

UNIVERSIDADE FEDERAL DE GOIÁS  
INSTITUTO DE INFORMÁTICA

MARCELO RUBENS GIESEL

**O Problema do Caminho Mínimo com  
Restrição de Subcaminhos de  
Comprimento Dois**

Goiânia  
2010

UNIVERSIDADE FEDERAL DE GOIÁS  
INSTITUTO DE INFORMÁTICA

**AUTORIZAÇÃO PARA PUBLICAÇÃO DE TRABALHO DE  
CONCLUSÃO DE CURSO EM FORMATO ELETRÔNICO**

Na qualidade de titular dos direitos de autor, **AUTORIZO** o Instituto de Informática da Universidade Federal de Goiás – UFG a reproduzir, inclusive em outro formato ou mídia e através de armazenamento permanente ou temporário, bem como a publicar na rede mundial de computadores (*Internet*) e na biblioteca virtual da UFG, entendendo-se os termos “reproduzir” e “publicar” conforme definições dos incisos VI e I, respectivamente, do artigo 5º da Lei nº 9610/98 de 10/02/1998, a obra abaixo especificada, sem que me seja devido pagamento a título de direitos autorais, desde que a reprodução e/ou publicação tenham a finalidade exclusiva de uso por quem a consulta, e a título de divulgação da produção acadêmica gerada pela Universidade, a partir desta data.

**Título:** O Problema do Caminho Mínimo com Restrição de Subcaminhos de Comprimento Dois

**Autor(a):** Marcelo Rubens Giesel

Goiânia, 02 de Dezembro de 2010.

---

Marcelo Rubens Giesel – Autor

---

Humberto José Longo – Orientador

---

Hugo Alexandre D. do Nascimento – Co-Orientador

MARCELO RUBENS GIESEL

# **O Problema do Caminho Mínimo com Restrição de Subcaminhos de Comprimento Dois**

Trabalho de Conclusão apresentado à Coordenação do Curso de Ciências da Computação do Instituto de Informática da Universidade Federal de Goiás, como requisito parcial para obtenção do título de Bacharel em Ciências da Computação.

**Área de concentração:** Grafos.

**Orientador:** Prof. Humberto José Longo

**Co-Orientador:** Prof. Hugo Alexandre D. do Nascimento

Goiânia  
2010

MARCELO RUBENS GIESEL

# O Problema do Caminho Mínimo com Restrição de Subcaminhos de Comprimento Dois

Trabalho de Conclusão apresentado à Coordenação do Curso de Ciências da Computação do Instituto de Informática da Universidade Federal de Goiás como requisito parcial para obtenção do título de Bacharel em Ciências da Computação, aprovada em 02 de Dezembro de 2010, pela Banca Examinadora constituída pelos professores:

---

**Prof. Humberto José Longo**

Instituto de Informática – UFG

Presidente da Banca

---

**Prof. Hugo Alexandre D. do Nascimento**

Instituto de Informática – UFG

---

**Prof. Thierson Couto Rosa**

Instituto de Informática – UFG

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador(a).

### **Marcelo Rubens Giesel**

Em sua graduação, na UFG – Universidade Federal de Goiás, foi monitor das disciplinas de Matemática Discreta e Teoria dos Grafos do Instituto de Informática – INF, foi pesquisador do CNPq em um trabalho de iniciação científica através de uma parceria UFG/PUC-Rio, eleito representante dos alunos de Ciência da Computação junto ao Conselho Diretor do INF, presidente do Centro Acadêmico de Ciência da Computação, integrante da primeira equipe a representar o INF/UFG na Maratona de Programação – competição brasileira da *International Collegiate Programming Contest*, organizada pela *Association for Computing Machinery* (ACM-ICPC) – e da primeira equipe do INF/UFG a se qualificar para uma fase nacional da mesma competição, participou da organização de dois ETI's – Encontro de Tecnologia e Informática – na UFG e realizou intercâmbio estudantil pelo período de um ano na Universidade do Algarve, Portugal, ao abrigo do Programa Luso-Brasileiro Santander Universidade.

Aos meus pais, por me ensinar, demonstrar e cultivar os valores de uma vida em família.

---

## Agradecimentos

---

Agradeço primeiramente a Deus, por me dar vida, saúde, paz, amigos e uma família sólida. Por olhar por mim e por eles, por me dar oportunidades, por me oferecer um futuro e por tudo mais.

Ao Instituto de Informática, funcionários e professores. Ao coordenador Plínio e ao diretor Auri, pelo auxílio nas questões burocrática. Aos professores Vagner, Fábio Moreira, Cláudio e Diane, pelo projeto de iniciação científica, pelas maratonas de programação e monitorias, pela qualidade de ensino e dedicação ao curso.

Ao meu co-orientador Hugo, ao meu avaliador Thierson e, principalmente, ao meu orientador Humberto, pelo empenho, dedicação e qualidade de ensino nas disciplinas ministradas, pela prestatividade, pelo grandioso auxílio na elaboração deste projeto, pelas maratonas de programação, pelo incentivo à realização do intercâmbio estudantil e pelas conversas informais.

Aos meus colegas de curso, com os quais convivi toda minha graduação e compartilhei bons e maus momentos. Aos amigos que deixei em outros lugares, para que fosse possível seguir com meus estudos. Aos meus irmãos e parceiros Heitor, Thiago e Salomão.

À Carol, minha namorada, pela ajuda na elaboração deste documento, pelo incentivo à realização do intercâmbio e pela presença no mesmo, o melhor ano de nossas vidas. Pelo carinho, pelo companheirismo e por todos os nossos momentos.

À minha família, que sempre esteve ao meu lado, mesmo distante fisicamente. Que sempre me incentivou e apoiou, tanto na minha vida acadêmica quanto na vida pessoal. Por me darem a certeza de que, independente do que aconteça, estaremos sempre unidos. Obrigado mãe, pai e mano.

O pessimista reclama do vento; o otimista espera ele mudar; o realista ajusta as velas.

**William George Ward,**  
*Teólogo e matemático inglês.*



---

## Resumo

---

Giesel, Marcelo Rubens. **O Problema do Caminho Mínimo com Restrição de Subcaminhos de Comprimento Dois**. Goiânia, 2010. 44p. Relatório de Graduação. Instituto de Informática, Universidade Federal de Goiás.

Este projeto tem por objetivo estudar o Problema do Caminho Mínimo com Restrição de Subcaminhos de Comprimento Dois, abordando algumas variações deste problema e a relação entre eles. Algumas soluções já propostas serão estudadas, novas modelagens apresentadas e comparadas, a fim de possibilitar a escolha da melhor abordagem para cada caso de uso específico. Como alvo principal, este trabalho visa encontrar a solução que mais se adequa ao problema de escolha de menor caminho em uma malha de trânsito urbana.

