UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS RUSSAS
CURSO DE GRADUAÇÃO EM ENGENHARIA DE SOFTWARE

DEYVISON NOGUEIRA RODRIGUES

PROBLEMA DE COMPARTILHAMENTO DE VEÍCULOS COM RESTRIÇÕES FÍSICAS E SOCIAIS: UMA ABORDAGEM DE PROGRAMAÇÃO DE INTEIRA

RUSSAS
2018

DEYVISON NOGUEIRA RODRIGUES

PROBLEMA DE COMPARTILHAMENTO DE VEÍCULOS COM RESTRIÇÕES FÍSICAS E SOCIAIS: UMA ABORDAGEM DE PROGRAMAÇÃO DE INTEIRA

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Software do Campus Russas da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Software.

Orientadora: Prof. Ms. Tatiane Fernandes Figueredo

RUSSAS

2018

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

R612p Rodrigues, Deyvison.
Problema de compartilhamento de veículos com restrições físicas e sociais : uma abordagem de programação de inteira / Deyvison Rodrigues. – 2018.
38 f.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Russas, Curso de Engenharia de Software, Russas, 2018.
Orientação: Profa. Ma. Tatiane Fernandes Figueiredo.

1. Compartilhamento de veículos. 2. Otimização combinatória. 3. Modelagem matemática. I. Título.

CDD 005.1

DEYVISON NOGUEIRA RODRIGUES

PROBLEMA DE COMPARTILHAMENTO DE VEÍCULOS COM RESTRIÇÕES FÍSICAS E SOCIAIS: UMA ABORDAGEM DE PROGRAMAÇÃO DE INTEIRA

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Software do Campus Russas da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Software.

Aprovada em:

BANCA EXAMINADORA

Prof. Ms. Tatiane Fernandes Figueredo (Orientadora)
Universidade Federal do Ceará (UFC)

Prof. Dr. Dmontier Pinheiro Aragão Júnior
Universidade Federal do Ceará (UFC)

Prof. Ms. Eurinardo Rodrigues Costa
Universidade Federal do Ceará (UFC)

AGRADECIMENTOS

Agradeço em primeiro lugar a Deus, pela a força e por ter iluminado o meu caminho durante esta jornada. Agradeço também a todos os professores que me acompanharam durante toda a graduação, em especial a Professora Tatiane Fernandes Figueiredo, que foi tão importante na minha vida acadêmica e no desenvolvimento deste trabalho. A toda minha família que, com muito carinho e apoio, não mediram esforços para que eu chegasse até esta etapa da minha vida. Ao grupo de pesquisa Núcleo de Estudo em aprendizado de Máquina e Otimização (NEMO) e aos meus amigos pelo apoio e incetivos constantes.

”Sonhos se tornam realidade quando se aproveita as oportunidades.”

(Russell Wilson)

RESUMO

A cada ano, estatísticas comprovam que o número de veículos particulares tem crescido de forma desenfreada, sendo um dos grandes responsáveis pela geração de problemas ambientais e de mobilidade. Dentre as possíveis soluções para essa realidade, destaca-se a ideia de que pessoas realizam itinerários semelhantes nas suas locomoções, conduzindo os veículos individualmente, quando poderiam realizar um compartilhamento. Essa solução requer certos cuidados, pois o compartilhamento com indivíduos desconhecidos pode gerar riscos à segurança dos envolvidos. A partir da análise da localização física e do uso das redes sociais dos indivíduos, este trabalho propõe uma abordagem de Modelagem Matemática para a resolução do Problema de Compartilhamento de Veículos com Restrições Físicas e Sociais. Utilizando conceitos de Teoria dos Grafos para a representação dos dados de entrada do problema, as técnicas aplicadas buscam por soluções que satisfaçam restrições semelhantes ao Problema de Caminho Mínimo com Visita Obrigatória de Vértices, procurando também maximizar as relações sociais entre os indivíduos envolvidos.

Palavras-chave: Compartilhamento de veículos. Otimização Combinatória. Modelagem Matemática.

ABSTRACT

Every year, statistical data show that the number of personal vehicles has grown a lot, being one of the responsible factors for generating environmental and mobility problems. Among possible solutions for this scenario is the idea that people perform similar itineraries in their tours, driving the vehicles individually instead of sharing them. This solution requires certain precautions, since the act of sharing among unknown individuals may create safety risks. From physical location analysis and use of social networks of individuals, this work proposes a Mathematical Programming approach to solve the Vehicle Sharing Problem with Physical and Social Constraints. Using Graph Theory concepts to represent input data of the problem, the applied techniques search for solutions that satisfy constraints related to the Shortest Path Problem Visiting Specified Vertices while trying to maximize social relations between individuals.

Keywords: Ridesharing. Combinatorial Optimization. Mathematical Modeling.

LISTA DE TABELAS

Tabela 1 – Grafos densos parte 1 32
Tabela 2 – Grafos densos parte 2 32
Tabela 3 – Grafos densos parte 3 33
Tabela 4 – Grafos densos parte 4 33
Tabela 5 – Grafos esparsos parte 1 34
Tabela 6 – Grafos esparsos parte 2 34
Tabela 7 – Grafos esparsos parte 3 35
Tabela 8 – Grafos esparsos parte 4 35

LISTA DE QUADROS

Quadro 1 – Modelo para instâncias 30
Quadro 2 – Variações de α e β 31

LISTA DE ABREVIATURAS E SIGLAS

B&C	Branch-and-Cut
DFJ	Dantzig–Fulkerson–Johnson
MTZ	Miller–Tucker–Zemlin
PCMVO	Problema de Caminho Mínimo com Visita Obrigatória de Vértices
PCV-MCa	Problema do Caixeiro Viajante com Múltiplas Caronas
PCVFS	Problema de Compartilhamento de Veículos com Restrições Físicas e Sociais
PLI	Programação Linear Inteira
PO	Pesquisa Operacional
RECs	Restrições de Eliminação de Ciclo

SUMÁRIO

1 INTRODUÇÃO 13

1.1 Objetivos 13

1.1.1 Objetivos Gerais 13

1.1.2 Objetivos Específicos 13

1.2 Metodologia 14

1.3 Organização do Trabalho 14

2 PROBLEMA DE COMPARTILHAMENTO DE VEÍCULO E *RIDESHARING*. 16

2.1 *Ridesharing* 16

2.2 O uso de redes sociais para resolução de problemas de segurança 16

3 REFERENCIAL TEÓRICO 18

3.1 Teoria dos Grafos 18

3.2 Pesquisa Operacional 19

3.2.1 Programação Linear 19

3.2.2 Programação Linear Inteira 21

3.2.3 Técnica de plano de corte 22

4 TRABALHOS RELACIONADOS 23

4.1 Problema do Caixeiro Viajante Com Múltiplas Caronas 23

4.2 Problemas de Roteamento com Coleta de Bônus 23

4.3 Problema de Caminho Mínimo com Visita Obrigatória de Vértices 24

5 O PROBLEMA DE COMPARTILHAMENTO DE VEÍCULOS COM RESTRIÇÕES FÍSICAS E SOCIAIS 25

5.1 Definição do problema 25

5.2 Formulação Matemática 26

5.2.1 Dantzig–Fulkerson–Johnson (DFJ) 28

5.2.2 Miller–Tucker–Zemlin (MTZ) 28

6 RESULTADOS 30

6.1 Configuração do ambiente computacional 30

6.2 Instâncias 30

6.2.1 Características das instâncias 30

6.2.2 *Geração de instâncias aleatórias* 30
6.3 **Estudo comparativo entre as formulações** 31
7 **CONCLUSÕES E TRABALHOS FUTUROS** 36
7.1 **Conclusões** 36
7.2 **Trabalhos Futuros** 36
REFERÊNCIAS 37

