido da *q-route* correspondente, dado por  $c_{0i} + R(d^*, i)$ , para ser inserido no *PMR*.

As restrições *SDC* eram adicionadas conforme explicado na Seção 3.4.2. As restrições *SSR* foram enumeradas usando  $|\mathbb{S}| = 3$  e  $\kappa = 2$ , conforme orientado pelos autores Jepsen et al. (2008). Além disso, no máximo  $\lfloor \frac{|\bar{N}|}{4} \rfloor$  *SSR* violadas pela solução ótima eram adicionadas ao mestre a cada iteração. Esse número foi definido empiricamente, pois se mostrou mais adequado durante os testes realizados.

Conforme feito por Baldacci, Hadjiconstantinou e Mingozzi (2004), os custos de conexão entre os dois clientes  $i$  e  $j$  é um valor inteiro dado por  $c_{ij} = \lfloor e_{ij} + 0,5 \rfloor$ , onde  $e_{ij}$  é a *distância euclidiana* entre  $i$  e  $j$ .

Uma ferramenta computacional que se mostrou muito útil e que acelerou boa parte dos testes foram as operações *bit-a-bit*. Ela foi usada para representar a composição dos conjuntos  $U$  e  $\Pi$  dos métodos. Para cada cliente que  $i \in \bar{N}$  que participasse ou fosse inviável em um rótulo nos algoritmos *SPPRC* (elementar ou não) e participasse uma *q-route* sinalizada numa posição da matriz  $R$ , o *bit* de posição  $(i - 1)$  será sinalizado como 1 e caso contrário 0. O redução do tempo computacional foi notável na troca das estruturas de *list<int>* por *bitset<|N|>*.

## 4.2 Resultados

Nesta seção os resultados obtidos com os *esquemas* apresentados na Seção 3.6. Abaixo são listados as informações geradas a cada teste executado.

- **Instância:** Nome da instância utilizada;
- **LI:** Limite inferior, obtido pela Geração de Colunas;
- **LS:** Limite superior, encontrado na literatura;
- **GAP(%):** Diferença percentual entre o *LS* e o *LI*, Eq. 4.1;
- **#cols:** número de colunas retornado ao fim do processo;
- **#SSR:** número de restrições da família *SSR* adicionadas ao *PMR*;
- **#SDC:** número de restrições da família *SDC* adicionadas ao *PMR*;
- **#int:** número de iterações do *laço* mais interno;
- **#ext:** número de iterações do *laço* mais externo;
- **t (s):** tempo, em segundos, gasto pelo modelo para encontrar o *LI*.

A diferença percentual (*GAP*) entre a solução ótima (*limite superior* - *LS*), e a provida pela geração de colunas, que conforme a teoria apresentada na Seção 2.5.1, provê um *limite inferior* (*LI*) para o problema, é apresentada na Equação 4.1.

$$GAP(\%) = \frac{LS - LI}{LS} \% \quad (4.1)$$

Para o *GAP* e o *tempo de processamento* dos testes, foi calculado tanto a média, quanto a mediana. Isso porque dada a diferença das instâncias e seus respectivos resultados, a média por vezes poderia se tornar imprópria e não mostrar a característica real do resultado obtido. Isso é mostrado pelo *coeficiente de variabilidade*, se seu resultado for abaixo de 25% a média é apropriada, caso contrário, usa-se a mediana para averiguar o comportamento dos dados.

Os gráficos das Seções 4.2.1, 4.2.2 e 4.2.3 apresentam: (a) o *GAP* obtido e (b) o tempo de processamento, além da média e mediana encontrada.

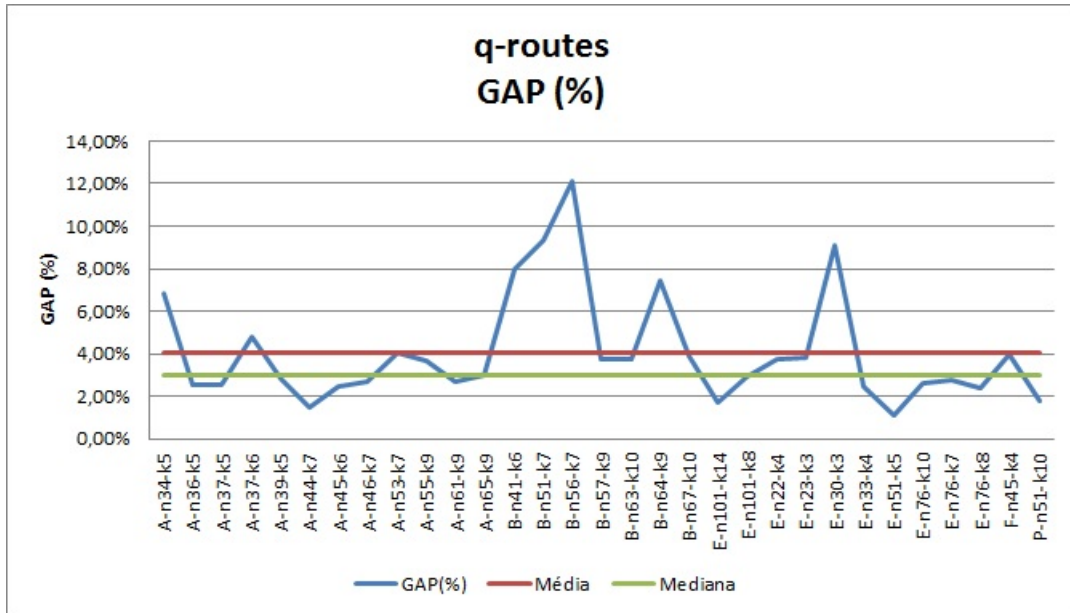
#### 4.2.1 Resultados utilizando o subproblema *q-routes*

A Tabela 4.2 apresenta os resultados utilizando o subproblema *q-routes* sem nenhum corte. Seu esquema de funcionamento pode ser representado pelas linhas 1 a 4 e 6 a 12 no Algoritmo 6, onde somente a estratégia *DSSR* é usada e na linha 7 o *PMR* é resolvido sem as *SDC* ou *SSR*.

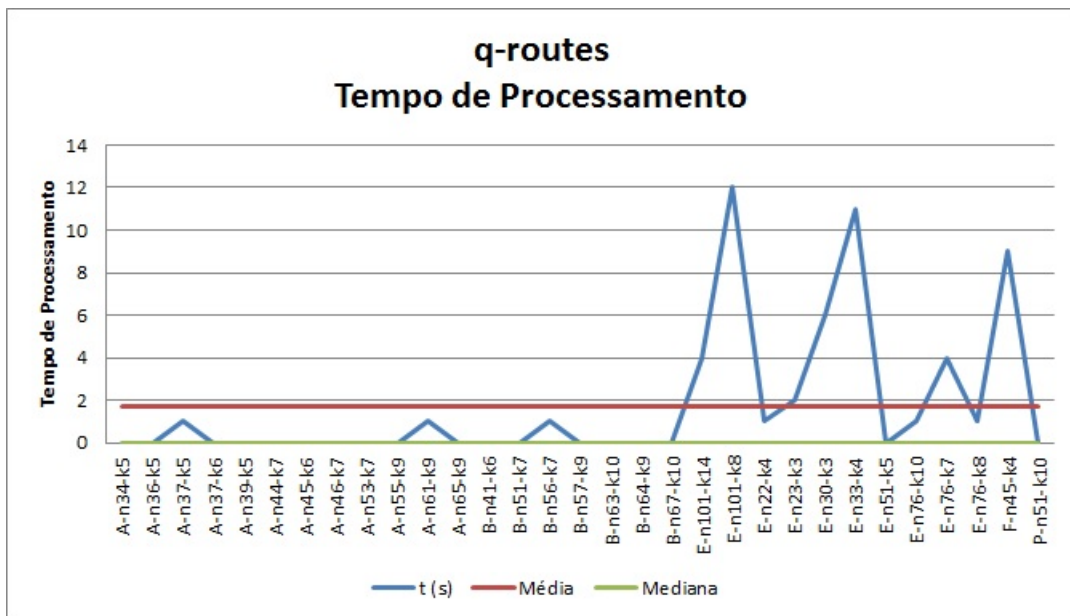
Tabela 4.2: Resultados para o modelo *q-routes*

Instância	LI	LS	GAP(%)	#cols	#iter	t (s)
A-n34-k5	724,565	778	6,87%	1.066	31	0
A-n36-k5	778,765	799	2,53%	1.690	47	0
A-n37-k5	652,036	669	2,54%	1.593	43	1
A-n37-k6	903,229	949	4,82%	1.200	32	0
A-n39-k5	798,527	822	2,86%	1.568	40	0
A-n44-k7	922,748	937	1,52%	1.775	40	0
A-n45-k6	921,064	944	2,43%	1.640	36	0
A-n46-k7	889,495	914	2,68%	1.543	33	0
A-n53-k7	968,885	1010	4,07%	2.978	56	0
A-n55-k9	1.033,990	1073	3,64%	1.800	32	0
A-n61-k9	1.006,290	1034	2,68%	2.718	44	1
A-n65-k9	1.139,290	1174	2,96%	2.770	42	0
B-n41-k6	762,668	829	8,00%	1.292	31	0
B-n51-k7	935,526	1032	9,35%	1.512	29	0
B-n56-k7	621,370	707	12,11%	2.598	46	1
B-n57-k9	1.537,660	1598	3,78%	1.921	33	0
B-n63-k10	1.439,470	1496	3,78%	2.376	37	0
B-n64-k9	796,637	861	7,48%	2.475	38	0
B-n67-k10	991,749	1032	3,90%	2.527	37	0
E-n101-k14	1.048,340	1067	1,75%	7.828	77	4
E-n101-k8	791,359	815	2,90%	12.716	126	12
E-n22-k4	360,910	375	3,76%	302	13	1
E-n23-k3	547,088	569	3,85%	687	30	2
E-n30-k3	485,375	534	9,11%	1.776	60	6
E-n33-k4	814,646	835	2,44%	1.448	44	11
E-n51-k5	515,297	521	1,09%	3.210	63	0
E-n76-k10	808,476	830	2,59%	3.620	47	1
E-n76-k7	663,197	682	2,76%	6.539	86	4
E-n76-k8	717,188	735	2,42%	4.516	59	1
F-n45-k4	695,098	724	3,99%	3.572	80	9
P-n51-k10	727,868	741	1,77%	1.320	25	0

Os Gráficos 4.1 *a* e *b* apresentam o *GAP* e *tempo de processamento*, respectivamente para cada uma das instâncias testadas. A média e a mediana do *GAP* foram 4,08% e 2,96%. Para o *tempo de processamento* 1,74 e 0,00 segundos, respectivamente.



(a) GAP



(b) Tempo de processamento

Figura 4.1: GAP e Tempo de processamento para os testes do modelo *q-routes*

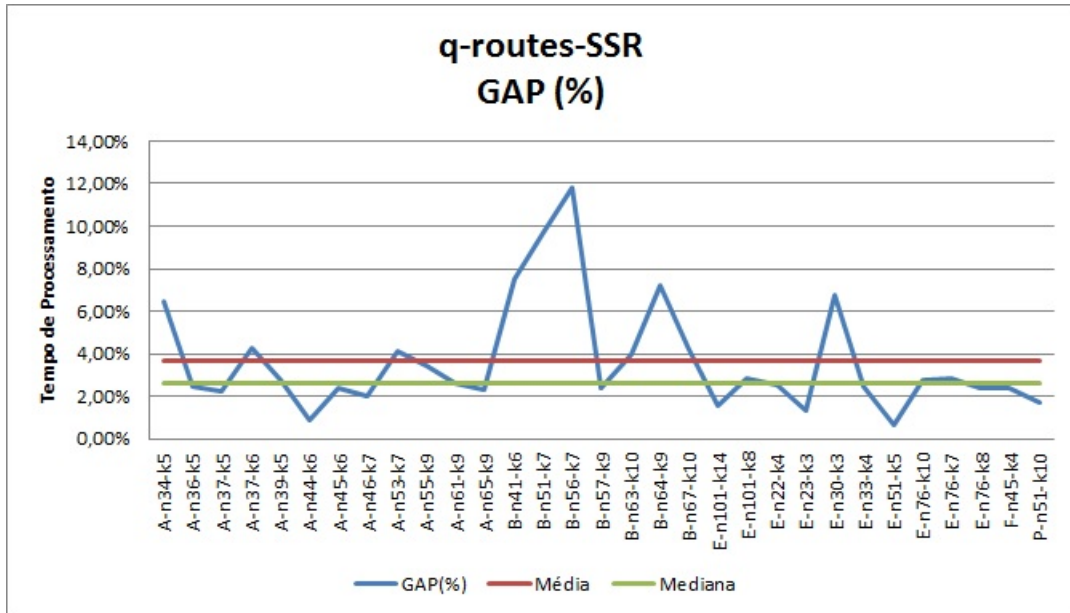


A Tabela 4.3 apresenta os resultados utilizando o subproblema *q-routes* com os cortes *SSR*. Seu esquema de funcionamento é representado pelo Algoritmo 6.