### Palavras-chave

Caminho Mínimo com Restrição de Subcaminhos, Caminho Mínimo, Rotas, Grafo, Tráfego Urbano

---

## Abstract

---

Giesel, Marcelo Rubens. **The Shortest Path Problem with Forbidden Paths of Length Two**. Goiânia, 2010. 44p. Relatório de Graduação. Instituto de Informática, Universidade Federal de Goiás.

This project aims at studying the Shortest Path Problem with Forbidden Paths of Length Two addressing some variations of this problem and the relationship between them. Some solutions already proposed will be studied, new models will be presented and compared in order to allow choosing the best algorithm for each specific case of use. As primary target, this work aims to find the solution that best fits the problem of choosing the shortest path in a mesh of urban traffic.

### Keywords

Shortest Path Problem with Forbidden Paths, Shortest Path, Graph, Route, Urban Traffic

---

# Sumário

---

Lista de Figuras	10
1 Introdução	11
2 O Problema do Caminho Mínimo e Suas Variantes	14
2.1 O Problema do Caminho Mínimo com Restrição de Recursos	15
2.2 O Problema do Caminho Evitando Pares Proibidos	17
2.3 O Problema do Caminho Mais Longo	18
2.4 O Problema do Caminho Mínimo com Restrição de Subcaminhos	18
3 Problema do Caminho Mínimo com Restrição de Subcaminhos de Comprimento Dois	20
3.1 Modelagem 1	23
3.1.1 Análise	26
3.2 Modelagem 2	27
3.2.1 Análise	28
3.3 Modelagem 3	30
3.3.1 Análise	32
3.4 Modelagem 4	34
3.4.1 Análise	37
4 Conclusão	39
4.1 Comparações com Resultados de um Caso de Teste	40
4.2 Trabalhos Futuros	42
Referências Bibliográficas	43

---

## Lista de Figuras

---

2.1	Grafo para o Problema da Mochila.	17
3.1	Mapa com caminho proibido no centro de Goiânia.	21
3.2	Grafo com caminho proibido no centro de Goiânia.	21
3.3	Grafo original.	23
3.4	Mapeamento de arcos em $G$ para vértices em $G'$ .	24
3.5	Inclusão dos primeiros arcos em $G'$ .	24
3.6	Inclusão dos vértices “externos” de $G'$ .	25
3.7	Atribuindo arcos aos vértices “externos” de $G'$ .	25
3.8	Grafo $G'$ após inclusão dos primeiros vértices e dos primeiros arcos.	27
3.9	Grafo $G'$ após inclusão do vértice origem e seus arcos.	27
3.10	Grafo $G'$ com todos os vértices e arcos.	28
3.11	Criação dos vértices iniciais e custos de seus arcos.	30
3.12	Adicionando o vértice origem e custo de suas arestas.	30
3.13	Grafo auxiliar final.	31
3.14	Grafo $H$ .	34
3.15	Grafo $H'$ , com uma quantidade maior de arcos que $H$ .	34
3.16	Criação dos vértices e dos arcos inter-vértices.	35
3.17	Grafo $G'$ completo.	36
3.18	Grafo $G'$ já sem caminhos proibidos.	36
3.19	Grafo $G'$ final.	37
4.1	Grafo $G$ utilizado para comparações entre as modelagens.	41

---

## Introdução

---

Um problema muito importante da teoria dos grafos é o Problema do Caminho Mínimo (SPP – *Shortest Path Problem*), que consiste em encontrar o caminho mais curto entre uma origem e um destino. Este problema é aplicado em várias áreas do conhecimento, não somente na computação, e há uma enorme quantidade de estudos realizados que o solucionam.

Conforme essas soluções para o SPP foram propostas, novas necessidades surgiam e foram criadas variações do problema para resolverem tais classes do SPP. Uma dessas variantes, o Problema do Caminho Mínimo com Restrição de Subcaminhos (SPPFP – *Shortest Path Problem with Forbidden Paths*), é o objeto de estudo deste projeto. Nessa variação o objetivo é encontrar um caminho de menor custo entre uma origem e um destino especificados, sem que este caminho passe por algum subcaminho proibido. Embora com uma quantidade considerável de pesquisas a seu respeito, o SPPFP ainda necessita de estudos, principalmente quando voltado para aplicações específicas.

Partindo da definição do problema no parágrafo anterior, Villeneuve e Desaulniers [22] propuseram uma solução para o SPPFP na qual utilizam o algoritmo de reconhecimento de padrões de Aho e Corasick [2] para filtrar, através de um grafo de estados, os caminhos produzidos por um algoritmo de caminho mínimo. Se um dado caminho mínimo é aceito pelo filtro (ou seja, é reconhecido pelo grafo de estados), este caminho mínimo não possui trechos proibidos e é válido. Se, por outro lado, um caminho não é aceito pelo filtro (ou seja, não é reconhecido pelo grafo de estados), este caminho mínimo possui algum subcaminho proibido, não sendo, portanto, um caminho mínimo válido. Quando esta última condição ocorre, o algoritmo de Aho e Corasick identifica o trecho proibido do caminho não aceito. Para contornar este caminho não-válido o algoritmo de Villeneuve e Desaulniers utiliza a abordagem de desvio de caminho de Martins [17] para construir uma rota alternativa e atualizar as informações de rotas (altera o grafo e seus menores caminhos), evitando assim que novos caminhos mínimos percorram o trecho proibido. Esta abordagem é executada todas as vezes que o grafo de estados identifica um trecho proibido em um caminho mínimo.

Uma outra solução, proposta por Ahmed e Lubiw [1], também utiliza a aborda-

gem de Martins para não percorrer um caminho proibido quando este é encontrado em um caminho mínimo. Entretanto, diferentemente da solução de Villeneuve e Desaulniers, a solução de Ahmed e Lubiw não faz uso de um conjunto de caminhos proibidos previamente conhecido. Por este motivo não é possível a criação de um grafo de estados que reconheça cada subcaminho proibido (algoritmo de Aho e Corasick) do grafo original. Para que sejam encontrados tais caminhos esta solução utiliza a técnica de tentativa-e-erro: quando uma tentativa de percorrer um caminho acarreta em algum erro, tal caminho é tratado como caminho proibido e um desvio é criado.

A aplicação desta última solução mostra-se ideal em redes óticas, para a qual foi criada, uma vez que retransmissões, apesar de indesejáveis, são possíveis, e onde também é impossível de se ter um conhecimento prévio de todos os caminhos proibidos. Neste caso, quando uma mensagem enviada por um caminho é perdida, tal caminho é tratado como caminho proibido. Entretanto, em um cenário onde retransmissões são impossíveis e a tomada de um caminho proibido é inadmissível, mesmo na sua descoberta, a primeira solução se mostra mais viável.