1 INTRODUÇÃO

Segundo dados do DENATRAN - Departamento de Trânsito de Veículos (2016), o Brasil possua circulando pelas ruas e estradas do país, aproximadamente, cerca de 50 milhões de veículos, responsáveis pela geração de diversos problemas ambientais e de mobilidade (COSTA, 2003). Estes problemas estão relacionados a diversos fatores, como por exemplo, falta de educação no trânsito, sistema de transporte público ineficiente, uso de um grande número de veículos com motores poluentes, dentre outros (SCARINGELLA, 2001). Uma solução ideal necessita de uma intervenção profunda e a longo prazo. Portanto, é comum entre a população, a busca de formas simples para minimizar o trânsito caótico e gastos financeiros. Dentre elas, destaca-se a ideia de que pessoas que realizam itinerários semelhantes nas suas locomoções, conduzindo os veículos individualmente, poderiam realizar um compartilhamento. Essa solução requer certos cuidados, pois o compartilhamento com indivíduos desconhecidos pode gerar riscos à segurança dos envolvidos.

A partir da análise da localização física e do uso de redes sociais dos indivíduos, este trabalho busca apresentar uma solução eficiente e segura para este tipo de compartilhamento. Para tal, é definido matematicamente um novo problema computacional denominado Problema de Compartilhamento de Veículos com Restrições Físicas e Sociais (PCVFS). Utilizando conceitos de Teoria dos Grafos para a representação dos dados de entrada do problema, são apresentadas formulações matemáticas que buscam por soluções que satisfaçam restrições semelhantes ao Problema de Caminho Mínimo com Visita Obrigatória de Vértices (ANDRADE, 2016), procurando também maximizar as relações sociais entre os indivíduos envolvidos.

1.1 Objetivos

1.1.1 Objetivos Gerais

Apresentar soluções para o Problema de Compartilhamento de Veículos com Restrições Físicas e Sociais através de técnicas de Programação Linear Inteira.

1.1.2 Objetivos Específicos

- Realizar um estudo da área de Teoria dos Grafos e Pesquisa Operacional;
- Definir matematicamente o Problema de Compartilhamento de Veículos com Restrições

Físicas e Sociais;

- Apresentar formulações matemáticas para o problema proposto;
- Estudo de técnicas de Programação Linear Inteira para melhorar as formulações propostas;
- Criar um gerador de instâncias para o problema, baseado em problemas semelhantes existentes na literatura;
- Efetuar uma análise estatística do desempenho das formulações matemáticas e melhorias propostas apresentadas.

1.2 Metodologia

Dada a natureza aplicada da pesquisa realizada, o trabalho será desenvolvido com base em uma adaptação das fases abordadas por Hillier e Lieberman (2013) para a elaboração de um estudo de pesquisa operacional:

- Definição do problema de interesse;
- Formulação de modelos matemáticos para representar o problema;
- Desenvolvimento de um procedimento computacional a fim de derivar soluções para o problema a partir dos modelos;
- Teste dos modelos e aprimoramento conforme necessário.

Em seguida, os algoritmos exatos propostos serão alimentados com as instâncias obtidas pelo gerador, criado a partir de estudos de problemas semelhantes existentes na literatura, a fim de comparar os seus desempenhos. O estudo será explicativo e seguirá uma abordagem quantitativa, utilizando-se do método de pesquisa de modelagem e simulação. Neste método, algoritmos exatos serão utilizados para auxílio na tomada de decisões e resolução do Problema de Compartilhamento de Veículos com Restrições Físicas e Sociais.

1.3 Organização do Trabalho

O restante do trabalho se divide em 6 capítulos. No Capítulo 2 é apresentado uma contextualização do Problema de Compartilhamento de Veículos na sociedade atual. No Capítulo 3 são apresentados os conceitos base necessários para resolução do problema proposto. O Capítulo 4 apresenta os trabalhos relacionados ao tema da monografia, enquanto o Capítulo 5 define matematicamente o Problema de Compartilhamento de Veículos com Restrições Físicas e

Sociais. O Capitulo 6 apresenta os resultados obtidos no trabalho. Por fim, o Capítulo 7 apresenta as conclusões e os trabalhos futuros.

2 PROBLEMA DE COMPARTILHAMENTO DE VEÍCULO E *RIDESHARING*.

Este capítulo descreve mais detalhes do Problema de Compartilhamento de Veículos e *Ridesharing*. Na Seção 2.1 é apresentado uma contextualização do Ridesharing e na Seção 2.2 é descrito como as redes sociais são utilizadas para diminuir dificuldades no *Ridesharing*

2.1 *Ridesharing*

O compartilhamento solidário de veículo, também conhecido como *Ridesharing*, pode ser definido como um problema dinâmico, onde as pessoas com itinerários semelhantes compartilham veículos, sem histórico de cooperação anterior, diferentemente do *Carpooling*, onde indivíduos percorrem o mesmo trajeto rotineiramente e uma alocação fixa de passageiros e carros pode ser estabelecida (DAILEY *et al.*, 1999). O *Ridesharing* se diferencia também do transporte público, pois neste, o compartilhamento é feito em trajetos definidos, com horários e capacidade preestabelecidos tendo uma finalidade específica.

Devido a alta taxa de congestionamentos nas grandes cidades e um grande número de veículos particulares com assentos ociosos, o estudo do problema de compartilhamento de veículos têm uma grande importância científica e social, pois o uso desta forma de transporte tem potencial para reduzir custos, impactos ambientais e otimizar a circulação de veículos (ARAÚJO, 2016).

Apesar das diversas vantagens do uso do compartilhamento de veículos, ele não é largamente utilizado, principalmente, devido à problemas de segurança, causados pela falta de informações pessoais sobre os indivíduos participantes do compartilhamento. Uma proposta para a diminuição deste problema é o auxílio das redes socais para selecionar os indivíduos que compartilharão esse serviço.

2.2 O uso de redes sociais para resolução de problemas de segurança

Diversas propostas para solucionar o problema de compartilhamento de veículos de forma segura foram apresentados na literatura ou disponibilizados para sociedade através de aplicativos, como exemplo, o *BlaBlarCar*. Criado na França em 2006, o serviço está presente em 20 países e conta com 20 milhões de usuários (DEBATIN; ANDRADE, 1998 apud CAPELAS, 2015). Outro aplicativo utilizado no Brasil com funcionalidade semelhante é o *Caronafacil* que reúne mais de 50 mil usuários. A ideia de ambos aplicativos para redução de problemas de segurança é organizar grupos de carona na plataforma social *Facebook*. Porém, as soluções apresentadas em ambos aplicativos, apenas indicam as possibilidades cadastradas no aplicativo, para que os próprios envolvidos busquem os interessados, não sendo apresentadas soluções que considerem a distância física dos envolvidos, sendo a única garantia de segurança a obrigatoriedade de pré-cadastros em redes sociais. Em Furtado e Luiz (2013) é apresentada uma solução que considera as distâncias entre os indivíduos envolvidos baseada no algoritmo das p-medianas, onde cada indivíduo recebe somente a informação se os demais envolvidos são ou não seu amigo na rede social *Faceboook*. A ideia de que a solução apresente as melhores opções considerando a distância física e social entre os envolvidos que não são amigos na rede social *Faceboook* não é implementado.