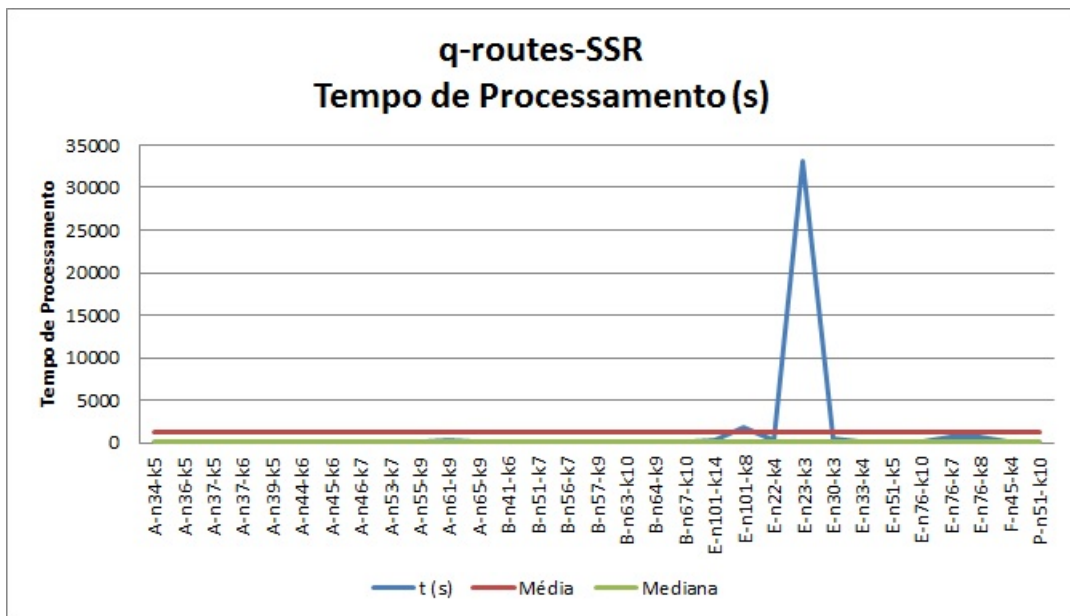
Tabela 4.3: Resultados para o modelo *q-routes-SSR*

Instância	LI	LS	GAP(%)	#cols	#SSR	#ext	#int	t (s)
A-n34-k5	727,937	778	6,43%	1.693	63	7	50	14
A-n36-k5	779,500	799	2,44%	4.420	180	20	20	147
A-n37-k5	654,045	669	2,24%	1.845	30	3	50	16
A-n37-k6	908,655	949	4,25%	1.416	30	3	38	10
A-n39-k5	799,439	822	2,74%	2.404	80	8	62	30
A-n44-k7	928,455	937	0,91%	1.990	22	2	45	16
A-n45-k6	921,594	944	2,37%	4.412	264	22	99	185
A-n46-k7	895,431	914	2,03%	3.433	144	12	75	85
A-n53-k7	968,684	1010	4,09%	4.434	112	8	84	88
A-n55-k9	1.036,030	1073	3,45%	2.934	98	7	53	53
A-n61-k9	1.006,980	1034	2,61%	5.478	272	17	90	250
A-n65-k9	1.147,240	1174	2,28%	4.242	85	5	65	77
B-n41-k6	766,672	829	7,52%	1.732	55	5	42	16
B-n51-k7	931,782	1032	9,71%	2.614	78	6	51	36
B-n56-k7	623,572	707	11,80%	3.423	70	5	61	48
B-n57-k9	1.559,280	1598	2,42%	3.713	150	10	65	93
B-n63-k10	1.436,500	1496	3,98%	3.802	80	5	60	57
B-n64-k9	798,968	861	7,20%	3.105	48	3	48	37
B-n67-k10	988,380	1032	4,23%	3.319	119	7	49	58
E-n101-k14	1.050,470	1067	1,55%	8.028	104	4	79	265
E-n101-k8	791,560	815	2,88%	17.116	260	10	170	1743
E-n22-k4	365,607	375	2,50%	1.142	90	15	53	297
E-n23-k3	561,512	569	1,32%	15.097	720	120	90	33150
E-n30-k3	497,807	534	6,78%	3.081	64	8	105	443
E-n33-k4	814,718	835	2,43%	1.672	36	4	51	108
E-n51-k5	517,630	521	0,65%	3.860	78	6	76	76
E-n76-k10	806,964	830	2,78%	4.520	95	5	59	102
E-n76-k7	662,778	682	2,82%	8.864	247	13	117	733
E-n76-k8	717,236	735	2,42%	8.266	266	14	109	693
F-n45-k4	706,758	724	2,38%	3.572	24	2	80	61
P-n51-k10	728,576	741	1,68%	1.920	65	5	37	23

Os Gráficos 4.2 *a* e *b* apresentam o *GAP* e *tempo de processamento*, respectivamente para cada uma das instâncias testadas. O aumento do tempo computacional gasto em alguns testes se deve a maior presença de variáveis com valores fracionários tendendo a 1 e com ciclos, o que gerava a inviabilidade das restrições *SSR* e *clique*. A média e a mediana do *GAP* foram 3,64% e 2,61%. Para o *tempo de processamento* 1258,39 e 77,00 segundos, respectivamente.



(a) GAP



(b) Tempo de processamento

Figura 4.2: GAP e Tempo de processamento para os testes do modelo *q-routes-SSR*

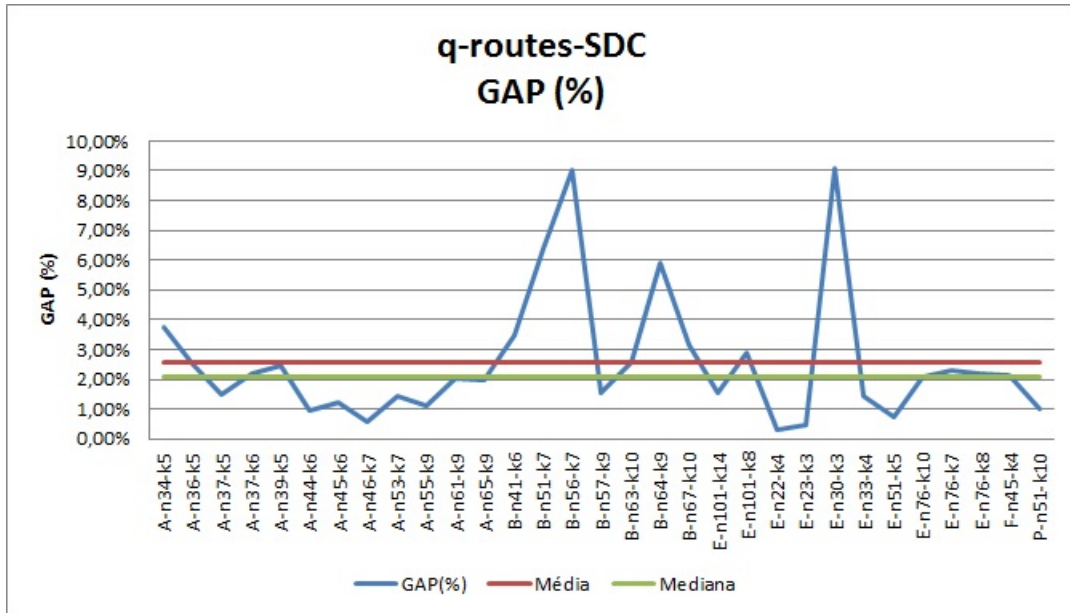
A Tabela 4.4 apresenta os resultados utilizando o subproblema *q-routes* com os cortes *SDC*. Seu esquema de funcionamento é representado pelo Algoritmo 7.

Tabela 4.4: Resultados para o modelo *q-routes-SDC*

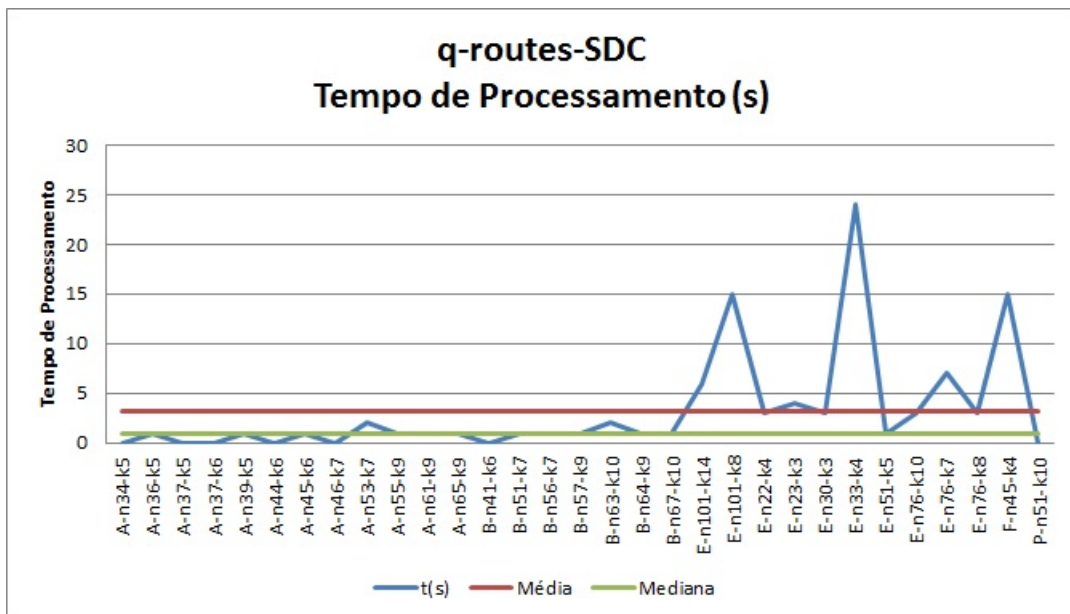
<b>Instância</b>	<b>LI</b>	<b>LS</b>	<b>GAP(%)</b>	<b>#cols</b>	<b>#SDC</b>	<b>#ext</b>	<b>#int</b>	<b>t(s)</b>
<b>A-n34-k5</b>	749,000	778	3,73%	2.023	10	11	60	0
<b>A-n36-k5</b>	778,765	799	2,53%	1.690	0	1	47	1
<b>A-n37-k5</b>	658,856	669	1,52%	2.097	5	6	57	0
<b>A-n37-k6</b>	928,357	949	2,18%	2.244	11	12	61	0
<b>A-n39-k5</b>	801,756	822	2,46%	1.910	3	4	49	1
<b>A-n44-k6</b>	927,923	937	0,97%	2.635	8	9	60	0
<b>A-n45-k6</b>	932,322	944	1,24%	3.312	14	15	74	1
<b>A-n46-k7</b>	908,550	914	0,60%	3.118	10	11	68	0
<b>A-n53-k7</b>	995,541	1010	1,43%	4.798	9	10	91	2
<b>A-n55-k9</b>	1.061,050	1073	1,11%	4.338	18	19	79	1
<b>A-n61-k9</b>	1.013,200	1034	2,01%	3.738	7	8	61	1
<b>A-n65-k9</b>	1.150,520	1174	2,00%	5.010	14	15	77	1
<b>B-n41-k6</b>	800,053	829	3,49%	2.772	14	15	68	0
<b>B-n51-k7</b>	965,889	1032	6,41%	3.714	13	14	73	1
<b>B-n56-k7</b>	643,000	707	9,05%	3.643	7	8	65	1
<b>B-n57-k9</b>	1.573,060	1598	1,56%	4.273	16	17	75	1
<b>B-n63-k10</b>	1.457,940	1496	2,54%	5.538	18	19	88	2
<b>B-n64-k9</b>	810,003	861	5,92%	4.869	17	18	76	1
<b>B-n67-k10</b>	999,299	1032	3,17%	4.837	14	15	72	1
<b>E-n101-k14</b>	1.050,470	1067	1,55%	10.528	16	17	104	6
<b>E-n101-k8</b>	791,364	815	2,90%	13.116	2	3	130	15
<b>E-n22-k4</b>	373,875	375	0,30%	638	7	8	29	3
<b>E-n23-k3</b>	566,333	569	0,47%	1.039	2	3	46	4
<b>E-n30-k3</b>	485,375	534	9,11%	1.776	0	1	60	3
<b>E-n33-k4</b>	822,774	835	1,46%	2.152	5	6	66	24
<b>E-n51-k5</b>	517,141	521	0,74%	3.710	4	5	73	1
<b>E-n76-k10</b>	812,572	830	2,10%	6.470	17	18	85	3
<b>E-n76-k7</b>	666,141	682	2,33%	8.264	6	7	109	7
<b>E-n76-k8</b>	718,869	735	2,19%	6.766	10	11	89	3
<b>F-n45-k4</b>	708,683	724	2,12%	4.320	1	2	97	15
<b>P-n51-k10</b>	733,337	741	1,03%	2.720	14	15	53	0