Walid, em seu trabalho intitulado “Uma Arquitetura de Software Interativo para Apoio à Decisão na Modelagem e Análise do Tráfego Urbano” [13], trata de um problema similar de restrição de caminhos de comprimento dois (quantidade de arestas) e o resolve de outra forma. A malha urbana é modelada em um grafo direcionado, aresta valorado, no qual os vértices representam as intersecções de ruas e avenidas, os arcos representam os trechos<sup>1</sup> de ruas e avenidas. Se qualquer caminho neste grafo é permitido, tal representação não condiz com a realidade do trânsito. Partindo do pressuposto de que, em uma malha urbana, todos os caminhos proibidos são caminhos de tamanho igual a dois, Walid constrói um grafo auxiliar a partir do grafo original. O grafo auxiliar é formado apenas pelos caminhos de tamanho igual a dois que correspondem às conversões permitidas nas vias de uma cidade. Como este grafo não contém conversões proibidas, ele forma apenas caminhos válidos para a malha urbana da qual foi modelado. Qualquer caminho no grafo auxiliar – e consequentemente na malha urbana – pode ser formado através da concatenação dos caminhos de tamanho dois representados neste grafo.

Embora esta solução resolva o problema de evitar caminhos proibidos de tamanho dois, não foram realizadas comparações com as outras abordagens, resultando em um desconhecimento da eficiência relativa entre tais abordagens.

Como a proposta deste projeto é estudar rotas em uma malha de trânsito urbana, o caso geral não será atacado, mas sim o subcaso específico onde todos os caminhos proibidos possuem tamanho menor ou igual a dois. Assim, qualquer caminho em um

---

<sup>1</sup>Um trecho de uma via é um segmento que inicia em uma intersecção com uma segunda via e estende-se até a sua intersecção mais próxima com uma terceira via.

grafo é formado ou por uma ligação direta entre dois vértices ou por uma combinação de caminhos de comprimento dois. Esta é a ideia que dá base para a formulação ou modificação das abordagens que propõem uma solução do Problema do Caminho Mínimo com Restrição de Subcaminhos e que serão objeto de estudo neste projeto.

A estrutura deste documento é descrita a seguir. O Capítulo 2 apresenta uma introdução ao Problema do Caminho Mínimo, juntamente de sua descrição formal; ele descreve também quatro variantes deste problema e suas complexidades. O Capítulo 3 aborda o Problema do Caminho Mínimo com Restrição de Subcaminhos de Comprimento Dois, variante do Problema do Caminho Mínimo que pode ser utilizado no trânsito urbano. Este Capítulo apresenta quatro modelagens provenientes de [6, 10] para a solução do subcaso e as avalia quanto à dimensão do grafo resultante. O Capítulo 4 apresenta as conclusões obtidas neste projeto e cita os trabalhos sobre o tema a serem realizados no futuro.

## O Problema do Caminho Mínimo e Suas Variantes

---

O Problema do Caminho Mínimo (SPP – *Shortest Path Problem*) tem por objetivo encontrar o caminho de menor custo entre uma origem e um destino. O custo é um valor que queremos minimizar, podendo ser a distância que uma entidade percorre ao sair da origem e chegar ao destino, a quantidade de intersecções de um caminho, o tempo gasto em um trajeto, a quantidade de combustível consumida, entre outros. Este valor é associado aos trechos de caminhos, indicando o custo de uma transição entre dois pontos específicos. O custo total de um caminho é obtido através da soma dos custos de transições consecutivas.

Formalmente, dado um grafo dirigido e aresta valorado,  $G(V, A)$ , onde  $V$  é o conjunto de vértices,  $A$  é o conjunto de arestas em  $G$  e  $c(s, t)$ ,  $c : A \rightarrow \mathbb{R}$ , a função que retorna o custo para percorrer um caminho entre dois vértices  $s$  e  $t$  em  $G$ . O objetivo é encontrar o menor valor de  $c(s, t)$ , para um par de vértices  $s$  e  $t$  especificado. Fazendo uso da observação de que uma porção  $u \rightsquigarrow v$  de um caminho mais curto  $s \rightsquigarrow t$  deve ser um caminho mais curto de  $u$  a  $v$ , os algoritmos que resolvem este problema encontram rotas ótimas de  $s$  a  $t$  em uma ordem crescente da função custo  $c$ , resultante da somatória dos menores caminhos intermediários [23]. Essa propriedade de sub-estrutura ótima é uma característica da aplicabilidade da programação dinâmica (como o algoritmo de Floyd-Warshall [8]) e do método guloso (como o algoritmo de Dijkstra [5]) ao problema de se encontrar o caminho mínimo [4].

O SPP é um problema importante de grafos, muito utilizado e explorado em várias áreas do conhecimento, como computação (escolha da rota mais curta para envio de pacotes de dados), telecomunicações (estabelecer uma ligação de voz que apresente a menor latência), transportes (rota mais econômica para transporte de mercadorias), logística (forma mais barata de armazenar e enviar mercadorias para um destino), biologia (descobrir a menor distância entre células de um vírus), economia, administração e outras. Em todas elas, deseja-se minimizar algum custo e há necessidades de ferramentas que auxiliem no processo de minimização deste custo.



Algumas soluções para os casos acima e ferramentas que implementam soluções estão presentes na ciência da computação, onde o SPP tem sido estudado por um longo período de tempo. Diferentes algoritmos de complexidade polinomial no número de vértices e arestas foram propostos para resolver o SPP. Para grafos aresta valorados, com custos de transição não-negativos, o algoritmo de Dijkstra [5] encontra o menor caminho de um vértice para todos os outros vértices do grafo. O algoritmo de Bellman-Ford [3, 9] faz o mesmo, com o adicional de que pode ser utilizado em grafos que possuem custos de transição negativos. Já o algoritmo de Floyd-Warshall [8] é utilizado para descobrir o menor caminho entre todos os pares de vértices de um grafo.

Dadas a importância e a abrangência do SPP, serão apresentadas, nas seções seguintes, algumas de suas variantes. A Seção 2.1 descreve o Problema do Caminho Mínimo com Restrição de Recursos, uma das primeiras variantes que apareceram; a Seção 2.2 apresenta o Problema do Caminho Evitando Pares Proibidos, no qual dois pontos não podem estar mutuamente presentes em um mesmo caminho; a Seção 2.3 mostra o Problema do Caminho Mais Longo, uma espécie de SPP ao inverso; por final, a Seção 2.4 introduz o Problema do Caminho Mínimo com Restrição de Subcaminhos, que apresenta um conjunto de caminhos impossibilitados de serem percorridos. Esta última, aliás, é a variante que mais exige atenção, visto que um de seus subcasos é o objeto de estudo deste projeto.

## 2.1 O Problema do Caminho Mínimo com Restrição de Recursos

O Problema do Caminho Mínimo com Restrição de Recursos (SPPRC – *Shortest Path Problem with Resource Constraints*) é uma variante do Problema do Caminho Mínimo na qual deseja-se encontrar um caminho de menor custo possível entre uma origem e um destino, obedecendo os limites de consumo de um conjunto de recursos.