3 REFERENCIAL TEÓRICO

Para definição matemática do problema abordado neste trabalho é necessário alguns conhecimentos iniciais na área de Teoria dos Grafos e Pesquisa Operacional apresentados nas subseções 3.1 e 3.2, respectivamente.

3.1 Teoria dos Grafos

Um *grafo* $G = (V(G), E(G))$ consiste de um conjunto finito não vazio $V(G)$ de elementos, chamados *vértices*, e um conjunto de pares não-ordenados de elementos distintos de $V(G)$, chamado *arestas*, isto é, $E(G) \subseteq \{\{u,v\}|u,v \in V(G), u \neq v\}$. Quando o grafo estiver claro pelo contexto, é utilizado V e E para $V(G)$ e $E(G)$ respectivamente, para simplificar a notação. Se $e = \{u,v\}$ é uma aresta, dizemos que *e incide em u e em v*; que *u e v são seus extremos*; e que *u e v são adjacentes*. Para simplificar a notação, é denotado uma aresta $e = \{u,v\}$ por uv.

A *vizinhança* ou *adjacência* de um vértice v de G é o conjunto contendo todos os vértices adjacentes a v. A vizinhança é frequentemente denotada por $N_G(v)$ ou simplesmente $N(v)$ quando o grafo está claro pelo contexto. Seja $V' \subseteq V$ a vizinhança de um subconjunto de vértices $V' \subseteq V$ é denotada por $N(V') = \bigcup_{v \in V'} N(v)$. denotamos por $\delta(V')$ o conjunto de arestas em $E(G)$ com um extremo em V' e o outro em $V \setminus V'$. Se G e G' são dois grafos tais que $V(G') \subseteq V(G)$ e $E(G') \subseteq E(G)$, então G' é dito um *subgrafo* de G e escrevemos $G' \subseteq G$.

Um *passeio* é definido como uma sequência de vértices $v \in V(G)$ dotada da seguinte propriedade: se u e v são vértices consecutivos na sequência então uv é um arco do grafo G. Um arco do passeio é qualquer arco $uv \in E(G)$ tal que v é o sucessor de u no passeio. Um *caminho* em um grafo G é um passeio com uma sequência de vértices $(v_0, v_1, ...v_k)$ de $V(G)$ distintos entre si, ou seja, um passeio em que os arcos são todos diferentes entre si.

Um *grafo orientado (digrafo)* $D = (V,E)$ consiste de um conjunto finito não-vazio $V(D)$ de elementos, chamados *vértices* ou *nós*, e um conjunto $E(D)$ de elementos, chamados *arcos* ou *arestas orientadas*, que são pares ordenados de vértices distintos, isto é, $E(D) \subseteq \{(u,v)|u,v \in V(D), u \neq v\}$. Se $a = (u,v)$ é um arco, dizemos que a incide em u e em v; que u e v são seus extremos. Também nos é referido a u como *início* ou *origem* e a v como *término* ou *destino* do arco a.

Dado um vértice $v \in V(D)$ é denotado por $\delta^+(v)$ o conjunto de arcos que possuem o vértice v como início e $\delta^-(v)$ o conjunto de arcos que possuem o vértice v como término. Se

$\delta^-(v) = \emptyset$, v é dito uma *fonte*; Se $\delta^+(v) = \emptyset$, v é dito um *sumidouro*. Se V' é um subconjunto de vértices em $V(D)$, denotamos por $\delta^+(V')$ o conjunto dos arcos de D com início em V' e término em $V \setminus V'$. Um conjunto da forma $\delta^+(V')$, onde $\emptyset \neq V' \subsetneq V$, é dito um *corte* em D. Se v' é um vértice em V' e v um vértice em $V \setminus V'$, então dizemos que $\delta^+(V')$ é um (v', v)-*corte*. A notação $[V', V \setminus V']$ é utilizada para representar um corte $\delta^+(V')$.

Todos os conceitos definidos para grafos que não envolvem a noção de orientação se aplicam analogamente para os grafos orientados. Embora seja utilizado preferencialmente a notação $D = (V, A)$ para um grafo orientado, vamos usar também $G = (V, E)$, quando não houver prejuízo para o entendimento, e neste caso E será um conjunto de arcos.

3.2 Pesquisa Operacional

A Pesquisa Operacional (PO) é um ramo interdisciplinar da matemática aplicada que faz uso de modelos matemáticos, estatísticos e de algoritmos na ajuda à tomada de decisão. É usada sobretudo para analisar sistemas complexos do mundo real, tipicamente com o objetivo de melhorar ou otimizar a performance. Os problemas determinísticos de PO podem ser classificados em algumas categorias genéricas: Problemas de Programação Linear, Modelos de Otimização em Redes, Simulação, Teoria dos Jogos, Programação Dinâmica e Programação Não-Linear. O foco deste trabalho reside na aplicação de técnicas de Programação Linear Inteira para resolução do PCVFS, solucionando-o através de uma variante do Problema de Caminho Mínimo denominada Problema de Caminho Mínimo com Visita Obrigatória de Vértices (PCMVO) (ANDRADE, 2016).

3.2.1 Programação Linear

Para Luna e Goldbarg (2000) a Programação Linear apresenta algoritmos eficientes e que podem ser facilmente resolvidos utilizando o computador. A Programação Linear é uma técnica que utiliza instrumentos matemáticos que permitem a otimização de operações, e é largamente utilizada na resolução de problemas que tenham seus modelos representados por expressões lineares.

> Os problemas de Programação Linear referem-se à distribuição eficiente de recursos limitados entre atividades competitivas, com a finalidade de atender a um determinado objetivo, por exemplo, maximização de lucros ou minimização de custos. Em se tratando de programação linear, esse objetivo será expresso por uma função linear, à qual se dá o nome de função objetiva. É claro que é

> necessário dizer quais as atividades que consomem cada recurso, e em que proporção é feito esse consumo. Essas informações serão fornecidas por equação ou inequações lineares, uma para cada recurso. Ao conjunto dessas equações ou inequações lineares dá-se o nome de restrição do modelo. Geralmente existem inúmeras maneiras de distribuir os escassos recursos entre as diversas atividades, bastando para isso que essas distribuições sejam coerentes com as equações de consumo de cada recurso, ou seja, que elas satisfaçam as restrições do problema. Entretanto, deseja-se achar aquela distribuição que satisfaça as restrições do problema, e que alcance o objetivo desejado, isto é, que maximize o lucro ou minimize o custo. A essa solução dá-se o nome de solução ótima. Uma vez obtido o modelo linear, constituído pela função objetiva (linear) e pelas restrições lineares, a programação linear se incumbe de achar a sua solução ótima (PUCCINI, 1980).

Segundo Caixeta-Filho (2001), algebricamente, a Programação Linear é o aprimoramento de uma técnica de resolução de sistema de equações lineares, utilizando inversões sucessivas de matrizes, incorporando uma equação linear adicional representativa de um dado comportamento que deverá ser otimizado. Para Garcia *et al.* (1997), matematicamente, pode-se formular o modelo de um problema de otimização de acordo com o seguinte esquema:

Maximizar ou minimizar:

$$Z = C_1X_1 + C_2X_2 + ... + C_nX_n$$

Sujeito a

$$a_{11}x_1 + a_{12}x_2 + ... + a_{1n}X_n < b_1$$
$$a_{21}x_1 + a_{22}x_2 + ... + a_{2n}X_n < b_2$$
$$...$$
$$a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}X_n < b_m$$
$$x_i \geq 0 \text{ e } b_j \geq 0, \forall i = 1,2...n \text{ e } j = 1,2...m$$

Onde:

Z = função a ser maximizada ou minimizada (geralmente ganho ou custo), respeitando o conjunto de elementos do problema ou restrições;

x_i = variáveis que representam as quantidades ou recursos que se quer determinar para otimizar o resultado global;

C_i = coeficientes de ganho ou custo que cada variável é capaz de gerar;

b_j = quantidade disponível de cada recurso ;

a_{ij} = quantidade de recurso em que cada variável decisória consome.