Os Gráficos 4.3 *a* e *b* apresentam o *GAP* e *tempo de processamento*, respectivamente para cada uma das instâncias testadas. A média e a mediana do *GAP* foram 2,59% e 2,10%. Para o *tempo de processamento* 3,20 e 1,00 segundos, respectivamente.

O principal ponto para este método, puro ou combinado com as restrições, é sua velocidade de convergência para um *LI*, mesmo para instâncias consideradas grandes pela literatura.



(a) GAP



(b) Tempo de processamento

Figura 4.3: GAP e Tempo de processamento para os testes do modelo *q-routes-SDC*

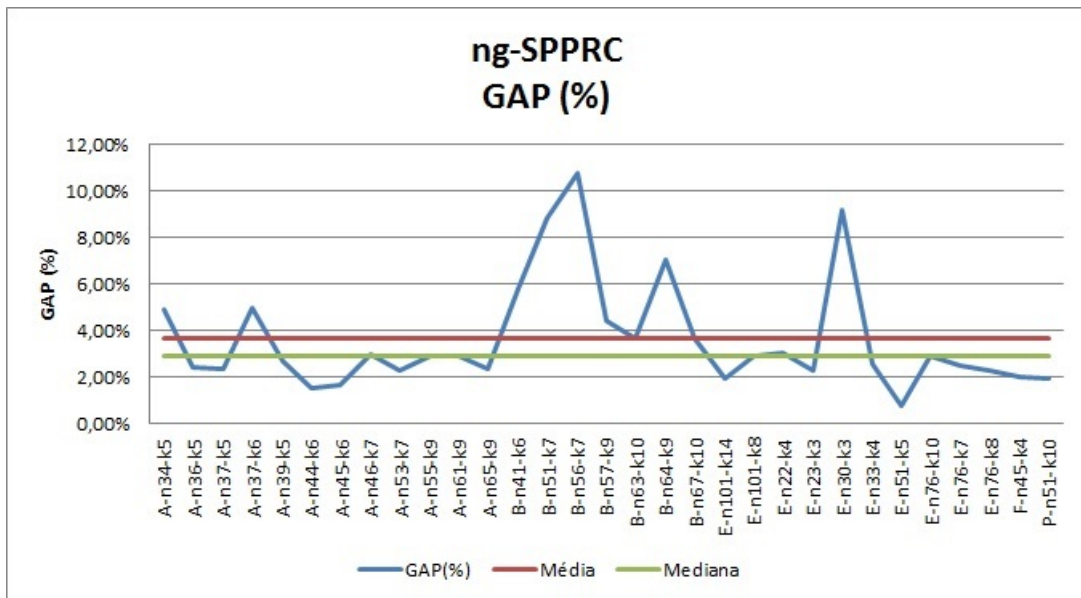
#### 4.2.2 Resultados utilizando o subproblema *ng-SPPRC*

A Tabela 4.5 apresenta os resultados utilizando o subproblema *ng-SPPRC* sem nenhum corte. Seu esquema de funcionamento pode ser representado pelas linhas 1 a 4 e 6 a 12 no Algoritmo 6, onde somente a estratégia *DSSR* é usada e na linha 7 o *PMR* é resolvido sem as *SDC* ou *SSR*.

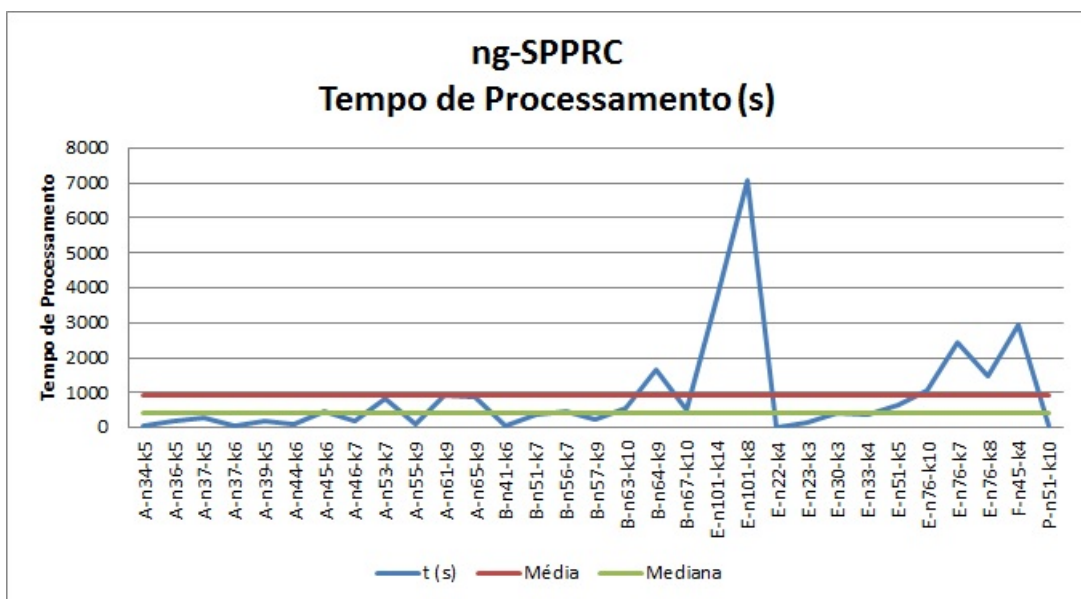
Tabela 4.5: Resultados para o modelo *ng-SPPRC*

Instância	LI	LS	GAP(%)	#cols	#iter	t (s)
<b>A-n34-k5</b>	739,878	778	4,90%	2.338	37	51
<b>A-n36-k5</b>	779,444	799	2,45%	4.444	55	200
<b>A-n37-k5</b>	653,250	669	2,35%	3.028	38	257
<b>A-n37-k6</b>	902,140	949	4,94%	2.036	34	41
<b>A-n39-k5</b>	799,604	822	2,72%	3.863	49	189
<b>A-n44-k6</b>	922,665	937	1,53%	3.343	49	114
<b>A-n45-k6</b>	928,450	944	1,65%	7.011	47	484
<b>A-n46-k7</b>	886,990	914	2,96%	3.325	47	187
<b>A-n53-k7</b>	986,768	1010	2,30%	7.468	67	842
<b>A-n55-k9</b>	1.041,800	1073	2,91%	3.196	50	113
<b>A-n61-k9</b>	1.004,100	1034	2,89%	8.754	70	917
<b>A-n65-k9</b>	1.146,080	1174	2,38%	7.231	64	897
<b>B-n41-k6</b>	781,012	829	5,79%	2.010	37	70
<b>B-n51-k7</b>	941,208	1032	8,80%	5.459	49	367
<b>B-n56-k7</b>	631,013	707	10,75%	4.378	54	460
<b>B-n57-k9</b>	1.527,150	1598	4,43%	3.247	46	231
<b>B-n63-k10</b>	1.441,660	1496	3,63%	5.404	63	547
<b>B-n64-k9</b>	800,164	861	7,07%	11.935	68	1671
<b>B-n67-k10</b>	994,525	1032	3,63%	4.809	57	488
<b>E-n101-k14</b>	1.046,420	1067	1,93%	17.330	103	3741
<b>E-n101-k8</b>	791,105	815	2,93%	31.066	144	7056
<b>E-n22-k4</b>	363,600	375	3,04%	923	17	3
<b>E-n23-k3</b>	556,000	569	2,28%	2.636	34	133
<b>E-n30-k3</b>	485,100	534	9,16%	6.738	46	406
<b>E-n33-k4</b>	813,604	835	2,56%	4.505	46	376
<b>E-n51-k5</b>	517,017	521	0,76%	7.658	66	634
<b>E-n76-k10</b>	806,080	830	2,88%	8.269	69	1081
<b>E-n76-k7</b>	665,233	682	2,46%	2.452	94	2452
<b>E-n76-k8</b>	718,375	735	2,26%	8.948	80	1486
<b>F-n45-k4</b>	709,438	724	2,01%	20.004	87	2932
<b>P-n51-k10</b>	726,476	741	1,96%	2.855	39	58

Os Gráficos 4.4 *a* e *b* apresentam o *GAP* e *tempo de processamento*, respectivamente para cada uma das instâncias testadas. A média e a mediana do *GAP* foram 3,62% e 2,88%. Para o *tempo de processamento* 918,8 e 406,0 segundos, respectivamente.



(a) GAP



(b) Tempo de processamento

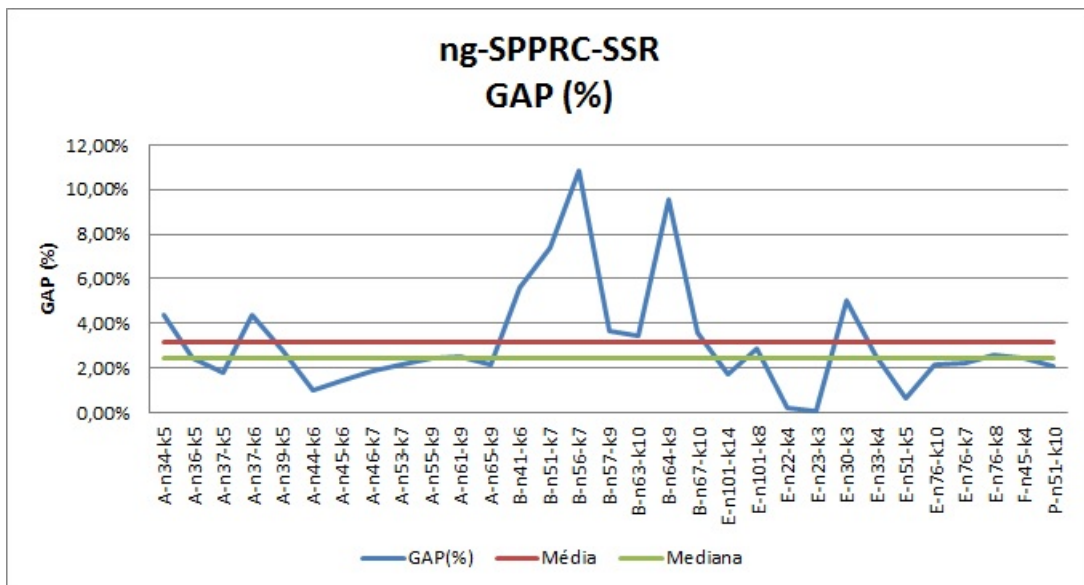
Figura 4.4: GAP e Tempo de processamento para os testes do modelo *ng-SPPRC*

A Tabela 4.6 apresenta os resultados utilizando o subproblema  $q$ -routes com os cortes  $SSR$ . Seu esquema de funcionamento é representado pelo Algoritmo 6.