Um recurso é um bem, com um valor, que é consumido ao longo de um percurso. Por vezes, deseja-se que um recurso seja consumido o mínimo possível, demonstrando eficiência e evitando desperdícios. Em outros casos o recurso é um limitador e deseja-se obter o maior número possível de um determinado bem – ou de um conjunto de bens – até que o recurso chegue a um limite mínimo. Por exemplo, o combustível de um veículo de uma transportadora de cargas é um recurso para aquele veículo, pois o combustível será consumido ao longo do seu percurso e a quantidade dele limita a distância que pode ser percorrida por tal veículo.

O SPPRC pode ser definido em um grafo  $G = (V, A)$ , onde  $V = \{v_1, v_2, \dots, v_n\}$

é o conjunto de vértices e  $A = \{(v_i, v_j), \dots\}$  é o conjunto de arestas, incluindo um vértice origem  $s$  e um vértice destino  $t$ . Cada aresta  $(v_i, v_j) \in A$  tem um custo  $c_{ij} \in \mathbb{R}$  e uma quantidade  $d_{ij}^\ell \geq 0$ , referente ao consumo do recurso  $\ell \in \{1, \dots, L\}$ , associados ao percurso da aresta.  $L$  é o total de recursos disponíveis. Assume-se que os valores  $d_{ij}^\ell$  satisfazem a desigualdade triangular para cada recurso  $\ell$ , ou seja,  $d_{ij}^\ell < d_{ik}^\ell + d_{kj}^\ell$ ,  $\forall v_i, v_j, v_k \in V$ . Para cada vértice  $v_i \in V$  e cada recurso  $\ell$ , associam-se dois valores não negativos  $a_i^\ell$  e  $b_i^\ell$ , tal que o consumo do recurso  $\ell$  ao longo de um caminho de  $s$  para  $v_i$  deve pertencer ao intervalo  $[a_i^\ell, b_i^\ell]$ . O objetivo é encontrar um caminho elementar, de custo mínimo, do vértice origem  $s$  ao vértice destino  $t$  que satisfaça todas as restrições de recursos ao longo de seu trajeto [18].

Portanto, ao adicionar ao SPP limites de recursos – como tempo e orçamento – o problema se torna  $\mathcal{NP}$ -Completo [14, 19, 24]. Uma forma de demonstrar tal complexidade é fazer uma redução do Problema da Mochila (KP – *Knapsack Problem*) para o SPPRC [19, 24], uma vez que o KP é um problema  $\mathcal{NP}$ -Completo conhecido.

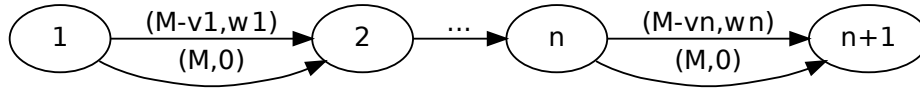
Conforme apresentado por Ziegelmann em [24], o Problema da Mochila consiste em, dado um conjunto  $I$  com uma quantidade  $n$  de itens, cada item  $i \in I$  com um peso  $w_i \in \mathbb{Z}^+$  e valor  $v_i \in \mathbb{Z}^+$  encontrar um subconjunto  $I' \subseteq I$  tal que  $\sum_{i \in I'} w_i \leq W$  e  $\sum_{i \in I'} v_i$  seja o maior possível. Este problema pode ser formulado como um problema de Programação Linear Inteira (PLI) como segue:

$$\begin{aligned} \text{Maximizar: } & \sum_{i=1}^n v_i x_i \\ \text{Sujeito a: } & \sum_{i=1}^n w_i x_i \leq W \\ & x_i \in \{0, 1\}, i = 1, \dots, n \end{aligned}$$

onde  $v_i, w_i, W$  são inteiros positivos e  $x_i$  é uma variável que recebe o valor 1 se o item  $i$  for incluído no conjunto  $I'$  e recebe o valor 0 caso o item  $i$  não seja incluído em  $I'$ . Para torná-lo um problema de Caminho Mínimo é construído um grafo de  $n + 1$  vértices com dois arcos paralelos de cada vértice  $i$  para o vértice  $i + 1$ , para  $i = 1, \dots, n$  (vide Figura 2.1). Escolhido  $M$  como o maior valor dentre todos os valores dos itens – ou seja, o valor do item mais “caro” – e, para o primeiro arco de cada vértice, a função de custo e de peso são definidas, respectivamente, como  $c(i, i + 1) = M - v_i$  e  $r(i, i + 1) = w_i$ . Para o segundo arco de cada vértice as mesmas funções são definidas como  $c(i, i + 1) = M$  e  $r(i, i + 1) = 0$ . Assim, caso um item seja incluído no conjunto de itens que irão para a mochila, o primeiro arco é escolhido e, caso contrário, o arco selecionado é o segundo.

Isto torna evidente que o Problema da Mochila pode ser solucionado através da descoberta de um caminho mínimo (levando em consideração a função de custo  $c$ ) do

vértice 1 ao vértice  $n + 1$ , sujeito à restrição de um recurso (considerando a função  $r$ ) que deve ser menor ou igual a  $W$ . Uma vez que o Problema da Mochila é  $\mathcal{NP}$ -Completo pode ser deduzido que o Problema do Caminho Mínimo com Restrição de Recursos é também  $\mathcal{NP}$ -Completo.



**Figura 2.1:** Grafo para o Problema da Mochila.

## 2.2 O Problema do Caminho Evitando Pares Proibidos

O problema surgiu com o nome de Problema do Caminho com Restrição de Pares Impossíveis (IPCPP – *Impossible Pairs Constrained Path Problem*) e sua aplicação era voltada para validação e teste automatizados de *software*. A ideia era tornar os testes mais eficientes considerando apenas aqueles caminhos de um programa que continham no máximo um ramo de cada par de ramos mutuamente não-executáveis [15]. Adicionalmente, era possível criar um conjunto de testes para percorrer todos os ramos de um programa, testando todas as suas saídas e validando assim seu funcionamento.

O Problema do Caminho Evitando Pares Proibidos (PAFPP – *Path Avoiding Forbidden Pairs Problem*) é um problema no qual deseja-se encontrar, em um grafo direcionado, um caminho que inicie em um vértice origem e termine em um vértice destino, de tal forma que nenhum dos pares de vértices proibidos esteja presente neste caminho escolhido. Vale ressaltar que um caminho  $P$  pode conter um vértice de um par proibido  $f$ , mas nunca os dois. Além disso, um vértice  $v$  selecionado em um par proibido é marcado como utilizado também em todos os outros pares proibidos em que se mostra presente. Assim, tomemos os pares  $f_1 = (v, w)$  e  $f_2 = (v, z)$  como pares proibidos, e um caminho  $P$  contendo o vértice  $w$  de  $f_1$ . Agora, olhando para  $f_2$ ,  $P$  não pode conter  $v$ , uma vez que a utilização deste vértice de  $f_2$  implica sua utilização também em  $f_1$  e o consequente aparecimento de  $f_1$  em  $P$ . Entretanto, caso o vértice  $v$  não esteja em  $P$ , os vértices  $w$  e  $z$  podem aparecer em  $P$  sem nenhum problema.