Problemas simples ou extremamente elaborados de Programação Linear podem ser resolvidos utilizando o método simplex, algoritmo introduzido por Dantzig (1953).

3.2.2 *Programação Linear Inteira*

Um problema de Programação Linear Inteira (PLI) se caracteriza pelo fato de conter todas as variáveis com valores discretos, ou seja, inteiras. No caso em que apenas algumas das variáveis são inteiras, o problema é chamado de Programa Linear Inteira Mista. Outro caso particular é quando as variáveis são inteiras e restritas aos valores 0 ou 1, esse caso é conhecido como Programa Linear Inteiro Binário (WOLSEY, 1998).

Mesmo sendo um grande foco de estudo em Pesquisa Operacional, ainda não foi desenvolvido um algoritmo eficiente para resolver um PLI de tamanho razoável, ou seja, não existe um algoritmo que resolva um PLI com um número de passos polinomial no tamanho da entrada. De fato, resolver um PLI é um problema da classe NP-Completo. Acredita-se que não exista solução eficiente para um problema dessa classe e qualquer solução correta do problema necessita de tempo exponencial no pior caso. Contudo, existem dois métodos aceitáveis, em termos computacionais, que resolvem problemas grandes. São eles: algoritmos de Planos de Corte e algoritmos Enumerativos (PAPADIMITRIOU; STEIGLITZ, 1998).

Um importante conceito utilizado por ambas as abordagens é o de relaxação linear. Uma relaxação linear de um PLI é a remoção da restrição que diz que as variáveis devem ser inteiras, ou seja, as variáveis discretas são transformadas em variáveis contínuas. A relaxação é uma importante ferramenta para a obtenção de limitantes para os problemas de programação linear inteira. Os algoritmos de Plano de Corte seguem uma ideia básica, em que primeiro se resolve a relaxação linear do problema utilizando o método Simplex. Em seguida, é verificado se a solução é inteira e, em caso afirmativo, a solução da relaxação é também a solução do problema original; se não for inteira, uma nova restrição é adicionada, que deve ser satisfeita por todas as soluções viáveis inteiras, mas não pela solução da relaxação. Depois de adicionada a restrição, os passos anteriores – relaxação, checagem e adição da restrição – são executados novamente até que se encontre uma solução inteira (BERTSIMAS; TSITSIKLIS, 1997). Já os algoritmos enumerativos consistem em enumerar as soluções viáveis de forma inteligente. Em outras palavras, o que um algoritmo enumerativo tenta fazer é provar de forma construtiva que uma solução é ótima, baseando-se em um número de partições do problema. Resolver a relaxação linear, como no algoritmo de Plano de Corte, é o primeiro passo de um algoritmo enumerativo. Também é feita a checagem da solução da relaxação, entretanto o próximo passo não é adicionar novas restrições e sim dividir o problema em dois subproblemas ao adicionar restrições que são mutuamente exclusivas e exaustivas. Em seguida é escolhido um dos subproblemas para ser relaxado e resolvido para passar pelo processo novamente (PAPADIMITRIOU; STEIGLITZ, 1998).

3.2.3 Técnica de plano de corte

Quando uma formulação possui um número exponencial de restrições, separar tais restrições da formulação pode ser uma alternativa interessante. Esta técnica, amplamente utilizada em Programação Linear Inteira, pode ser definida através das seguintes etapas: Inicialmente, são removidas as restrições de número exponencial da formulação original, gerando uma nova formulação. A partir da resolução da relaxação linear (remoção das restrições de integralidade da formulação) obtêm-se uma solução que pode, ou não, ser uma solução para a formulação original. Para garantir que a solução da nova formulação relaxada é solução para formulação original, verifica-se através de um algoritmo polinomial, se essa viola alguma das restrições excluídas ou satisfaz todas elas. No primeiro caso, incluímos na formulação as restrições violadas identificadas e é repetido o processo. No segundo, é obtido a solução ótima. A etapa para identificar restrições violadas chama-se *separação* e o processo como um todo define o *método de planos de corte* (GILMORE; GOMORY, 1961).

4 TRABALHOS RELACIONADOS

Este capítulo descreve alguns trabalhos da literatura mais relevantes para a contextualização do problema proposto nesta monografia. Nas Seções 4.1, 4.2 e 4.3 são apresentados problemas semelhantes encontrado na literatura.

4.1 Problema do Caixeiro Viajante Com Múltiplas Caronas

O Problema do Caixeiro Viajante com Múltiplas Caronas (PCV-MCa) é um problema que mescla a classe dos problemas do Caixeiro Viajante com problemas de *Ridesharing* (ARAÚJO, 2016). Considerando que o Caixeiro Viajante utiliza um veículo com determinada capacidade, e que pode aproveitar os assentos ociosos para oferecer carona a passageiros que queiram se deslocar para trechos ainda não visitados pelo Caixeiro Viajante, compartilhando despesas de viagem nestes trechos. O Caixeiro Viajante e os passageiros podem, dessa maneira, reduzir suas despesas de deslocamento. O PCV-MCa a encaixa-se no padrão 4 na classificação proposta por (FURUHATA *et al.*, 2013), uma vez que o trajeto do motorista é adaptado em função dos trajetos dos passageiros, pois o custo mínimo é alcançado através da partilha dos custos entre todos os participantes. Este Problema se diferencia do PCVFS pois, não considera questões sociais para escolher os indivíduos que irão receber caronas, podendo trazer problemas de segurança.

4.2 Problemas de Roteamento com Coleta de Bônus

Os problemas de roteamento com coleta de bônus, são definidos em grafos com bônus associados aos vértices. O motorista não é obrigado a visitar qualquer vértice específico do grafo, contudo deve realizar um ciclo Hamiltoniano que permita recolher pelo menos uma quota conhecida de bônus (AWERBUCH *et al.*, 1998). Essa classe de problemas foi introduzida por Balas (1989). Awerbuch *et al.* (1998) estuda um caso de coleta semelhante denominado como *Quota Traveling Salesman Problem*. No tema, destacam-se os trabalhos de Ausiello *et al.* (2004) que relata um algoritmo para o *The On-Line Quota Traveling Salesman Problem*, bem como no trabalho de Yu *et al.* (2014), qual investiga algoritmos determinísticos ótimos para algumas variantes do Problema do Caixeiro Viajante com Quotas *On-Line*. Em Jozefowiez *et al.* (2008) é introduzido uma nova abordagem para o Problema do Roteamento com Coleta de Bônus multiobjetivo que lida com dois objetivos conflitantes: minimizar o tamanho do *tour* e maximizar a coleta de bônus. Note que os problemas de roteamento com coleta de bônus visam somente diminuir o custo e considerando sempre o indivíduo que tem uma maior contribuição na redução deste custo, sem a obrigatoriedade da dar caronas e sem considerar questões sociais como o PCVFS.