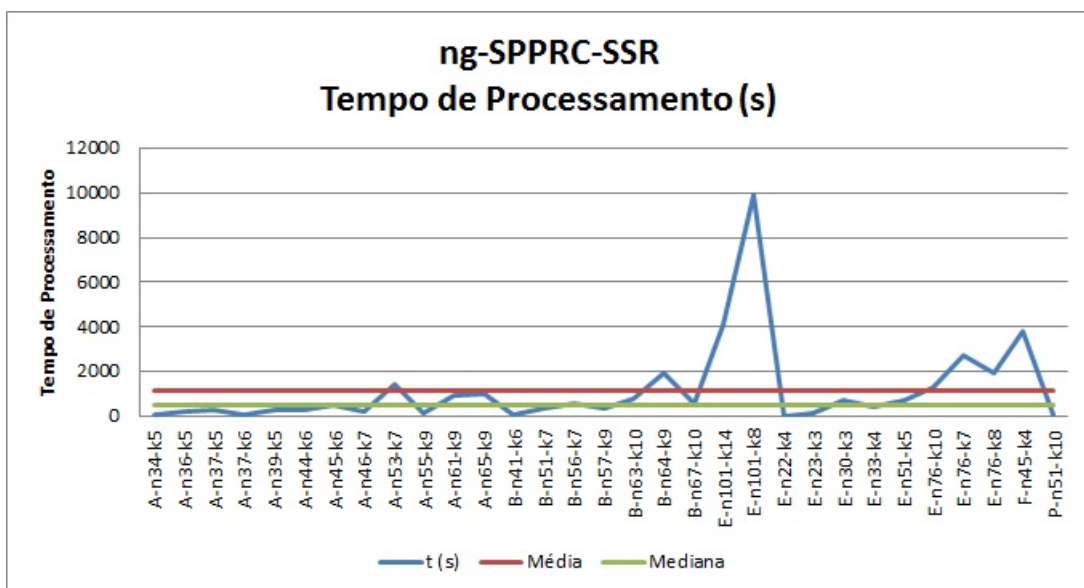
Tabela 4.6: Resultados para o modelo  $ng$ - $SPPRC$ - $SSR$

Instância	LI	UB	GAP(%)	#cols	#SSR	#ext	#int	t (s)
A-n34-k5	744,092	778	4,36%	2.561	27	3	45	75
A-n36-k5	779,500	799	2,44%	4.268	36	4	74	253
A-n37-k5	656,968	669	1,80%	3.002	20	5	48	270
A-n37-k6	907,618	949	4,36%	2.448	40	4	54	74
A-n39-k5	798,970	822	2,80%	4.958	20	2	62	302
A-n44-k7	927,866	937	0,97%	4.975	209	19	116	288
A-n45-k6	930,616	944	1,42%	7.792	60	5	60	542
A-n46-k7	896,656	914	1,90%	3.669	60	6	65	247
A-n53-k7	988,537	1010	2,13%	9.384	294	21	130	1474
A-n55-k9	1.046,710	1073	2,45%	3.406	56	4	57	155
A-n61-k9	1.008,130	1034	2,50%	8.423	64	4	79	972
A-n65-k9	1.148,410	1174	2,18%	8.259	34	2	78	1039
B-n41-k6	782,715	829	5,58%	2.308	110	10	54	109
B-n51-k7	955,606	1032	7,40%	4.962	26	2	51	366
B-n56-k7	630,282	707	10,85%	4.728	28	2	63	594
B-n57-k9	1.538,940	1598	3,70%	3.836	75	5	73	331
B-n63-k10	1.444,680	1496	3,43%	5.796	128	8	79	821
B-n64-k9	778,952	861	9,53%	11.255	64	4	79	1950
B-n67-k10	994,826	1032	3,60%	4.587	34	2	57	564
E-n101-k14	1.048,310	1067	1,75%	16.944	52	2	102	4077
E-n101-k8	791,692	815	2,86%	33.739	182	7	168	9922
E-n22-k4	374,098	375	0,24%	1.787	30	5	42	16
E-n23-k3	568,500	569	0,09%	2.843	24	4	50	159
E-n30-k3	507,190	534	5,02%	9.242	72	9	93	708
E-n33-k4	813,500	835	2,57%	4.542	36	4	56	428
E-n51-k5	517,660	521	0,64%	7.084	91	7	85	714
E-n76-k10	812,320	830	2,13%	8.200	76	4	76	1273
E-n76-k7	666,584	682	2,26%	11.261	76	4	139	2740
E-n76-k8	716,068	735	2,58%	9.968	133	7	109	1970
F-n45-k4	706,373	724	2,43%	19.955	36	3	103	3834
P-n51-k10	725,750	741	2,06%	2.752	52	4	42	83

Os Gráficos 4.5 *a* e *b* apresentam o  $GAP$  e *tempo de processamento*, respectivamente para cada uma das instâncias testadas. Neste caso, o uso das  $SSR$  também provocou aumento do tempo computacional. A média e a mediana do  $GAP$  foram 3,16% e 2,45%. Para o *tempo de processamento* 1172,6 e 542,0 segundos, respectivamente.



(a) GAP



(b) Tempo de processamento

Figura 4.5: GAP e Tempo de processamento para os testes do modelo *ng-SPPRC-SSR*

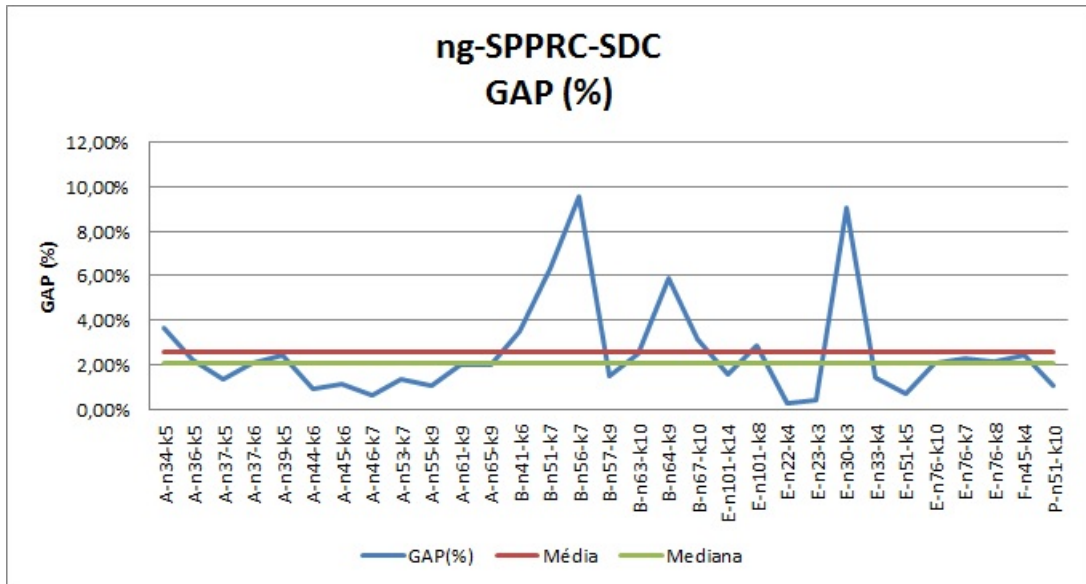


A Tabela 4.7 apresenta os resultados utilizando o subproblema *ng-SPPRC* com os cortes *SDC*. Seu esquema de funcionamento é representado pelo Algoritmo 7.

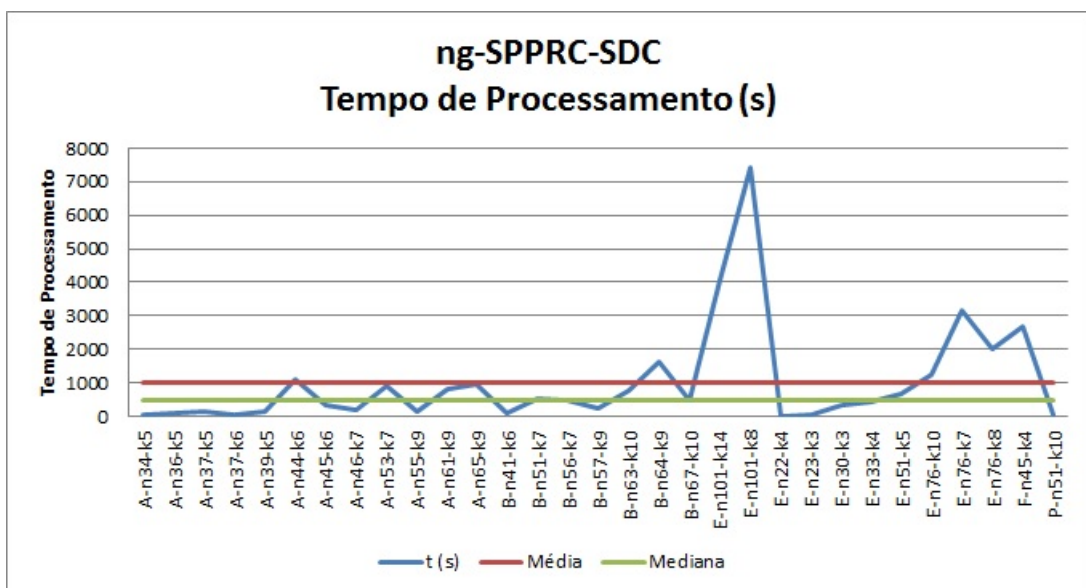
Tabela 4.7: Resultados para o modelo *ng-SPPRC-SDC*

Instância	LI	UB	GAP(%)	#cols	#SDC	#ext	#int	t (s)
<b>A-n34-k5</b>	749,707	778	3,64%	2.836	5	6	58	50
<b>A-n36-k5</b>	781,184	799	2,23%	4.154	0	1	64	122
<b>A-n37-k5</b>	660,000	669	1,35%	3.281	2	3	52	139
<b>A-n37-k6</b>	929,323	949	2,07%	2.816	7	8	68	61
<b>A-n39-k5</b>	801,900	822	2,45%	4.520	2	3	71	160
<b>A-n44-k7</b>	928,068	937	0,95%	3.695	4	5	66	1115
<b>A-n45-k6</b>	933,440	944	1,12%	7.206	3	4	62	343
<b>A-n46-k7</b>	908,351	914	0,62%	4.222	4	5	66	200
<b>A-n53-k7</b>	995,845	1010	1,40%	8.271	9	10	113	893
<b>A-n55-k9</b>	1.061,500	1073	1,07%	3.716	4	5	67	159
<b>A-n61-k9</b>	1.013,230	1034	2,01%	8.148	1	2	77	794
<b>A-n65-k9</b>	1.150,150	1174	2,03%	8.723	4	5	99	965
<b>B-n41-k6</b>	799,649	829	3,54%	2.506	5	6	56	104
<b>B-n51-k7</b>	966,778	1032	6,32%	6.897	3	4	69	517
<b>B-n56-k7</b>	639,475	707	9,55%	4.763	2	3	77	498
<b>B-n57-k9</b>	1.573,640	1598	1,52%	4.069	11	12	82	264
<b>B-n63-k10</b>	1.458,220	1496	2,53%	6.426	7	8	95	760
<b>B-n64-k9</b>	810,023	861	5,92%	13.992	2	3	85	1613
<b>B-n67-k10</b>	999,549	1032	3,14%	5.705	1	2	74	493
<b>E-n101-k14</b>	1.050,450	1067	1,55%	18.887	4	5	129	4069
<b>E-n101-k8</b>	791,296	815	2,91%	37.431	2	3	147	7425
<b>E-n22-k4</b>	373,875	375	0,30%	1.041	0	1	23	3
<b>E-n23-k3</b>	566,500	569	0,44%	2.301	0	1	37	42
<b>E-n30-k3</b>	485,750	534	9,04%	6.807	0	1	47	338
<b>E-n33-k4</b>	822,774	835	1,46%	4.926	3	4	64	425
<b>E-n51-k5</b>	517,275	521	0,71%	8.563	2	3	75	669
<b>E-n76-k10</b>	812,566	830	2,10%	9.103	5	6	87	1227
<b>E-n76-k7</b>	666,129	682	2,33%	14.522	0	1	109	3183
<b>E-n76-k8</b>	719,224	735	2,15%	11.345	2	3	100	1999
<b>F-n45-k4</b>	706,500	724	2,42%	21.570	0	1	79	2689
<b>P-n51-k10</b>	732,955	741	1,09%	3.126	3	4	52	55

Os Gráficos 4.5 *a* e *b* apresentam o *GAP* e *tempo de processamento*, respectivamente para cada uma das instâncias testadas. A média e a mediana do *GAP* foram 2,58% e 2,07%. Para o *tempo de processamento* 1012,1 e 493,0 segundos, respectivamente.