Seguindo a definição de [15], dado um grafo  $G(V, A)$ , onde  $V$  é o conjunto de vértices e  $A$  o conjunto de arcos, um vértice origem  $s$ , um vértice destino  $t$ , tal que  $s, t \in V$ , e um conjunto de pares de vértices proibidos  $F = \{(a, b), (c, d), \dots\}$ , tal que  $F \subset (V \times V)$ ,

o problema consiste em encontrar um caminho  $s \rightsquigarrow t$  que contenha no máximo um vértice de cada par em  $F$ , ou reconheça que tal caminho não existe.

Em [11] foi provado que o problema é  $\mathcal{NP}$ -Completo em grafos direcionados acíclicos.

## 2.3 O Problema do Caminho Mais Longo

O Problema do Caminho Mais Longo (LPP – *Longest Path Problem*) é o problema no qual deseja-se encontrar o caminho de maior custo entre um ponto de origem e um ponto de destino. O custo neste caso pode ser a quantidade de vértices no caminho ou uma função que leva em consideração valores associados às arestas.

De um modo mais formal, o Problema do Caminho Mais Longo pede que, dado um grafo não-direcionado  $G(V, A)$  e um  $K \geq 0$ ,  $K \in \mathbb{N}$ , decida se existe ou não em  $G$  um caminho simples de tamanho  $N$ , tal que  $N \geq K$ . Um caminho simples entende-se como um caminho que não contém ciclos, ou seja, não pode haver repetições de vértices ao longo de todo o caminho.

O LPP é outro problema que, embora resolvido de forma eficiente para algumas poucas classes de grafos, não possui uma solução conhecida de ordem polinomial para o caso geral [21]. Uma prova de que o LPP pertence à classe de problemas  $\mathcal{NP}$ -Completo utiliza a redução do Caminho Hamiltoniano ao mesmo.

Para a prova de tal complexidade do LPP, considere a variante deste problema onde dado um grafo não-direcionado  $G(V, A)$ , dois vértices  $s, t \in V$ , e um inteiro positivo  $K \leq |V|$ , decida se  $G$  possui um caminho simples  $s \rightsquigarrow t$  com uma quantidade  $N \geq K$  de arestas. Agora, o Caminho Hamiltoniano pode ser reduzido ao LPP descrevendo-o da seguinte forma: dado um grafo não-direcionado  $H$  e  $v, w \in V(H)$  dois de seus vértices, decida se há um caminho hamiltoniano  $v \rightsquigarrow w$  entre os dois vértices. Como este último problema é provado ser  $\mathcal{NP}$ -Completo em [12], temos que LPP pertence também à mesma classe de complexidade.

## 2.4 O Problema do Caminho Mínimo com Restrição de Subcaminhos

O Problema do Caminho Mínimo com Restrição de Subcaminhos (SPPFP – *Shortest Path Problem with Forbidden Paths*) é uma variante do Problema do Caminho Mínimo na qual deseja-se encontrar um caminho de menor custo entre uma origem e

todos os destinos possíveis, sem que algum subcaminho proibido seja percorrido durante o processo.

Formalmente, dado um grafo direcionado e aresta valorado  $G(V, A)$ , onde  $V$  e  $A$  são, respectivamente, o conjunto de vértices e o conjunto de arcos pertencentes a  $G$ , dados ainda o vértice  $s \in V$  como vértice origem,  $c : A \rightarrow \mathbb{R}$  como a função de custo dos arcos, e um conjunto de caminhos proibidos  $F = \{u \rightsquigarrow v, w \rightsquigarrow z, \dots\}$ ,  $u, v, w, z \in V$ , encontre um caminho mais curto  $P$  de  $s$  a todos os outros vértices  $t \in V$ , sob a restrição de que nenhum caminho  $f \in F$  seja um subcaminho de  $P$  [22].

Em [20] é provado que o problema em encontrar um caminho mínimo simples com restrição de subcaminhos é  $\mathcal{NP}$ -Completo para determinadas classes de grafos, mesmo considerando caminhos proibidos pequenos, com apenas duas arestas [1]. A prova utiliza uma redução do 3-SAT, problema este apresentado por [12] como um problema  $\mathcal{NP}$ -Completo.

## **Problema do Caminho Mínimo com Restrição de Subcaminhos de Comprimento Dois**

---

O Problema do Caminho Mínimo com Restrição de Subcaminhos foi definido formalmente na Seção 2.4. Este Capítulo aborda o subcaso em que todos os caminhos proibidos possuem comprimento igual a dois, levando em consideração o número de arcos.

A escolha de caminhos proibidos de tamanho igual a dois foi motivada por aplicações na área de modelagem e simulação do tráfego urbano, uma vez que há conversões proibidas que impossibilitam a utilização de uma rota específica. A proibição de uma conversão depende do trecho de via (Rua, Avenida etc) em que a pessoa se encontra, do sentido de sua direção e de qual trecho de via pretente tomar. Como exemplo, suponha que um usuário pretenda seguir o caminho marcado pela seta na figura que ilustra uma parte do mapa do centro da cidade de Goiânia (Figura 3.1). Partindo da Rua Sete, no trecho entre a Rua Quatro e a Avenida Anhanguera, sentido Avenida Anhanguera, o usuário pode fazer a conversão à sua direita pois, segundo o trecho de rua e o sentido especificados, tal conversão é válida (conversão marcada com o número 1 no mapa). Estando na Avenida Anhanguera, trecho entre Rua Sete e Avenida Goiás, sentido Avenida Goiás, o usuário faz uma conversão à sua esquerda (conversão de número 2 no mapa). Mas, apesar de estar no sentido correto do fluxo de veículos da Avenida Goiás, esta última conversão é proibida, não sendo permitido ao usuário realizá-la. Tal conversão é uma instância dos caminhos proibidos que ocorrem no trânsito urbano e que desejamos tratar com os resultados deste projeto.

Dois trechos consecutivos de via devem ser levados em consideração para definir se uma conversão é permitida ou proibida, justificando assim a decisão de abordar um subcaso do SPPFP no qual os caminhos são formados por caminhos menores de tamanho dois.

Nos mapas, a identificação de um determinado local se dá através dos nomes das vias ou de suas intersecções. Na tradução dos mapas para grafos, as identificações são através dos vértices, que são as intersecções das vias em um mapa. Para definir se uma



$u, v, w$ , vértices de um grafo  $G$ , de tal forma que  $(u, v)$ ,  $(v, w)$  sejam arcos de  $G$  e  $(u, v, w)$  não pertença ao conjunto de caminhos proibidos em  $G$ ,  $(u, v, w)$  é uma conversão livre de  $G$ . Na Figura 3.2, o caminho que percorre os arcos  $(d, i)$  e  $(i, h)$  forma a conversão livre  $(d, i, h)$ .