4.3 Problema de Caminho Mínimo com Visita Obrigatória de Vértices

Segundo Andrade (2016), dado um grafo direcionado $D = (V, E)$ com um conjunto de vértices V, um conjunto ponderados de arestas E e um conjunto de vértices P tal que $P \subseteq V - \{s,t\}$, o Problema de Caminho Mínimo com Visita Obrigatórias de Vértices (PCMVO) consiste em encontrar o menor $(s - P - t)$-caminho entre o vértice de origem $s \in V$ e o vértice de destino $t \in V$ visitando somente uma vez todos os vértices de um conjunto P. Quando é definido $P = V - \{s,t\}$ o problema é igual a encontrar um caminho *Hamiltoniano* de menor custo, o que é NP-Difícil. O PCVFS pode ser visto como uma aplicação PCMVO quando não se considera os requisitos de relações sociais. Neste caso, cada vértice de P pode ser visto como um indivíduo que deseja participar do compartilhamento, onde seria necessário acrescentar a restrição para garantir que apenas um número $w \in \mathbb{Z}$ de vértices de P fossem obrigatoriamente visitados. Note ainda que o PCVFS, com as restrições de relações sociais, pode ser visto como um generalização do PCMVO.

5 O PROBLEMA DE COMPARTILHAMENTO DE VEÍCULOS COM RESTRIÇÕES FÍSICAS E SOCIAIS

Neste capítulo, é apresentado o Problema de Compartilhamento de Veículos com Restrições Físicas e Sociais (PCVFS), que é foco de estudo deste trabalho. A definição formal do problema aparece na Seção 5.1. Na Seção 5.2 é apresentado duas formulações matemáticas para o PCVFS.

5.1 Definição do problema

O Problema de Compartilhamento de Veículos com Restrições Físicas e Sociais (PCVFS) pode ser visto como um Problema de Caminho Mínimo com Visita Obrigatória de Vértices PCMVO procurando também maximizar as relações sociais entre os indivíduos envolvidos, apresentados na Seção 2.2. Para descrevê-lo, será utilizado um dígrafo que descreve o mapa utilizado para locomoção dos indivíduos participantes do compartilhamento, ou seja, dado um dígrafo $D = (V,E)$ ponderado nas arestas, associamos um subconjunto de vértice $V' \subseteq V(G)$ uma função $f(v) \subseteq S$. Então é desejado formar um conjunto de indivíduos mais harmoniosos possíveis, considerando os graus de relacionamentos, e que respeitem as restrições de caminho mínimo.

Problema 5.1.1 *O Problema de Compartilhamento de Veículos com Restrições Físicas e Sociais (PCVFS)*

Uma tupla $(D, V', s, t, w, f, d, \alpha, \beta)$, onde:

- *$D = (V,A)$ é um digrafo, onde $V(D)$ são as localizações do mapa e $A(D)$ são arestas rotuladas com a distância entre u e v;*
- *V' é um subconjunto de $V(D)$, tal que $V' \subset V(D)$ e indica os indivíduos v' que podem receber carona;*
- *s é um vértice, tal que $s \in V(D)$, representando a localização inicial de quem dá carona;*
- *t é um vértice, tal que $t \in V(D)$, representando o destino dos indivíduos;*
- *w é uma constante, tal que $w \in \mathbb{N}$ representa o número de pessoas que receberão carona;*
- *Função $f : V' \to \mathbb{Z}_+$ associadas a V', onde quanto menor o valor de $f(u)$, maior é o grau de afinidade entre os vértices s e u;*
- *Função $d : A \to \mathbb{R}_+$ associadas a $A(D)$, que denota o peso do arco $uv \in A(D)$ representando a distância entre os vértices u e v;*
- *α e β são multiplicadores da função objetivo tal que $\{\alpha \in \mathbb{R} : 0 \leq \alpha \leq 1\}$, $\{\beta \in \mathbb{R} : 0 \leq \beta \leq 1\}$ e $\alpha + \beta = 1$ definindo a prioridade de cada medida.*

***Questão**: Determinar $s-t$ caminho, onde:*

1. *Exatamente w vértices pertencentes a V' estejam no caminho $s-t$;*
2. *O somatório dos pesos $f(v) \in V'$ e $d(uv)$ pertencentes ao caminho $s-t$ sejam minimizados.*

Ou a verificação de que esse caminho não pode ser formado.

5.2 Formulação Matemática

Dada uma instância $I_{PCVFS} = (G, V', w, f, d, \alpha, \beta)$ para o problema $PCVFS$, como definida na Seção 5.1, considere as seguintes variáveis.

$$X_{ij} = \begin{cases} 1, & \text{se a aresta } ij \in (s-V-t) \text{ caminho} \\ 0, & \text{caso contrário} \end{cases} \quad ,\forall ij \in A(G). \tag{5.1}$$

$$Y_i = \begin{cases} 1, & \text{se } i \text{ é vértice escolhido} \\ 0, & \text{caso contrário} \end{cases} \quad ,\forall i \in V'(G). \tag{5.2}$$

$$\min \quad = \alpha \sum_{ij \in A(G)} d_{ij} X_{ij} + \beta \sum_{i \in V'} f_i Y_i \tag{5.3}$$

$$\text{s.a:} \quad \sum_{\substack{j \in V(G): \\ sj \in A(G)}} X_{sj} = 1 \qquad \forall j \in V(G) \tag{5.4}$$

$$\sum_{\substack{i \in V(G): \\ it \in A(G)}} X_{it} = 1 \qquad \forall i \in V(G) \tag{5.5}$$

$$\sum_{\substack{i \in V(G): \\ ij \in A(G)}} X_{ij} - \sum_{\substack{i \in V(G): \\ ij \in A(G)}} X_{ji} = 0 \qquad \forall j \in V(G)/\{s,t\} \tag{5.6}$$

$$\sum_{\substack{i \in V(G): \\ ij \in A(G)}} X_{ij} = Y_j \qquad \forall j \in V' \tag{5.7}$$

$$\sum_{j \in V'} Y_j = w \qquad \forall j \in V' \tag{5.8}$$

$$\text{Restrições de Eliminação de Ciclo (RECs)} \tag{5.9}$$

$$X_{ij} \in \{0,1\}, \quad \forall (i,j) \in A \tag{5.10}$$

$$Y_i \in \{0,1\}, \quad \forall i \in V' \tag{5.11}$$

A Função objetivo 5.3 busca minimizar a distância do caminho e a não afinidade entre os indivíduos, note que as funções d e f são multiplicadas por α e β respectivamente. Supondo o $\alpha = 1$ e $\beta = 0$ é reduzido ao PCMVO, visitando w vértices de V' ignorando a afinidade entre s e os elementos de V'. Por outro lado, supondo o $\alpha = 0$ e $\beta = 1$ é conduzido a visitar w vértices com maior afinidade com s, desconsiderando a distância entre eles.

As Restrições (5.4) e (5.5) definem que haverá apenas uma saída do vértice s e uma chegada no vértice t, respectivamente. A restrição (5.6) (Conservação de fluxo) juntamente com as restrições (5.4) e (5.5) constroem um e apenas um caminho. Por fim (5.8) e (5.7) definem o número de indivíduos (vértices) que receberão carona e obriga que o caminho contenha esse vértice.

Os diferentes modelos se diferenciam como é proposto a Restrição de Eliminações de Ciclos (RECs) 5.9 para o PCVFS. A diferentes abordagens das RECs são a restrição Dantzig–Fulkerson–Johnson (DFJ) (DANTZIG *et al.*, 1954) e a restrição Miller–Tucker–Zemlin (MTZ) (MILLER *et al.*, 1960) que serão demonstradas nas subseções 5.2.1 e 5.2.2 respectivamente.