(a) GAP



(b) Tempo de processamento

Figura 4.6: GAP e Tempo de processamento para os testes do modelo *ng-SPPRC-SDC*

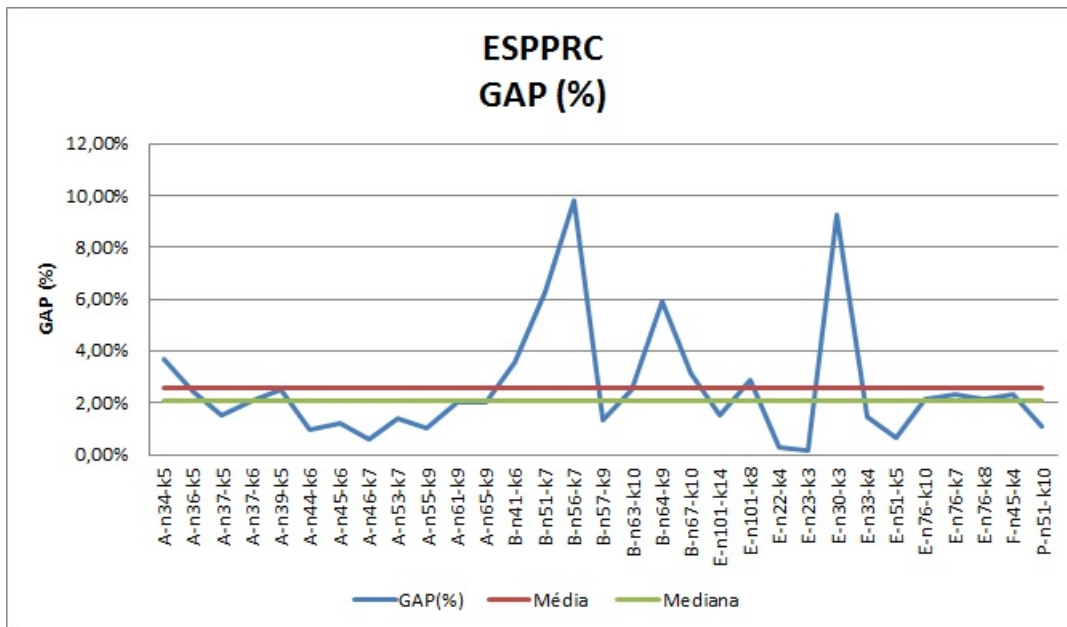
### 4.2.3 Resultados utilizando o subproblema *ESPPRC*

A Tabela 4.8 apresenta os resultados utilizando o subproblema *ESPPRC* utilizando o *esquema* apresentado pelo Algoritmo 5 sem o uso das restrições *SSR*.

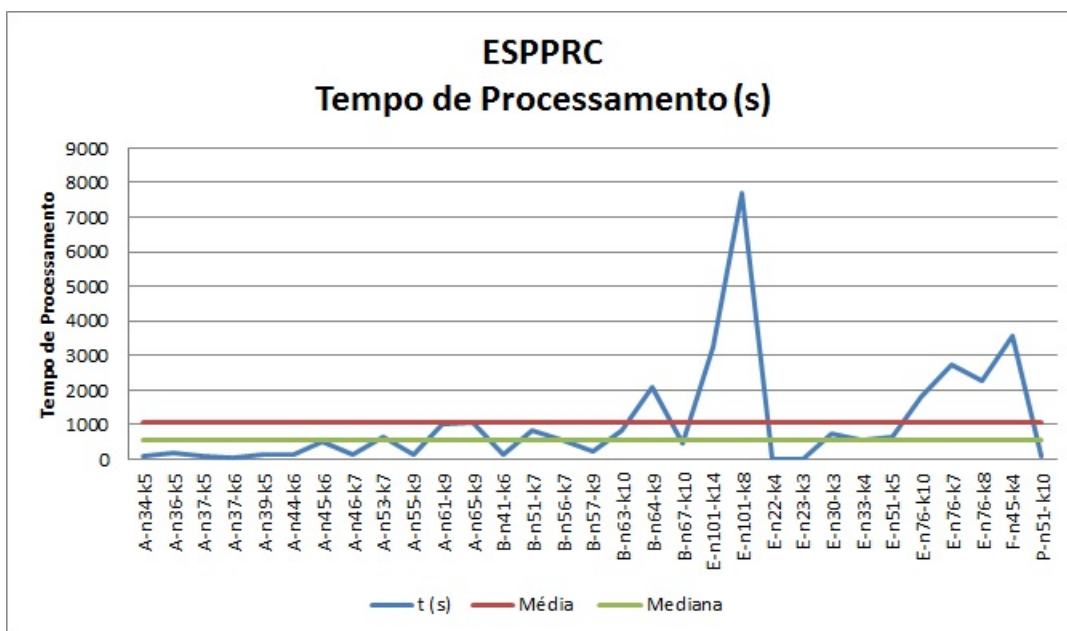
Tabela 4.8: Resultados para o modelo *ESPPRC*

<b>Instância</b>	<b>LI</b>	<b>LS</b>	<b>GAP(%)</b>	<b>#cols</b>	<b>#int</b>	<b>t (s)</b>
<b>A-n34-k5</b>	749,072	778	3,72%	2.354	42	62
<b>A-n36-k5</b>	779,444	799	2,45%	4.837	50	175
<b>A-n37-k5</b>	658,826	669	1,52%	3.161	42	93
<b>A-n37-k6</b>	929,548	949	2,05%	2.433	42	46
<b>A-n39-k5</b>	801,556	822	2,49%	4.022	57	139
<b>A-n44-k6</b>	927,766	937	0,99%	4.389	57	147
<b>A-n45-k6</b>	932,324	944	1,24%	7.260	53	487
<b>A-n46-k7</b>	908,817	914	0,57%	3.709	48	120
<b>A-n53-k7</b>	995,814	1010	1,40%	7.208	60	647
<b>A-n55-k9</b>	1.061,750	1073	1,05%	3.581	48	129
<b>A-n61-k9</b>	1.013,270	1034	2,00%	9.652	76	999
<b>A-n65-k9</b>	1.150,150	1174	2,03%	8.735	72	1053
<b>B-n41-k6</b>	799,674	829	3,54%	2.644	49	114
<b>B-n51-k7</b>	967,200	1032	6,28%	8.798	75	836
<b>B-n56-k7</b>	637,750	707	9,79%	5.307	68	528
<b>B-n57-k9</b>	1.576,910	1598	1,32%	4.560	54	218
<b>B-n63-k10</b>	1.458,560	1496	2,50%	6.275	74	816
<b>B-n64-k9</b>	809,804	861	5,95%	15.330	82	2087
<b>B-n67-k10</b>	999,807	1032	3,12%	5.716	63	456
<b>E-n101-k14</b>	1.050,470	1067	1,55%	10.085	183	3244
<b>E-n101-k8</b>	791,338	815	2,90%	27.079	218	7706
<b>E-n22-k4</b>	373,875	375	0,30%	1.026	23	3
<b>E-n23-k3</b>	568,000	569	0,18%	2.233	43	10
<b>E-n30-k3</b>	484,542	534	9,26%	10.236	52	744
<b>E-n33-k4</b>	822,774	835	1,46%	5.070	48	524
<b>E-n51-k5</b>	517,646	521	0,64%	9.781	63	643
<b>E-n76-k10</b>	812,012	830	2,17%	12.555	82	1820
<b>E-n76-k7</b>	665,936	682	2,36%	15.035	99	2718
<b>E-n76-k8</b>	719,095	735	2,16%	13.628	89	2243
<b>F-n45-k4</b>	707,000	724	2,35%	35.589	90	3585
<b>P-n51-k10</b>	732,985	741	1,08%	3.224	42	72

Os Gráficos 4.7 *a* e *b* apresentam o *GAP* e *tempo de processamento*, respectivamente para cada uma das instâncias testadas. Algo que se faz notar é que quanto maior a capacidade permitida de cada rota da instância, maior o número de nós e menor a capacidade de rotas, mais difícil foi a convergência. A média e a mediana do *GAP* foram 2,59% e 2,05%. Para o *tempo de processamento* 1047,2 e 524,0 segundos, respectivamente.



(a) GAP



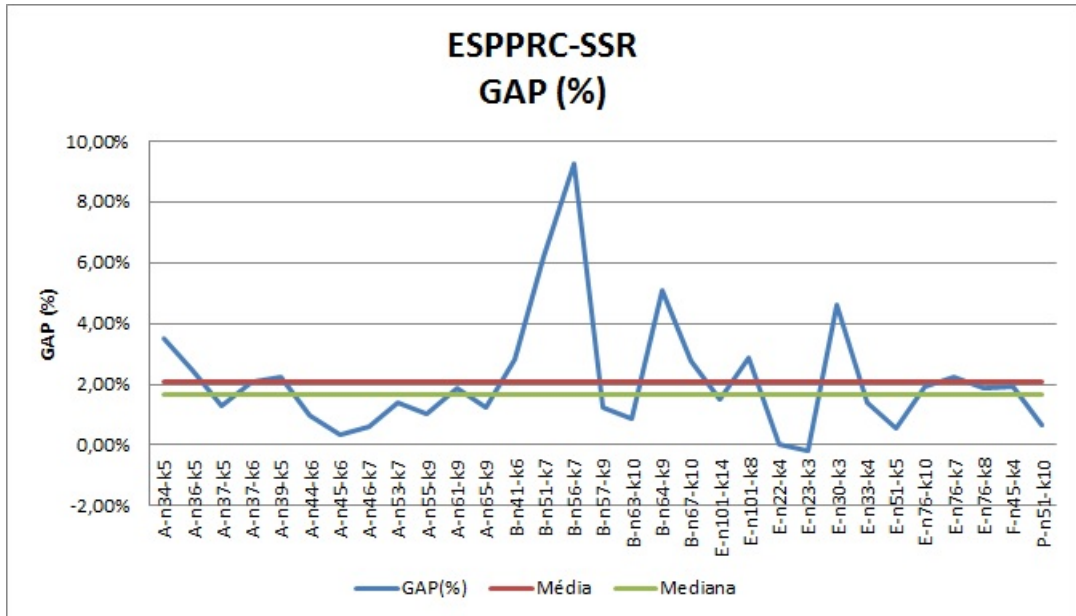
(b) Tempo de processamento

Figura 4.7: GAP e Tempo de processamento para os testes do modelo *ESPPRC*

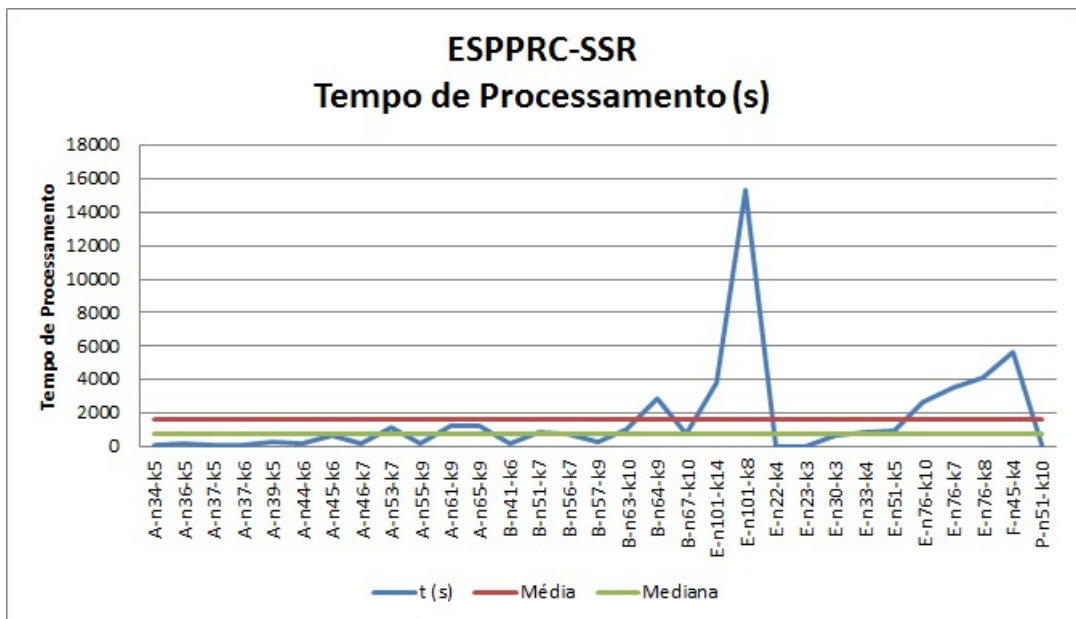
Tabela 4.9: Resultados para o modelo *ESPPRC-SSR*