Por fim, definimos “caminho livre” como sendo um caminho de qualquer tamanho no qual nenhum subcaminho proibido pode ser encontrado. Caminhos livres são formados por ligações diretas entre vértices, por uma conversão livre ou por uma sequência de conversões livres. Dados  $u, v, w, x, \dots, p, q, t$  vértices de um grafo  $G$ ,  $(u, v)$ ,  $(v, w)$ ,  $(w, x)$ ,  $\dots$ ,  $(p, q)$ ,  $(q, t)$  arcos de  $G$ , o caminho  $u \rightsquigarrow t$  é um caminho livre se nenhum dos seus subcaminhos  $(u, v, w)$ ,  $(v, w, x)$ ,  $\dots$ ,  $(p, q, t)$  pertence ao conjunto de caminhos proibidos de  $G$ .

Fazendo uso das ideias acima, serão apresentadas quatro propostas de modelagens (Seções 3.1 a 3.4). A intenção em todas elas é, a partir de um grafo original  $G$  do SPPFP, criar um grafo resultante  $G'$  onde não existam transições proibidas e, desta forma, resolver o problema como um SPP. Para cada uma das modelagens é feita a análise do tamanho do grafo resultante, em relação ao grafo original. A análise temporal para a busca de caminhos em  $G'$  não é abordada uma vez que estes problemas devem ser resolvidos como um SPP em  $G'$ , e a complexidade deste problema já é bem conhecida nos dias atuais.

Para melhor compreensão dos trabalhos seguem as notações que serão utilizadas na descrição das modelagens.

- $G(V, A) = G \Rightarrow$  grafo direcionado, aresta valorado, com o conjunto  $V$  de vértices e  $A$  de arcos;
- $V(G) \Rightarrow$  conjunto dos vértices de um grafo  $G$ ;
- $A(G) \Rightarrow$  conjunto dos arcos de um grafo  $G$ ;
- $(u, v) \Rightarrow$  arco entre vértices  $u$  e  $v$ ;
- $(u, v, w) \Rightarrow$  conversão livre e/ou caminho proibido.
- $c(u, v) \Rightarrow$  custo para percorrer o arco  $(u, v)$ ;
- $d^+(u) \Rightarrow$  grau de saída de um vértice  $u$ . O grau de saída é a quantidade de arcos que partem de um determinado vértice;
- $F(G) \Rightarrow$  conjunto dos caminhos proibidos de tamanho igual a dois no grafo  $G$ ;
- $L(G) \Rightarrow$  conjunto dos conversões livres de um grafo  $G$ ;
- $C_2(G) \Rightarrow$  conjunto dos caminhos de tamanho dois no grafo  $G$ .  
 $C_2(G) = F(G) \cup L(G)$ , e  $F(G) \cap L(G) = \emptyset$ ;
- $L_c(G) \Rightarrow$  conjunto das conversões livres consecutivas em um grafo  $G$ . Dados duas conversões livres  $(u, v, w)$  e  $(v, w, z) \in L(G)$  o par  $((u, v, w), (v, w, z)) \in L_c(G)$ ;
- $L(u) \Rightarrow$  conjunto das conversões livres em um grafo  $G$  que iniciam no vértice  $u$ ;

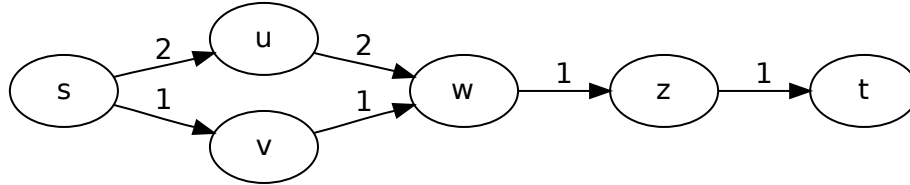


- $||$  (*Módulo*)  $\Rightarrow$  quantidade de elementos de um conjunto. Exemplo:  $|V(G)|$  é a quantidade de vértices de  $G$ .

### 3.1 Modelagem 1

A Modelagem 1 constrói um grafo  $G'$  através do mapeamento de arcos em vértices: cada arco  $(u, v)$  em  $G$  se torna um vértice  $\langle u, v \rangle$  em  $G'$ . Desta forma, cada conversão livre  $\ell = (u, v, w)$  em  $G$  gera um arco  $(\langle u, v \rangle, \langle v, w \rangle)$  em  $G'$ . O custo  $c'$  do arco em  $G'$  é dado por  $c'(\langle u, v \rangle, \langle v, w \rangle) = (c(u, v) + c(v, w))/2$ , que corresponde à metade do custo do caminho  $(u, v, w)$  em  $G$ .

Para exemplificar a descrição da modelagem será utilizado o grafo exibido na Figura 3.3. Nele temos o conjunto de vértices  $V(G) = \{s, u, v, w, z, t\}$ , o conjunto de arcos  $A(G) = \{(s, u), (u, w), (w, z), (z, t), (s, v), (v, w)\}$ , o conjunto de conversões livres  $L(G) = \{(s, u, w), (s, v, w), (u, w, z), (w, z, t)\}$  e a função de custo  $c(s, t)$ . O conjunto de caminhos proibidos  $F(G) = \{(v, w, z)\}$  pode ser encontrado através da eliminação de  $L(G)$  do conjunto que contém todos os caminhos de  $G$  com comprimento dois, uma vez que  $L(G) \cup F(G) = C_2(G)$  e  $L(G) \cap F(G) = \emptyset$ .

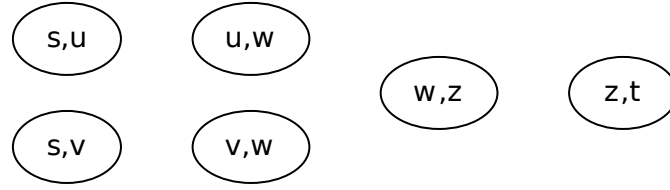


**Figura 3.3:** Grafo original.

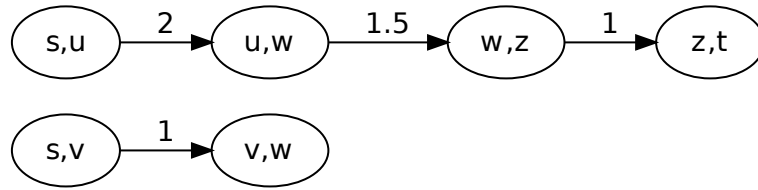
A Figura 3.4 nos mostra como fica o grafo  $G'$  após a geração dos primeiros vértices, e a Figura 3.5 nos mostra como fica o grafo  $G'$  após inclusão dos primeiros arcos.