5.2.1 Dantzig–Fulkerson–Johnson (DFJ)

Uma maneira simples de quebrar ciclos para o PCVFS consiste em obrigar o número de arestas em qualquer subconjunto de vértices $S \subseteq V$ ser menor ou igual a $|S| - 1$. Isto realizado considerando a desigualdade de DFJ (DANTZIG *et al.*, 1954)

$$\sum_{ij \in A(S)} X_{ij} \leq |S| - 1 \qquad ,\forall \quad S \subset V : |S| \geq 2 \tag{5.12}$$

A principal desvantagem da restrição 5.12 é o número exponencial de desigualdades. Para contornar este problema, aplicaremos a técnica de plano de corte. Criaremos um *Branch-and-Cut (B&C)* para resolver as instâncias do PCVFS quando é implementado a restrição DFJ. Para tal, inicialmente omitimos a restrição 5.12 da formulação, sempre que for gerada uma solução em cada nó criado pelo *branch* na árvore de soluções, verifica-se, em tempo polinomial, a existência de ciclos. Se a solução obtida contém ciclos é adicionada a restrição 5.12 associada ao(s) ciclo(s) encontrado(s), caso não seja encontrado nenhum ciclo na solução recorrente obtém-se uma solução viável. No final de processo, a solução final gerada é solução ótima para a instância em execução.

5.2.2 Miller–Tucker–Zemlin (MTZ)

A restrição MTZ foi inicialmente utilizada para a quebra de ciclos (MILLER *et al.*, 1960) em formulações para Problema do Caxeiro Viajante. Resumidamente, a ideia principal desta restrição é definir rótulos para os vértices de um grafo de modo que estes rótulos representem a ordem que os vértices foram visitados pela rota do caixeiro viajante.

A Restrição MTZ foi adaptada para utilização na formulação para PCVFS da seguinte forma: inicialmente, definimos π_i variáveis contínuas que representam os rótulos mencionados acima, $i = \{1, 2, ..., |V|\}$; definimos também uma constante $\mathscr{M} \in \mathbb{Z}$, com valor suficientemente grande, responsável por garantir que a ordem dos valores π_i restrinjam a existência de ciclos na formulação. O conjunto de restrições MTZ utilizadas na formulação do PCVFS são apresentadas a seguir:

$$\pi_j - \pi_i \geq 2 - x_{ij} - \mathscr{M}(1 - x_{ij}), \quad \forall (i,j) \in A \tag{5.13}$$

$$\pi_s = 0 \tag{5.14}$$

$$\pi_i \geq 0, \quad \forall i \in V - \{s\} \tag{5.15}$$

As restrições inserem rótulos somente a variáveis x quando existe um arco $(i,j) \in E(G)$, somente se for parte da solução.

6 RESULTADOS

Este capítulo apresenta os resultados da implementação dos algoritmos exatos das formulações propostas. Inicialmente na Seção 6.1 a Configuração do ambiente computacional. Na Seção 6.2 é demonstrado como as instâncias foram criadas e suas características. Por fim, na Seção 6.3 é apresentado uma comparação entre as formulações propostas.

6.1 Configuração do ambiente computacional

Para Implementação das formulações dos modelos propostos na seção 5.2 foi utilizado o pacote de otimização ILOG CPLEX®12.7 implementado na Linguagem JAVA. Os experimentos realizados foram executados utilizando uma máquina Intel Pentium i7, 8 x 3.60 GHz, 16 GB DDR3 RAM e sistema operacional Linux Ubuntu 14.05. O tempo de execução limite para cada instância de 3600 segundos.

6.2 Instâncias

6.2.1 Características das instâncias

As instâncias foram geradas de forma aleatória, separado em seis grupos divididos, onde cada grupo diferentes dimensões de tamanho (n) e em dois subgrupos, densos e esparsos. As instâncias estão disponíveis através do link <https://github.com/deyvisonnogueira/instancias>.

Quadro 1 – Modelo para instâncias

	Densos	Esparsos	n
Grupo 1	10	10	100
Grupo 2	10	10	120
Grupo 3	10	10	140
Grupo 4	10	10	160
Grupo 5	10	10	180
Grupo 6	10	10	200

Fonte: Elaborado pelo Autor (2018).

6.2.2 Geração de instâncias aleatórias

Devido a escassez de tempo não foi possível gerar instâncias reais para o PCVFS, portanto para realização de experimentos computacionais com as formulações apresentadas para resolução do PCVFS, foi criado conjunto de instâncias a partir de gerador de instâncias pseudo-aleatório. O valor de w, que indica o número de compartilhamentos, neste conjunto, possui valores entre 1 a 4. Os valores das funções f e d variaram entre valores de 1 a 10 e entre valores de 1 a 6, respectivamente.

Quadro 2 – Variações de α e β

	α	β
Grupo 1	0.25	0.75
Grupo 2	0.5	0.5
Grupo 3	0.75	0.25

Fonte: Elaborado pelo Autor (2018).

6.3 Estudo comparativo entre as formulações

Como mencionado na Seção 5.2, as formulações propostas se diferenciam basicamente de como são implementada as RECs, DFJ ou MTZ. As tabelas fornecem resultados para os diferentes grupos de instâncias de acordo com o quadro 1, onde para cada grupo foram sorteados valores para α e β de acordo com o quadro 2:

- *inst*: representa a instância em consideração, é definido como $|V|niID$, onde o ID é um identificador da instância;
- z: representa o valor ótimo da função objetivo;
- DFJ: informações relativas ao DFJ;
- MTZ: informações relativas ao MTZ;
- *bb*: representa o número de subproblemas relaxado;
- cpu: representa o tempo de execução, em segundos, para obtenção do valor ótimo;
- gap: representa o diferença, em %, entre a relaxação linear do problema e o valor ótimo;
- n/a: não se aplica durante o tempo válido.

Tabela 1 – Grafos densos parte 1

		(DFJ)			(MTZ)		
inst	*z*	*bb*	*cpu*	*gap*	*bb*	*cpu*	*gap*
100n01	5.00	0	0.56	15.00	0	0.28	15.00
100n02	5.00	2101	112.43	30.00	0	**0.39**	29.90
100n03	8.50	0	0.28	0.00	0	0.32	0.00
100n04	7.00	0	0.30	0.00	0	0.21	0.00
100n05	5.00	0	0.15	0.00	0	0.04	0.00
100n06	8.00	68	6.31	4.69	0	**0.32**	4.67
100n07	5.50	70756	1854.73	18.18	0	**0.32**	18.18
100n08	6.00	0	0.22	0.00	0	0.38	0.00
100n09	4.00	77	5.93	12.50	0	**0.39**	12.50
100n10	8.00	0	0.30	0.00	0	0.31	0.00
120n01	5.25	0	3.60	0.00	0	**0.47**	0.00
120n02	8.25	5002	207.74	18.18	0	**3.23**	18.11
120n03	4.75	0	1.46	5.26	0	0.42	5.26
120n04	2.75	0	1.06	6.82	0	0.34	6.82
120n05	5.00	0	0.44	0.00	0	0.35	0.00

Fonte: Elaborado pelo Autor (2018).

Tabela 2 – Grafos densos parte 2

		(DFJ)			(MTZ)		
inst	*z*	*bb*	*cpu*	*gap*	*bb*	*cpu*	*gap*
120n06	7.25	59	4.43	15.52	0	**0.89**	15.49
120n07	4.50	0	0.54	0.00	0	0.36	0.00
120n08	2.50	4	2.15	35.00	0	0.30	35.00
120n09	4.75	0	0.28	0.00	0	0.32	0.00
120n10	3.75	0	0.74	11.11	0	0.49	11.11
140n01	4.50	0	0.62	0.00	0	0.48	0.00
140n02	7.25	0	1.11	6.21	0	1.11	6.21
140n03	5.25	0	0.69	4.76	0	0.98	4.76
140n04	7.50	0	0.89	1.67	0	0.64	1.67
140n05	2.75	0	0.70	0.00	0	0.28	0.00
140n06	5.25	0	0.49	0.00	0	0.31	0.00
140n07	3.75	0	0.48	0.00	0	0.34	0.00
140n08	5.25	5	3.83	8.93	0	**0.58**	8.93
140n09	4.00	0	0.59	3.13	0	0.56	3.13
140n10	6.00	4	1.23	6.25	0	0.60	6.25

Fonte: Elaborado pelo Autor (2018).