Instância	LI	LS	GAP(%)	#cols	#SSR	#ext	#int	t (s)
<b>A-n34-k5</b>	750,800	778	3,50%	3.195	63	7	62	87
<b>A-n36-k5</b>	779,500	799	2,44%	5.036	27	3	64	225
<b>A-n37-k5</b>	660,233	669	1,31%	3.404	30	3	57	113
<b>A-n37-k6</b>	929,367	949	2,07%	2.634	50	5	51	77
<b>A-n39-k5</b>	803,389	822	2,26%	4.530	150	150	83	271
<b>A-n44-k7</b>	927,908	937	0,97%	3.821	22	2	55	159
<b>A-n45-k6</b>	940,898	944	0,33%	8.619	36	3	64	682
<b>A-n46-k7</b>	908,300	914	0,62%	3.887	24	2	49	170
<b>A-n53-k7</b>	995,758	1010	1,41%	8.368	140	10	10	1119
<b>A-n55-k9</b>	1.061,800	1073	1,04%	3.661	56	4	61	198
<b>A-n61-k9</b>	1.014,500	1034	1,89%	10.029	144	9	97	1281
<b>A-n65-k9</b>	1.159,450	1174	1,24%	9.254	51	3	79	1265
<b>B-n41-k6</b>	805,642	829	2,82%	3.032	22	2	58	193
<b>B-n51-k7</b>	967,987	1032	6,20%	8.004	26	2	69	843
<b>B-n56-k7</b>	641,382	707	9,28%	5.630	84	6	77	761
<b>B-n57-k9</b>	1.578,640	1598	1,21%	4.719	75	5	70	303
<b>B-n63-k10</b>	1.482,650	1496	0,89%	6.962	32	2	80	1044
<b>B-n64-k9</b>	817,266	861	5,08%	17.152	48	3	85	2831
<b>B-n67-k10</b>	1.003,700	1032	2,74%	7.103	102	6	99	746
<b>E-n101-k14</b>	1.050,870	1067	1,51%	17.580	78	3	93	3865
<b>E-n101-k8</b>	791,642	815	2,87%	43.257	624	24	208	15346
<b>E-n22-k4</b>	375,000	375	0,00%	1.407	24	4	37	11
<b>E-n23-k3*</b>	570,000	569	-0,18%	2.140	12	2	39	33
<b>E-n30-k3</b>	509,400	534	4,61%	9.609	24	3	62	715
<b>E-n33-k4</b>	823,211	835	1,41%	5.732	27	3	50	865
<b>E-n51-k5</b>	518,253	521	0,53%	10.004	78	6	86	979
<b>E-n76-k10</b>	814,133	830	1,91%	13.781	304	16	127	2709
<b>E-n76-k7</b>	666,778	682	2,23%	17.011	114	6	107	3562
<b>E-n76-k8</b>	721,375	735	1,85%	17.708	494	26	186	4104
<b>F-n45-k4</b>	710,129	724	1,92%	28.234	72	6	94	5669
<b>P-n51-k10</b>	736,044	741	0,67%	3.237	52	4	54	114

Os Gráficos 4.8 *a* e *b* apresentam o *GAP* e *tempo de processamento*, respectivamente para cada uma das instâncias testadas. O alto tempo computacional gasto em alguns testes se deve a maior presença de variáveis com valores fracionários tendendo a 1, o que gerava a inviabilidade das retrições *SSR*. O A média e a mediana para o *GAP* foram 2,10% e 1,68%. Para o *tempo de processamento* 1675,1 e 753,5 segundos, respectivamente. Quando comparado ao método *ESPPRC* fica notável a curva de *trade-off* entre melhores limites e tempo de processamento.



(a) GAP



(b) Tempo de processamento

Figura 4.8: GAP e Tempo de processamento para os testes do modelo *ESPPRC-SSR*

### 4.3 Análise dos resultados

Os resultados dos experimentos serão analisados nesta seção. A Tabela 4.10 apresenta um resumo geral.

Tabela 4.10: Resumo dos Resultados

		<i>q-routes</i>			<i>ng-SPPRC</i>			<i>ESPPRC</i>	
		<i>puro</i>	<i>SSR</i>	<i>SDC</i>	<i>puro</i>	<i>SSR</i>	<i>SDC</i>	<i>puro</i>	<i>SSR</i>
<b>GAP (%)</b>	<b>média</b>	4,08%	3,64%	2,59%	3,62%	3,16%	2,58%	2,59%	2,10%
	<b>mediana</b>	2,96%	2,61%	2,10%	2,88%	2,45%	2,07%	2,05%	1,68%
<b>Tempo (s)</b>	<b>média</b>	1,74	1258,39	3,2	918,8	1172,6	1012,1	1047,2	1675,1
	<b>mediana</b>	0,00	77,00	1,0	406,0	542,0	493,0	524,0	753,5

Comparar os critérios *limite* e *tempo* simultaneamente é algo difícil, pois a curva de troca para ambos é inversamente atraente. Enquanto busca-se menor *tempo de processamento* fica mais difícil melhorar o *limite inferior*. Quando se busca melhorar o *limite inferior*, necessita-se de mais *tempo* para gerar colunas melhores para o problema. Dessa maneira, eles serão analisados separadamente à seguir.

#### Limites (*GAP*(%)):

- Comparando os métodos que permitem ciclos, o *ng-SPPRC* obteve melhores limites. Isso provavelmente é consequência do uso de políticas de dominância mais complexas, que geram melhores variáveis, do que a simples comparação de custos reduzidos utilizada pelo método das *q-routes*;
- Os métodos combinados com as *SSR* mostraram melhoria nos limites obtidos em relação aos métodos *não-elementares*. Dentre os 3 testados, o *ESPPRC* obteve os melhores, com média de 2,10%. Isso se dá pelo fato de que todas as colunas geradas são elementares e assim, mais compatíveis com a estrutura do problema fixadas pelas restrições (3.2);
- Quando se compara os métodos *q-routes-SDC vs. ng-SPPRC-SDC vs. ESPPRC (puro)*, sendo os limites obtidos elementares, o *GAP* obtido é praticamente o mesmo, o que representa que a qualidade dos limites são equiparáveis.

#### Tempo de Processamento:

- O *esquema* que utiliza o subproblema das *q-routes* foi superior em todas as combinações;
- O *esquema* usando *ng-SPPRC* foi melhor do que o *ESPPRC* quando compara-se *ng-SPPRC-SDC versus ESPPRC* e *ng-SPPRC-SSR versus ESPPRC-SSR*;
- O uso das *SSR* piorou o *tempo de execução* para todos os métodos. O que apresentou melhor média utilizando estes cortes foi o *ng-SPPRC*, porém observando a mediana, o método das *q-routes* tem um valor melhor e mais coerente;
- Usar as *SDC* piorou muito pouco o tempo total de execução do *ng-SPPRC* (cerca de 10%). Quando combinado com as *q-routes*, piorou cerca de 83%, mas que é irrisório quando pensa-se em deixar de gastar 1 segundo e passar a gastar aproximadamente 3 segundos numa solução *elementar*.

A análise realizada sobre os testes leva a conclusão de que a combinação *q-routes-SDC* é a mais vantajosa. Ela é capaz de oferecer os mesmos *limites elementares* que os demais métodos

com o tempo computacional muito melhor (inferior). Isso pode ser reforçado com a análise da Tabela 4.11 feita à seguir.

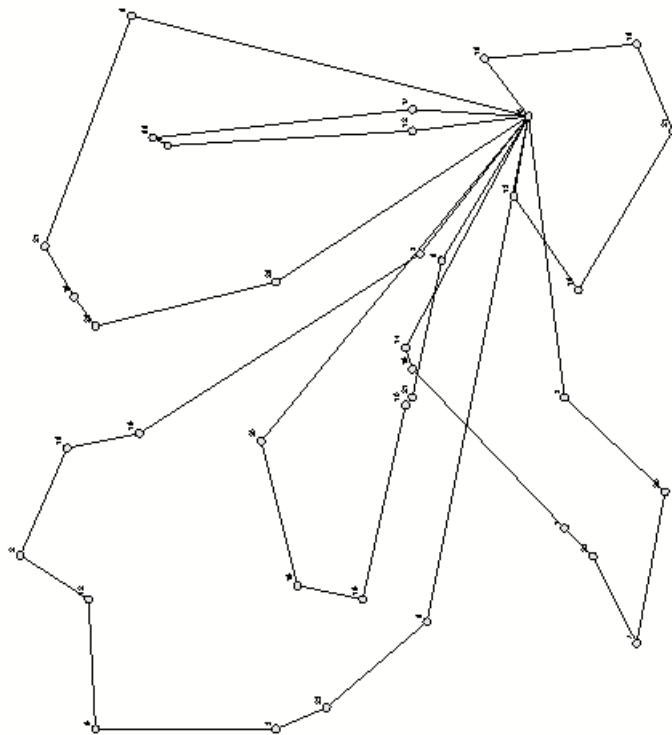
Algumas instâncias (**P** e **F**) de grande porte não foram incluídas na análise dos resultados, pois os alguns testes as utilizando não conseguiram atingir um *limite inferior* adequado, ficando estagnadas na solução inicial provida pela Heurística de Clarke e Wright (1964). Isso só ocorreu para os algoritmos do *Problema do Caminho Mínimo com Restrições de Recursos*, sendo o *elementar* ou não. Ao contrário, o método das *q-routes*, tanto *puro* quanto combinado com as restrições estudadas, conseguiu oferecer bons limites inferiores para estas instâncias. Na Tabela 4.11 são apresentados os resultados atingidos utilizando o subproblema das *q-routes* combinado com as restrições *SDC*, pois é o cenário que mais interessa para trabalhos futuros. A combinação mostrou robustez até mesmo para as instâncias de Fisher (1994), conseguindo um bom *LI* para a *F-n135-k7* e a otimalidade para a *F-n72-k4*.