Todos os vértices originais de  $G$  são também adicionados a  $G'$  (vide Figura 3.6). Enquanto os primeiros vértices adicionados a  $G'$  (aqueles gerados a partir de arcos de  $G$ ) serão partes internas de qualquer caminho, estes últimos vértices adicionados (os vértices *herdados* do grafo inicial  $G$ ) serão a origem e o destino<sup>1</sup> de qualquer caminho em  $G'$ . Um

<sup>1</sup>Dado um caminho  $s \rightsquigarrow t$ , que se inicia em  $s$  e termina em  $t$ , dizemos que  $s$  é o vértice origem (ou vértice inicial) e  $t$  é o vértice destino (ou vértice final) do caminho.



**Figura 3.4:** Mapeamento de arcos em  $G$  para vértices em  $G'$ .

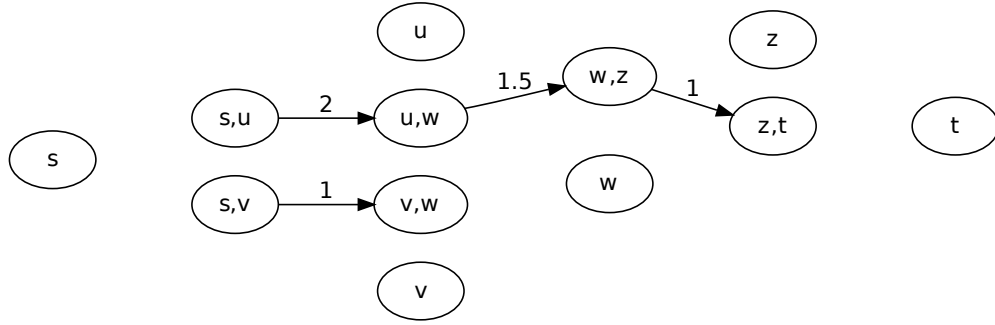


**Figura 3.5:** Inclusão dos primeiros arcos em  $G'$ .

caminho em  $G'$  sempre irá começar em um dos vértices *herdados* de  $G$ , passará por um ou mais vértices gerados a partir dos arcos de  $G$ , e terminará em outro vértice *herdado* de  $G$ . Para que isso seja possível todos os vértices de  $G$  são incluídos em  $G'$  e, dado  $s$  como o vértice inicial de um caminho em  $G$ , todo arco  $(s, v)$  em  $G$  gera um arco  $(s, \langle s, v \rangle)$  em  $G'$ , com custo  $c'$  dado por  $c'(s, \langle s, v \rangle) = c(s, v)/2$ . Isso garante que qualquer caminho em  $G'$  seja iniciado em um vértice “externo” (que possui um vértice correspondente em  $G$ ) e então os vértices “internos” de  $G'$  (os vértices que originaram de arcos em  $G$ ) serão percorridos, garantindo apenas os caminhos permitidos.

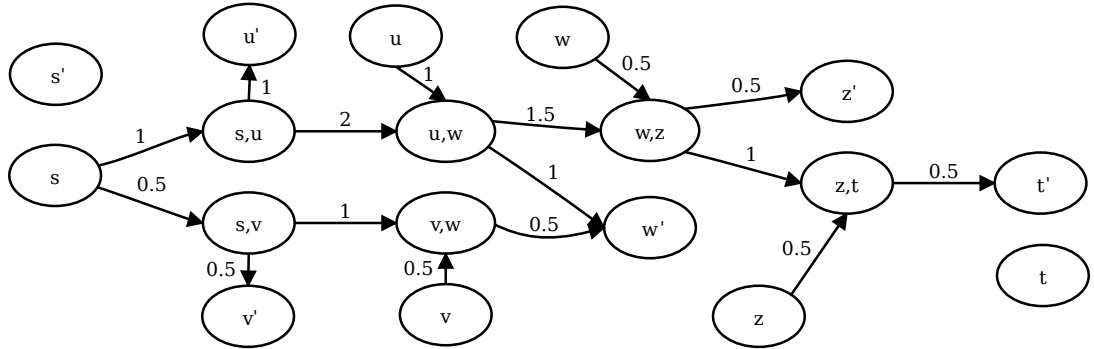
De modo semelhante, para que os vértices destinos possam ser alcançados, todo arco  $(u, v)$  em  $G$  gera também um arco  $(\langle u, v \rangle, v)$  em  $G'$ , com a mesma função de custo  $c'$  definida por  $c'(\langle u, v \rangle, v) = c(u, v)/2$ . Com a inclusão destes arcos os caminhos em  $G'$  podem alcançar os vértices “externos” do mesmo grafo, resultando assim em um caminho em  $G'$  e também, após uma fácil tradução de vértices, em um caminho em  $G$ .

Podemos notar que a modelagem foi definida pressupondo caminhos do tipo  $s \rightsquigarrow x$ , onde  $s$  é o vértice de interesse inicial e  $x$  é todo outro vértice pertencente ao grafo. Contudo, ela pode ser alterada para tratar caminhos entre quaisquer pares de vértices. Isso não é feito de forma direta, pois, para todo e qualquer  $(u, v) \in A$ , a criação de  $(u, \langle u, v \rangle)$  e  $(\langle u, v \rangle, v)$  em  $A'$  gera caminhos adicionais que podem ser proibidos. Uma solução encontrada foi a replicação de todos os vértices  $x$  de  $V'$ , criando os vértices  $x'$  de  $V'$ , de forma que os vértices  $x$  sejam apenas origens de arcos e os vértices  $x'$  sejam apenas



**Figura 3.6:** Inclusão dos vértices “externos” de  $G'$ .

destinos de arcos. Assim, todo e qualquer  $(u, v) \in A$  irá gerar  $(u, \langle u, v \rangle)$  e  $(\langle u, v \rangle, v')$  em  $A'$ , fazendo com que nenhum caminho inicie em  $v'$  nem termine em  $u$ . Isso evita que caminhos adicionais sejam gerados e deixa o grafo resultante preparado para encontrar caminhos entre todos os pares de vértices (Figura 3.7).



**Figura 3.7:** Atribuindo arcos aos vértices “externos” de  $G'$ .

Agora, com o grafo auxiliar já bem definido, precisamos saber como mapear os caminhos de  $G'$  para  $G$ . Uma vez encontrado um caminho  $P'$  em  $G'$ , partindo de  $s$  e terminando em um vértice  $t'$ , é possível construir um caminho  $P$  correspondente em  $G$  eliminando o segundo identificador de um vértice em uma ocorrência consecutiva do identificador em  $P$ . Supondo  $P = \{s, \langle s, u \rangle, \langle u, w \rangle, \langle w, z \rangle, \langle z, t \rangle, t'\}$  do grafo da Figura 3.7, obtemos o caminho correspondente em  $G$  (grafo da Figura 3.3) ao subtrair de  $P$  os identificadores sublinhados em  $\{s, \underline{\langle s, u \rangle}, \underline{\langle u, w \rangle}, \underline{\langle w, z \rangle}, \underline{\langle z, t \rangle}, \underline{t'}\}$ . O caminho em  $G$  é, portanto,  $\{s, u, w, z, t\}$ , e o custo do caminho correspondente em  $G$  é o mesmo custo do caminho em  $G'$ .