Tabela 3 – Grafos densos parte 3

		(DFJ)			(MTZ)		
inst	*z*	*bb*	*cpu*	*gap*	*bb*	*cpu*	*gap*
160n01	5.50	6	6.09	9.09	0	**0.79**	9.05
160n02	2.00	0	2.20	0.00	0	0.46	0.00
160n03	5.00	0	1.27	0.00	0	0.61	0.00
160n04	2.50	4	11.04	20.00	0	**0.37**	20.00
160n05	4.00	0	1.22	0.00	0	0.68	0.00
160n06	3.00	2	4.38	0.00	0	**0.64**	0.00
160n07	2.50	1160	318.26	20.00	0	**0.61**	20.00
160n08	2.00	0	2.49	0.00	0	**0.46**	0.00
160n09	4.00	159	29.29	12.50	0	**0.63**	12.50
160n10	1.50	0	1.94	0.00	0	0.40	0.00
180n01	4.50	0	2.80	0.00	0	0.83	0.00
180n02	4.00	0	2.69	0.00	0	1.15	0.00
180n03	5.75	2	5.75	5.22	0	**0.78**	5.22
180n04	3.75	0	3.56	3.33	0	**1.03**	3.33
180n05	3.25	0	2.65	0.00	0	0.70	0.00

Fonte: Elaborado pelo Autor (2018).

Tabela 4 – Grafos densos parte 4

		(DFJ)			(MTZ)		
inst	*z*	*bb*	*cpu*	*gap*	*bb*	*cpu*	*gap*
180n06	4.50	43	13.32	8.89	0	**1.70**	8.89
180n07	2.50	0	3.06	10.00	0	**0.69**	10.00
180n08	3.00	3	12.02	8.33	0	**1.89**	8.33
180n09	4.50	210	157.81	11.11	0	**0.92**	11.11
180n10	2.75	0	6.99	12.12	0	**0.52**	12.12
200n01	1.50	426	870.11	33.33	0	**0.64**	33.25
200n02	3.50	0	3.54	0.00	0	**0.81**	0.00
200n03	1.75	0	4.94	0.00	0	**1.64**	0.00
200n04	2.00	0	4.04	0.00	0	**0.34**	0.00
200n05	4.75	n/a	n/a	15.79	0	**2.03**	15.79
200n06	4.75	5	10.88	5.26	0	**1.07**	5.26
200n07	2.50	0	2.53	0.00	0	0.75	0.00
200n08	2.75	61	39.87	22.73	0	**1.34**	22.73
200n09	2.75	1	5.68	4.55	0	**1.03**	4.55
200n10	4.50	5051	2375.92	11.11	0	**1.17**	11.11

Fonte: Elaborado pelo Autor (2018).

Tabela 5 – Grafos esparsos parte 1

		(DFJ)			(MTZ)		
inst	*z*	*bb*	*cpu*	*gap*	*bb*	*cpu*	*gap*
100n01	9.00	25	1.08	3.70	0	0.29	3.70
100n02	4.50	0	0.16	11.11	0	0.16	11.04
100n03	4.50	0	0.23	16.67	0	0.24	16.61
100n04	11.00	0	0.28	1.52	0	0.12	1.52
100n05	5.50	0	0.10	2.27	0	0.15	2.27
100n06	5.00	0	0.06	0.00	0	0.11	0.00
100n07	2.50	0	0.09	0.00	0	0.06	0.00
100n08	5.00	0	0.06	0.00	0	0.16	0.00
100n09	3.50	0	0.20	9.52	0	0.14	9.52
100n10	4.00	0	0.14	25.00	0	0.15	24.87
120n01	7.00	1606	21.24	7.14	0	**0.18**	7.11
120n02	6.00	0	0.10	0.00	0	0.14	0.00
120n03	7.00	0	0.31	0.00	0	0.19	0.00
120n04	3.00	0	0.28	27.78	0	0.12	27.78
120n05	5.50	12	0.94	14.55	0	0.27	14.51

Fonte: Elaborado pelo Autor (2018).

Tabela 6 – Grafos esparsos parte 2

		(DFJ)			(MTZ)		
inst	*z*	*bb*	*cpu*	*gap*	*bb*	*cpu*	*gap*
120n06	7.50	0	0.34	0.00	0	0.20	0.00
120n07	3.00	0	0.15	8.33	0	0.18	8.33
120n08	6.50	3	0.77	7.69	0	0.26	7.67
120n09	5.00	11217	77.00	20.00	0	**0.19**	19.92
120n10	3.00	0	0.16	5.56	0	0.13	5.56
140n01	7.50	0	0.61	5.00	0	0.29	4.96
140n02	4.50	0	0.15	0.00	0	0.21	0.00
140n03	6.75	189	4.81	11.11	0	**0.51**	11.11
140n04	7.50	0	0.15	0.00	0	0.23	0.00
140n05	3.00	0	0.31	0.00	0	0.15	0.00
140n06	8.00	0	1.12	6.25	0	0.40	6.24
140n07	9.00	24	1.27	2.78	0	0.28	2.78
140n08	6.25	0	0.16	0.00	0	0.24	0.00
140n09	7.00	0	0.87	5.95	0	0.25	5.95
140n10	6.00	0	0.17	4.17	0	0.19	4.17

Fonte: Elaborado pelo Autor (2018).

Tabela 7 – Grafos esparsos parte 3

		(DFJ)			(MTZ)		
inst	*z*	*bb*	*cpu*	*gap*	*bb*	*cpu*	*gap*
160n01	4.25	0	0.67	5.88	0	0.32	5.88
160n02	6.75	0	0.24	1.85	0	0.24	1.85
160n03	6.25	47	9.25	12.00	0	**0.49**	11.97
160n04	5.50	16	3.29	11.69	0	**0.82**	11.69
160n05	7.00	0	0.62	5.00	0	0.27	5.00
160n06	4.50	0	0.23	5.56	0	0.22	5.56
160n07	3.25	0	0.39	0.00	0	0.21	0.00
160n08	6.00	0	0.23	0.00	0	0.27	0.00
160n09	7.25	11	1.12	5.17	0	0.50	5.17
160n10	7.00	0	0.60	2.55	0	0.47	2.55
180n01	3.50	0	0.77	14.29	0	0.22	14.29
180n02	7.50	0	0.57	0.83	0	0.36	0.83
180n03	3.75	0	1.08	6.67	0	0.33	6.67
180n04	2.50	0	0.53	0.00	0	0.25	0.00
180n05	6.25	0	0.90	2.67	0	0.58	2.67

Fonte: Elaborado pelo Autor (2018).