Tabela 4.11: Instâncias que obtiverem *LI* somente usando o subproblema das *q-routes*

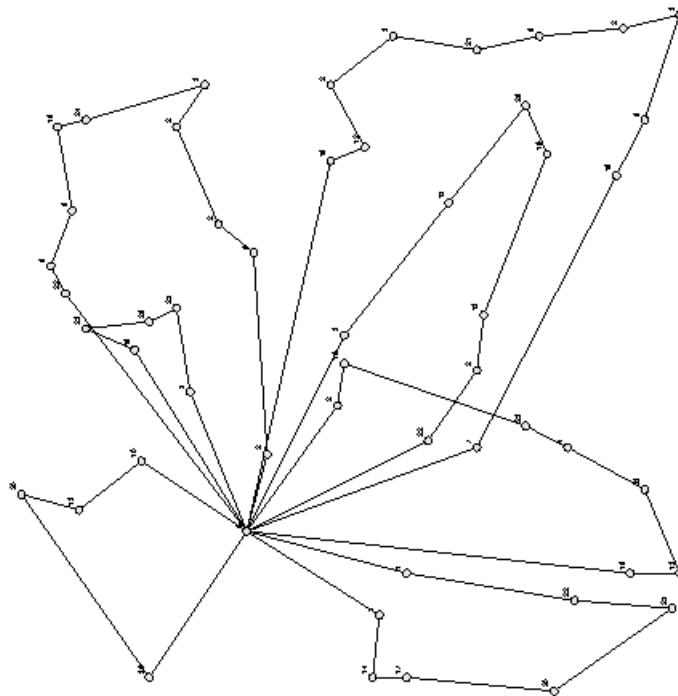
Instância	LI	LS	q-routes-SDC					t(s)
			GAP(%)	#Cols	#SDC	#ext	#int	
<b>F-n135-k7</b>	1139,39	1162	1,95%	94350	7	8	703	2050
<b>F-n72-k4</b>	237	237	0,00%	20171	0	1	283	2655
<b>P-n101-k4</b>	672,054	681	1,31%	39108	0	1	390	222
<b>P-n76-k4</b>	587,594	593	0,91%	17183	2	3	228	31
<b>P-n76-k5</b>	617,208	627	1,56%	11935	4	5	158	14

Por fim, a ferramenta de aceleração *DSSR* se mostrou muito útil, auxiliando na construção progressiva de rotas *elementares*. Em testes com instâncias maiores, sem o uso da técnica, o tempo computacional era bem maior do que os obtidos, assim como a necessidade de um número maior de *SDC*. Como exemplo, para a instância *P-n101-k4* sem o *DSSR* foram chamados 90 cortes, com 283.408 colunas geradas, *GAP* de 1,30% e 6100 segundos de execução. Com a *DSSR*, nenhum corte foi necessário, 39.108 colunas foram geradas, o tempo de execução foi de 222 segundos e o *GAP* praticamente o mesmo, 1,31%.





(a) Rotas geradas para a instância A-n37-k6 composta por 36 nós clientes, 6 rotas de capacidade igual a 100 unidades



(b) Rotas geradas para a instância A-n39-k7 composta por 38 nós clientes, 7 rotas de capacidade igual a 100 unidades

Figura 4.9: Rotas ótimas para problemas da literatura. Fonte Augerat (1995).

## Capítulo 5

# Conclusões e Propostas Futuras

“Se você pode sonhar algo, você pode realizá-lo” (Walt Disney)

### 5.1 Conclusões

Solucionar o Problema de Roteamento de Transporte Escolar Rural é um grande avanço na conquista de qualidade de vida da população rural, visto que no contexto brasileiro e seu processo histórico, existe uma desigualdade imensa causada pelas poucas oportunidades oferecidas à esta importante parcela da população. Devido a inacessibilidade de diversos recursos, também não tem nem mesmo meios de clamar por seus direitos básicos, entre eles o acesso a educação de qualidade.

Com base na necessidade de auxiliar na melhoria do acesso e criação de rotas para o Transporte Escolar Rural, este trabalho aborda através do método *Column-and-cut*, diferentes maneiras de se obter um *limite inferior* para o *Problema de Roteamento de Veículos Capacitado*. O *problema mestre (PMR)* escolhido foi o por *particionamento de conjunto*. Para gerar as colunas que seriam adicionadas ao *PMR* trabalhou-se com três subproblemas: o problema de Precificação Simples (*q-routes*), o Problema do Caminho Mínimo com Restrição de Recursos (*SPPRC*) em suas formas *elementar e não-elementar*. Com o propósito de melhorar os limites obtidos, duas famílias de *cortes* foram utilizadas, as *Restrições sobre Subconjunto de Linhas (SSR)* e *Restrições Fortes de Conexão (SDC)*.

Com os testes realizados, alguns pontos se fazem notáveis. Não se encontra na literatura sobre a real dificuldade em se resolver o *SPPRC*, *elementar* ou não, utilizando somente a capacidade como recurso que dita a viabilidade de um novo rótulo. A maioria dos trabalhos sempre envolve a combinação de *janelas de tempo* ou *coleta e entrega* com a *capacidade* como forma de tornar mais *afiada* a política de dominância do algoritmo. Esse problema com a convergência só foi identificado quando iniciou-se os testes com instâncias de grande porte. As restrições utilizadas se mostraram bastante promissoras, melhorando significativamente a qualidade dos limites obtidos quando comparadas com modelos que utilizaram os subproblemas *puros*. A técnica *DSSR* também foi de muita importância quando conjugada com os subproblemas *não-elementares*, facilitando a formação gradativa de rotas *elementares*.

Dentre as alternativas testadas a que se mostrou mais promissora e equilibrada, na curva de *trade-off* entre *tempo de processamento* e *limites obtidos*, foi a combinação entre *q-routes* e *SDC*. Os tempos de execução foram muito menores do que os outros dois métodos com a qualidade dos limites equiparáveis. Além disso, o método das *q-routes* foi o único que conseguiu atingir um *limite inferior* adequado para um conjunto de instâncias de grande porte, o que é de extrema valia, dada a necessidade de aplicação em problemas de porte equivalente aos testados ou maiores.

## 5.2 Propostas Futuras

- Implementar um algoritmo *Branch-Price-and-Cut* (DESROSIERS; LÜBBECKE, 2011), utilizando as *q-routes* e as *SDC*, pois foi a combinação que se mostrou mais produtiva e buscar a resposta ótima para algumas instâncias em aberto da literatura;
- Inclusão de novos elementos na política de dominância nos algoritmos do *Problema do Caminho Mínimo com Restrição de Recursos*, como a proposta por Jepsen et al. (2008) e também o estudo da introdução do conceito do número de ciclos por rótulos na dominância;
- Heurísticas para auxiliar durante a geração de colunas e no provimento de uma solução inicial viável melhor;
- Implementar a eliminação de *2-ciclos* ou até mesmo de *k-ciclos* para verificar se é capaz de melhorar os resultados antes de usar o *branch-and-bound*;
- Implementar um novo subproblema, o *ng-routes*, proposto por Baldacci, Mingozzi e Roberti (2011), e verificar se é possível oferecer melhores limites e tempo computacional que os obtidos por este trabalho;
- Viabilizar testes com instâncias sobre dados reais e *clusterizar* a demanda através de algum método heurístico.

# Referências Bibliográficas

- ALBERT, S. *Solving mixed integer linear programs using branch and cut algorithm*. Tese (Doutorado) — North Carolina State University, 2006.
- ARANTES, C. O. *Planejamento de rede escolar questões teóricas e metodológicas*. [S.l.]: Cedate, 1986.
- AUGERAT, P. *Approche polyédrale du problème de tournées de véhicules*. Tese (Doutorado) — Institut National Polytechnique de Grenoble-INPG, 1995.
- BALAS, E. Some valid inequalities for the set partitioning problem. *Annals of Discrete Mathematics*, Elsevier, v. 1, p. 13–47, 1977.
- BALAS, E. The prize collecting traveling salesman problem. *Networks*, Wiley Online Library, v. 19, n. 6, p. 621–636, 1989.
- BALAS, E.; CERIA, S.; CORNUÉJOLS, G. Mixed 0-1 programming by lift-and-project in a branch-and-cut framework. *Management Science*, INFORMS, v. 42, n. 9, p. 1229–1246, 1996.
- BALAS, E. et al. Gomory cuts revisited. *Operations Research Letters*, Elsevier, v. 19, n. 1, p. 1–9, 1996.
- BALDACCI, R.; HADJICONSTANTINO, E.; MINGOZZI, A. An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation. *Operations Research*, INFORMS, v. 52, n. 5, p. 723–738, 2004.
- BALDACCI, R.; MINGOZZI, A.; ROBERTI, R. New route relaxation and pricing strategies for the vehicle routing problem. *Operations research*, INFORMS, v. 59, n. 5, p. 1269–1283, 2011.
- BALDACCI, R.; TOTH, P.; VIGO, D. Recent advances in vehicle routing exact algorithms. *4OR*, Springer, v. 5, n. 4, p. 269–298, 2007.
- BALINSKI, M. L.; QUANDT, R. E. On an integer program for a delivery problem. *Operations Research*, INFORMS, v. 12, n. 2, p. 300–304, 1964.
- BALLOU, R. H. *Gerenciamento da cadeia de suprimentos: logística empresarial*. [S.l.]: Bookman, 2006.
- BARNHART, C. et al. Branch-and-price: Column generation for solving huge integer programs. *Operations research*, INFORMS, v. 46, n. 3, p. 316–329, 1998.
- BENDERS, J. F. Partitioning procedures for solving mixed-variables programming problems. *Numerische mathematik*, Springer, v. 4, n. 1, p. 238–252, 1962.
- BOF, A. M.; SAMPAIO, C. E. M. *A educação no Brasil rural*. [S.l.]: INEP, 2006.
- BRACA, J. et al. A computerized approach to the New York Cityschool bus routing problem. *Iie Transactions*, Springer, v. 29, n. 8, p. 693–702, 1997.

- CARVALHO, W. L. et al. Rural school transportation in emerging countries: The Brazilian case. *Research in transportation economics*, Elsevier, v. 29, n. 1, p. 401–409, 2010.
- CARVALHO, W. L.; YAMASHITA, Y. Rural school transport in Brazil: an indispensable service for the education policy. *12th International Conference on Competition and Ownership in Land Passenger Transport.*, 2011.
- CHABRIER, A. Vehicle routing problem with elementary shortest path based column generation. *Computers & Operations Research*, Elsevier, v. 33, n. 10, p. 2972–2990, 2006.
- CHEN, D.-S. et al. A bus routing system for rural school districts. *Computers & Industrial Engineering*, Elsevier, v. 19, n. 1, p. 322–325, 1990.
- CHEN, Z.-L.; POWELL, W. B. A column generation based decomposition algorithm for a parallel machine just-in-time scheduling problem. *European Journal of Operational Research*, Elsevier, v. 116, n. 1, p. 220–232, 1999.
- CHRISTOFIDES, N.; EILON, S. An algorithm for the vehicle-dispatching problem. *Or*, JSTOR, p. 309–318, 1969.
- CHRISTOFIDES, N.; MINGOZZI, A.; TOTH, P. Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. *Mathematical programming*, Springer, v. 20, n. 1, p. 255–282, 1981.
- CHRISTOFIDES, N.; MINGOZZI, A.; TOTH, P. State-space relaxation procedures for the computation of bounds to routing problems. *Networks*, Wiley Online Library, v. 11, n. 2, p. 145–164, 1981.
- CHVÁTAL, V. Edmonds polytopes and weakly Hamiltonian graphs. *Mathematical Programming*, Springer, v. 5, n. 1, p. 29–40, 1973.
- CLARKE, G. u.; WRIGHT, J. W. Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, INFORMS, v. 12, n. 4, p. 568–581, 1964.
- CONTARDO, C.; CORDEAU, J.-F.; GENDRON, B. An exact algorithm based on cut-and-column generation for the capacitated location-routing problem. *INFORMS Journal on Computing*, INFORMS, v. 26, n. 1, p. 88–102, 2013.
- CONTARDO, C.; DESAULNIERS, G.; LESSARD, F. Reaching the elementary lower bound in the vehicle routing problem with time windows. *Networks*, Wiley Online Library, 2015.
- CORDEAU, J.-F. et al. A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational research society*, Palgrave Macmillan, v. 52, n. 8, p. 928–936, 2001.
- CORTÉS, C. E.; MATAMALA, M.; CONTARDO, C. The pickup and delivery problem with transfers: Formulation and a branch-and-cut solution method. *European Journal of Operational Research*, Elsevier, v. 200, n. 3, p. 711–724, 2010.
- DANTZIG, G.; FULKERSON, R.; JOHNSON, S. Solution of a large-scale traveling-salesman problem. *Journal of the operations research society of America*, INFORMS, v. 2, n. 4, p. 393–410, 1954.
- DANTZIG, G. B.; RAMSER, J. H. The truck dispatching problem. *Management science*, Inform, v. 6, n. 1, p. 80–91, 1959.
- DANTZIG, G. B.; WOLFE, P. Decomposition principle for linear programs. *Operations research*, v. 8, n. 1, p. 101–111, 1960.

- DEMIRIZ, A.; BENNETT, K. P.; SHAW-TAYLOR, J. Linear programming boosting via column generation. *Machine Learning*, Springer, v. 46, n. 1-3, p. 225–254, 2002.
- DESAULNIERS, G.; DESROSIERS, J.; SOLOMON, M. M. *Column generation*. [S.l.]: Springer, 2005.
- DESAULNIERS, G.; DESROSIERS, J.; SPOORENDONK, S. Cutting planes for branch-and-price algorithms. *Networks*, Wiley Online Library, v. 58, n. 4, p. 301–310, 2011.
- DESAULNIERS, G.; LESSARD, F.; HADJAR, A. Tabu search, partial elementarity, and generalized k-path inequalities for the vehicle routing problem with time windows. *Transportation Science*, INFORMS, v. 42, n. 3, p. 387–404, 2008.
- DESROCHERS, M. *La fabrication d'horaires de travail pour les conducteurs d'autobus par une méthode de génération de colonnes*. [S.l.]: Montréal: Université de Montréal, Centre de recherche sur les transports, 1986.
- DESROCHERS, M.; DESROSIERS, J.; SOLOMON, M. A new optimization algorithm for the vehicle routing problem with time windows. *Operations research*, INFORMS, v. 40, n. 2, p. 342–354, 1992.
- DESROCHERS, M.; LAPORTE, G. Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints. *Operations Research Letters*, Elsevier, v. 10, n. 1, p. 27–36, 1991.
- DESROCHERS, M. et al. Vehicle routing with time windows: Optimization and approximation. *Vehicle Routing: Methods and Studies*, Amsterdam, p. 65–84, 1988.
- DESROCHERS, M.; SOUMIS, F. A column generation approach to the urban transit crew scheduling problem. *Transportation Science*, INFORMS, v. 23, n. 1, p. 1–13, 1989.
- DESROSIERS, J. *An Overview of School Busing System*. [S.l.]: Montréal: Université de Montréal, Centre de recherche sur les transports, 1980.
- DESROSIERS, J.; LÜBBECKE, M. E. Branch-Price-and-Cut Algorithms. *Wiley encyclopedia of operations research and management science*, Wiley Online Library, 2011.
- DUMAS, Y.; DESROSIERS, J.; SOUMIS, F. The pickup and delivery problem with time windows. *European Journal of Operational Research*, Elsevier, v. 54, n. 1, p. 7–22, 1991.
- FEILLET, D. et al. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks*, Wiley Online Library, v. 44, n. 3, p. 216–229, 2004.
- FINKE, G.; CLAUS, A.; GUNN, E. A two-commodity network flow approach to the traveling salesman problem. *Congressus Numerantium*, v. 41, n. 1, p. 167–178, 1984.
- FISHER, M. L. Optimal solution of vehicle routing problems using minimum k-trees. *Operations research*, INFORMS, v. 42, n. 4, p. 626–642, 1994.
- FNDE. *Fundo Nacional de Desenvolvimento da Educação. Caminho da Escola - Histórico*. 2014. <<http://www.fnde.gov.br/programas/caminho-da-escola/caminho-da-escola-apresentacao/caminho-da-escola-historico>>. [Online; accessed 19-October-2014].
- FÜGENSCHUH, A. Solving a school bus scheduling problem with integer programming. *European Journal of Operational Research*, Elsevier, v. 193, n. 3, p. 867–884, 2009.
- FÜGENSCHUH, A. A set partitioning reformulation of a school bus scheduling problem. *Journal of Scheduling*, Springer, v. 14, n. 4, p. 307–318, 2011.

- FUKASAWA, R. et al. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Mathematical programming*, Springer, v. 106, n. 3, p. 491–511, 2006.
- GAMACHE, M. et al. A column generation approach for large-scale aircrew rostering problems. *Operations research*, INFORMS, v. 47, n. 2, p. 247–263, 1999.
- GAUVIN, C.; DESAULNIERS, G.; GENDREAU, M. A branch-cut-and-price algorithm for the vehicle routing problem with stochastic demands. *Computers & Operations Research*, Elsevier, v. 50, p. 141–153, 2014.
- GAVISH, B.; GRAVES, S. C. The travelling salesman problem and related problems. Massachusetts Institute of Technology, Operations Research Center, 1978.
- GENDREAU, M.; HERTZ, A.; LAPORTE, G. A tabu search heuristic for the vehicle routing problem. *Management science*, INFORMS, v. 40, n. 10, p. 1276–1290, 1994.
- GENDREAU, M.; POTVIN, J.-Y. *Dynamic vehicle routing and dispatching*. [S.l.]: Springer, 1998.
- GEOFFRION, A. M. *Lagrangian relaxation for integer programming*. [S.l.]: Springer, 1974.
- GLOVER, F. Tabu search-part I. *ORSA Journal on computing*, INFORMS, v. 1, n. 3, p. 190–206, 1989.
- GOMORY, R. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, v. 64, n. 5, p. 275–278, 1958.
- HADJICONSTANTINO, E.; CHRISTOFIDES, N.; MINGOZZI, A. A new exact algorithm for the vehicle routing problem based one-paths and k-shortest paths relaxations. *Annals of Operations Research*, Springer, v. 61, n. 1, p. 21–43, 1995.
- HOUCK, D. et al. *The traveling salesman problem as a constrained shortest path problem: Theory and computational experience*. [S.l.]: Ecole polytechnique de Montréal, 1978.
- Instituto Brasileiro de Geografia e Estatística. *IBGE | Comitê de Estatísticas Sociais*. 2014. <<http://ces.ibge.gov.br/base-de-dados/metadados/inep/educacao-basica>>. [Online; accessed 15-October-2014].
- IRNICH, S.; VILLENEUVE, D. The shortest-path problem with resource constraints and k-cycle elimination for k greater or equal 3. *INFORMS Journal on Computing*, INFORMS, v. 18, n. 3, p. 391–406, 2006.
- JEPSEN, M. et al. Subset-row inequalities applied to the vehicle-routing problem with time windows. *Operations Research*, INFORMS, v. 56, n. 2, p. 497–511, 2008.
- LAND, A. H.; DOIG, A. G. An automatic method of solving discrete programming problems. *Econometrica: Journal of the Econometric Society*, JSTOR, p. 497–520, 1960.
- LANGEVIN, A.; SOUMIS, F.; DESROSIERS, J. Classification of travelling salesman problem formulations. *Operations Research Letters*, Elsevier, v. 9, n. 2, p. 127–132, 1990.
- LAPORTE, G. The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, Elsevier, v. 59, n. 3, p. 345–358, 1992.
- LAPORTE, G.; NOBERT, Y.; DESROCHERS, M. Optimal routing under capacity and distance restrictions. *Operations research*, INFORMS, v. 33, n. 5, p. 1050–1073, 1985.

- LARSEN, J. *Parallelization of the vehicle routing problem with time windows*. Tese (Doutorado) — Technical University of Denmark Danmarks Tekniske Universitet, Department of Informatics and Mathematical Modeling Institut for Informatik og Matematisk Modellering, 1999.
- LAVOIE, S.; MINOUX, M.; ODIER, E. A new approach for crew pairing problems by column generation with an application to air transportation. *European Journal of Operational Research*, Elsevier, v. 35, n. 1, p. 45–58, 1988.
- LENSTRA, J. K.; KAN, A. Complexity of vehicle routing and scheduling problems. *Networks*, Wiley Online Library, v. 11, n. 2, p. 221–227, 1981.
- LETCHFORD, A. N.; SALAZAR-GONZÁLEZ, J.-J. Projection results for vehicle routing. *Mathematical Programming*, Springer, v. 105, n. 2-3, p. 251–274, 2006.
- LI, L.; FU, Z. The school bus routing problem: a case study. *Journal of the Operational Research Society*, JSTOR, p. 552–558, 2002.
- LIMA, M. P. Custos logísticos na economia brasileira. *Revista Tecnológica*, v. 11, n. 122, p. 64–69, 2006.
- LYSGAARD, J. Clarke & Wright’s Savings Algorithm. *Department of Management Science and Logistics, The Aarhus School of Business*, 1997.
- MANDUJANO, P.; GIESEN, R.; FERRER, J.-C. Model for optimization of locations of schools and student transportation in rural areas. *Transportation Research Record: Journal of the Transportation Research Board*, Trans Res Board, v. 2283, n. 1, p. 74–80, 2012.
- MILLER, C. E.; TUCKER, A. W.; ZEMLIN, R. A. Integer programming formulation of traveling salesman problems. *Journal of the ACM (JACM)*, ACM, v. 7, n. 4, p. 326–329, 1960.
- NEWTON, R. M.; THOMAS, W. H. Design of school bus routes by computer. *Socio-Economic Planning Sciences*, Elsevier, v. 3, n. 1, p. 75–85, 1969.
- PACHECO, J. et al. Bi-objective bus routing: an application to school buses in rural areas. *Transportation Science*, INFORMS, v. 47, n. 3, p. 397–411, 2013.
- PACHECO, J.; MARTÍ, R. Tabu search for a multi-objective routing problem. *Journal of the Operational Research Society*, Nature Publishing Group, v. 57, n. 1, p. 29–37, 2006.
- PARK, J.; KIM, B.-I. The school bus routing problem: A review. *European Journal of operational research*, Elsevier, v. 202, n. 2, p. 311–319, 2010.
- PESSOA, A.; UCHOA, E.; Poggi de Aragão, M. A robust branch-cut-and-price algorithm for the heterogeneous fleet vehicle routing problem. *Networks*, Wiley Online Library, v. 54, n. 4, p. 167–177, 2009.
- RIERA-LEDESMA, J.; SALAZAR-GONZÁLEZ, J.-J. Solving school bus routing using the multiple vehicle traveling purchaser problem: A branch-and-cut approach. *Computers & Operations Research*, Elsevier, v. 39, n. 2, p. 391–404, 2012.
- RIERA-LEDESMA, J.; SALAZAR-GONZÁLEZ, J. J. A column generation approach for a school bus routing problem with resource constraints. *Computers & Operations Research*, Elsevier, v. 40, n. 2, p. 566–583, 2013.
- RIGHINI, G.; SALANI, M. New dynamic programming algorithms for the resource constrained elementary shortest path problem. *Networks*, Wiley Online Library, v. 51, n. 3, p. 155–170, 2008.



- ROCHAT, Y.; TAILLARD, E. Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics*, v. 1, p. 147–167, 1995.
- ROPKE, S.; CORDEAU, J.-F. Branch and cut and price for the pickup and delivery problem with time windows. *Transportation Science*, INFORMS, v. 43, n. 3, p. 267–286, 2009.
- SAVELSBERGH, M. A branch-and-price algorithm for the generalized assignment problem. *Operations Research*, INFORMS, v. 45, n. 6, p. 831–841, 1997.
- SAVELSBERGH, M. W.; SOL, M. The general pickup and delivery problem. *Transportation science*, INFORMS, v. 29, n. 1, p. 17–29, 1995.
- SPADA, M.; BIERLAIRE, M.; LIEBLING, T. M. Decision-aiding methodology for the school bus routing and scheduling problem. *Transportation Science*, INFORMS, v. 39, n. 4, p. 477–490, 2005.
- SPOORENDONK, S. *Cut and column generation*. Tese (Doutorado) — PhD thesis, Technical University of Denmark, 2008.
- THANGIAH, S. R.; NYGARD, K. E. School bus routing using genetic algorithms. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. *Aerospace Sensing*. [S.l.], 1992. p. 387–398.
- TOTH, P.; VIGO, D. *The vehicle routing problem*. [S.l.]: Siam, 2001.
- Van den Akker, J.; HURKENS, C. A.; SAVELSBERGH, M. W. Time-indexed formulations for machine scheduling problems: Column generation. *INFORMS Journal on Computing*, INFORMS, v. 12, n. 2, p. 111–124, 2000.
- VASCONCELLOS, E. A. Transporte rural: o resgate de um tema esquecido. *Revista dos Transportes públicos*, v. 1, n. 75, p. 31–48, 1997.