Tabela 8 – Grafos esparsos parte 4

		(DFJ)			(MTZ)		
inst	*z*	*bb*	*cpu*	*gap*	*bb*	*cpu*	*gap*
180n06	5.75	0	1.71	14.49	0	1.05	14.49
180n07	4.00	0	0.73	0.00	0	0.20	0.00
180n08	4.25	0	0.36	0.00	0	0.28	0.00
180n09	2.50	0	0.78	0.00	0	0.28	0.00
180n10	4.75	0	0.92	15.79	0	**0.47**	15.79
200n01	6.00	0	1.21	0.00	0	0.31	0.00
200n02	5.25	0	0.82	0.00	0	0.93	0.00
200n03	6.75	0	1.24	0.00	0	0.41	0.00
200n04	3.75	0	0.53	6.67	0	0.48	6.67
200n05	5.25	0	0.55	2.38	0	0.69	2.38
200n06	3.00	38814	2039.52	25.00	0	**0.42**	25.00
200n07	3.00	0	0.73	0.00	0	0.10	0.00
200n08	6.50	124	18.53	7.69	0	**1.77**	7.69
200n09	3.00	0	1.19	8.33	0	0.52	8.32
200n10	3.75	9	4.15	8.00	0	**0.34**	8.00

Fonte: Elaborado pelo Autor (2018).

7 CONCLUSÕES E TRABALHOS FUTUROS

7.1 Conclusões

Neste trabalho foi definido o Problema de Compartilhamento de Veículos com Restrições Físicas e Sociais (PCVFS). Foram implementadas duas formulações matemáticas para o problema e apresentado os experimentos computacionais. Os dois modelos se diferenciam da forma a qual é implementado as RECs. A primeira formulação implementada realiza a quebra de ciclos através das restrições de DFJ. Devido ao número exponencial de restrições, foi utilizado um plano de corte através de uma separação, onde a verificação polinomial das restrições violadas foi realizada por meio de um algoritmo de detecção de conjuntos conexos. A segunda formulação implementa a quebra de ciclos através das restrições MTZ, que consiste em rotular os vértices escolhidos para o caminho, afim de evitar ciclos.

Nos testes realizados com instâncias aleatórias, o modelo implementado com o MTZ obteve melhores resultados em comparação ao implementado com o DFJ. Pode-se concluir que o modelo MTZ se mostrou mais rápido que o modelo DFJ. A diferença entre os tempos de resolução dos modelos aumentou de forma proporcional ao número de vértices relacionado a cada grupo de instâncias. Também foi possível notar através dos experimentos, de maneira geral, que as instâncias que possuíam grafos esparsos obtiveram um tempo menor de execução que as instâncias que possuíam grafos densos.

7.2 Trabalhos Futuros

Para trabalhos futuros pode-se realizar novas formulações para o PCVFS e realizar testes das formulações com instâncias reais para o problema. Estudo e aplicações de técnicas de Programação Linear Inteira para resolução do PCVFS, por exemplo, busca de desigualdades válidas para a criação de métodos de B&C, estudo Poliedro do problema, assim como a busca de possíveis facetas advindas das desigualdades válidas.

REFERÊNCIAS

ANDRADE, R. C. de. New formulations for the elementary shortest-path problem visiting a given set of nodes. **European Journal of Operational Research**, Elsevier, v. 254, n. 3, p. 755–768, 2016.

ARAÚJO, G. F. de. **Algoritmos Meta-heurísticos Para a solução do Problema do Caxeiro Viajante com Múltiplas Caronas**. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Norte, Natal, 2016.

AUSIELLO, G.; DEMANGE, M.; LAURA, L.; PASCHOS, V. Algorithms for the on-line quota traveling salesman problem. **Information Processing Letters**, Elsevier, v. 92, n. 2, p. 89–94, 2004.

AWERBUCH, B.; AZAR, Y.; BLUM, A.; VEMPALA, S. New approximation guarantees for minimum-weight k-trees and prize-collecting salesmen. **SIAM Journal on computing**, SIAM, v. 28, n. 1, p. 254–262, 1998.

BALAS, E. The prize collecting traveling salesman problem. **Networks**, Wiley Online Library, v. 19, n. 6, p. 621–636, 1989.

BERTSIMAS, D.; TSITSIKLIS, J. N. **Introduction to linear optimization**. [S.l.]: Athena Scientific Belmont, MA, 1997. v. 6.

CAIXETA-FILHO, J. V. **Pesquisa Operacional**. [S.l.]: São Paulo: Atlas, 2001.

CAPELAS. **BlaBlaCar chega ao Brasil com foco em economia compartilhada ?de verdade?** 2015. Disponível em: <http://link.estadao.com.br/noticias/empresas,blablacar-chega-ao-\%09brasil-com-foco-em-economia-compartilhada-de-verdade,10000028798>. Acesso em: 2017-27-10.

COSTA, M. d. S. Mobilidade urbana sustentável: um estudo comparativo e as bases de um sistema de gestão para brasil e portugal. **EESC/USP. São Paulo**, 2003.

DAILEY, D.; LOSEFF, D.; MEYERS, D. Seattle smart traveler: dynamic ridematching on the world wide web. **Transportation Research Part C: Emerging Technologies**, Elsevier, v. 7, n. 1, p. 17–32, 1999.

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. **Computational algorithm of the revised simplex method**. [S.l.]: Rand Corporation, 1953.

DEBATIN, A. N.; ANDRADE, L. D. de. **Política de Planejamento de Transportes e Desenvolvimento Urbano**: Considerações para a cidade de florianópolis. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, 1998.

DENATRAN - Departamento de Trânsito de Veículos. **Frotas de Veículos**. 2016. Disponível em: <http://www.denatran.gov.br/frota.htm>. Accessado: 24 out. 2017.

FURTADO, J. C.; LUIZ, D. J. O. Otimização da mobilidade urbana através da modelagem e implementação de sistema em rede social para compartilhamento de transporte. **Anais do Salão de Ensino e de Extensão**, p. 329, 2013.

FURUHATA, M.; DESSOUKY, M.; ORDÓÑEZ, F.; BRUNET, M.-E.; WANG, X.; KOENIG, S. Ridesharing: The state-of-the-art and future directions. **Transportation Research Part B: Methodological**, Elsevier, v. 57, p. 28–46, 2013.

GARCIA, S.; GUERREIRO, R.; CORRAR, L. J. Teoria das restrições e programação linear. In: **Congresso Internacional de Costos**. [S.l.: s.n.], 1997.

GILMORE, P. C.; GOMORY, R. E. A linear programming approach to the cutting-stock problem. **Operations research**, INFORMS, v. 9, n. 6, p. 849–859, 1961.

HILLIER, F. S.; LIEBERMAN, G. J. **Introdução à pesquisa operacional**. [S.l.]: McGraw Hill Brasil, 2013.

JOZEFOWIEZ, N.; GLOVER, F.; LAGUNA, M. Multi-objective meta-heuristics for the traveling salesman problem with profits. **Journal of Mathematical Modelling and Algorithms**, Springer, v. 7, n. 2, p. 177–195, 2008.

LUNA, H. P.; GOLDBARG, M. C. Otimização combinatória e programação linear. **Rio de Janeiro: Campus**, 2000.

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.

PAPADIMITRIOU, C. H.; STEIGLITZ, K. **Combinatorial optimization: algorithms and complexity**. [S.l.]: Courier Corporation, 1998.

PUCCINI, A. d. L. **Introdução à Programação Linear**. [S.l.]: Rio de Janeiro: LTC, 1980.

SCARINGELLA, R. S. A crise da mobilidade urbana em são paulo. **São Paulo em perspectiva**, SciELO Brasil, v. 15, n. 1, p. 55–59, 2001.

WOLSEY, L. A. **Integer programming**. [S.l.]: Wiley, 1998.

YU, W.; LIU, Z.; BAO, X. Optimal deterministic algorithms for some variants of online quota traveling salesman problem. **European Journal of Operational Research**, Elsevier, v. 238, n. 3, p. 735–740, 2014.