### 3.1.1 Análise

Considerando  $|A|$  como sendo a quantidade de arcos em  $G$ ,  $|V|$  sendo a quantidade de vértices em  $G$ ,  $|A'|$  e  $|V'|$  os respectivos de  $G'$ , e  $|L|$  a quantidade de caminhos livres em  $G$ , podemos inferir a complexidade desta modelagem.

Na parte inicial, todos os arcos  $(u, v)$  de  $G(V, A)$  são transformados em vértices  $\langle u, v \rangle$  de  $G'(V', A')$  (vide Figura 3.4). Isso nos leva a  $|V'| = |A|$ . O grafo  $G'$  possui ainda todos os vértices de  $G$ , que servem como origem nos caminhos percorridos em  $G'$  (o que pode ser verificado na Figura 3.6). Completando os chamados vértices “externos”, temos as réplicas dos vértices de  $G$ , que servem como destinos dos caminhos de  $G'$  (vide Figura 3.7). Somando o número de vértices “externos” à quantidade já calculada, temos que:

$$|V'| = 2|V| + |A|$$

Na primeira fase os arcos são criados apenas entre os vértices  $\langle u, v \rangle$  de  $G'$ . A criação de um arco entre os vértices  $\langle u, v \rangle$  e  $\langle v, w \rangle$  significa que o subcaminho  $(u, v, w)$  em  $G$  é uma conversão livre e que  $|A'| \geq |L|$ . Na próxima etapa da modelagem são criados todos os arcos que partem do vértice origem do caminho. Supondo  $s$  tal vértice e  $x$  um vértice qualquer em  $G$ , todos os vértices do tipo  $\langle s, x \rangle$  em  $G'$  recebem um arco de  $s$  em  $G'$ . Com isso adicionamos a  $|A'|$  o grau de saída de  $s$ . Na última etapa são gerados arcos  $(\langle u, v \rangle, v')$  para todos os vértices do tipo  $\langle u, v \rangle$  em  $G'$ . Tais arcos correspondem aos arcos de  $G$  e temos mais  $|A|$  arcos em  $|A'|$ . Assim:

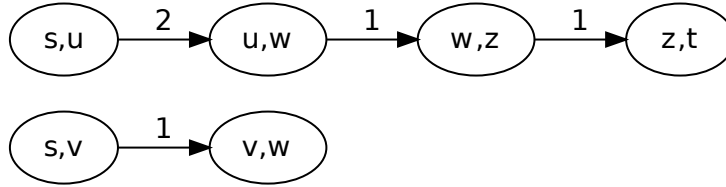
$$|A'| = |L| + d^+(s) + |A|$$

Isso considerando que teremos apenas  $s$  como vértice inicial de qualquer caminho em  $G$ . Em outras palavras, essa análise leva em consideração, até agora, somente os caminhos que iniciam em  $s$ . Se quisermos considerar todos os caminhos em  $G$  que comecem em qualquer vértice e não somente os iniciados em  $s$  – o que é verdade no mundo real –, também as arestas do tipo  $(v, \langle v, w \rangle)$  devem ser criadas para todos os outros vértices  $v \neq s$  pertencentes ao grafo. Tenha em mente que  $v \neq v'$  e que  $v'$  não é origem de nenhum arco. Assim, adicionando-se as arestas agora citadas com as arestas do vértice  $s$  já presentes em  $G'$ , temos que todas as arestas que saem de algum vértice de  $G$  são agora acrescentadas também em  $G'$ . Sabendo que  $d^+(s) + \sum(d^+(v)) \forall v \in \{V(G) - \{s\}\} = |A|$  temos agora que:

$$|A'| = |L| + 2|A|$$

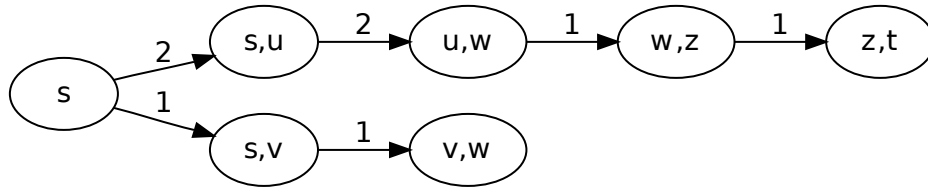
## 3.2 Modelagem 2

A Modelagem 2 surge a partir de uma modificação da Modelagem 1, na qual não são incluídos em  $G'$  os vértices de  $G$  diferentes do vértice inicial  $s$ . O grafo  $G'$  também é construído através do mapeamento de arcos em vértices, onde cada arco  $(u, v)$  em  $G$  se torna um vértice  $\langle u, v \rangle$  em  $G'$ , exatamente como na Modelagem 1 (vide Figura 3.4). Cada conversão livre  $\ell = (u, v, w)$  em  $G$  gera um arco entre os vértices  $\langle u, v \rangle$  e  $\langle v, w \rangle$  em  $G'$ . O custo  $c'$  do arco em  $G'$  sofre uma pequena alteração e é dado por  $c'(\langle u, v \rangle, \langle v, w \rangle) = c(v, w)$ , levando em consideração apenas o custo do arco de  $v$  a  $w$ , último arco do caminho  $u, v, w$  em  $G$  (Figura 3.8).



**Figura 3.8:** Grafo  $G'$  após inclusão dos primeiros vértices e dos primeiros arcos.

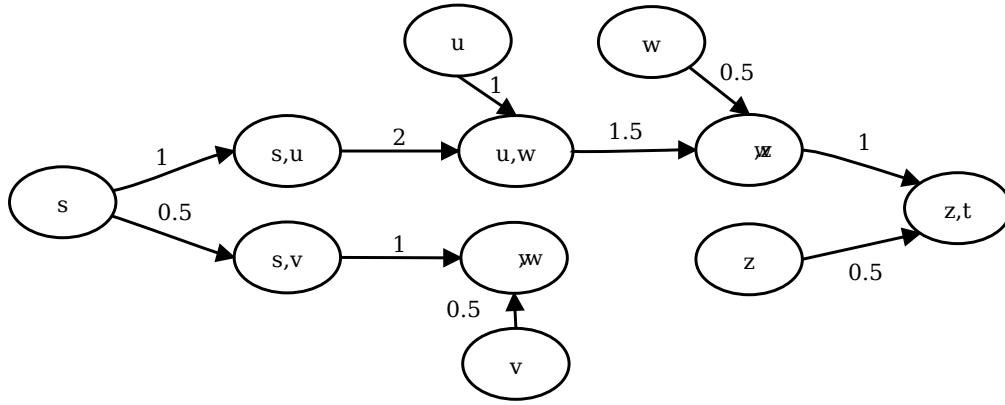
Diferentemente da Modelagem 1, onde todos os vértices de  $G$  se fazem presentes em  $G'$ , nesta Modelagem apenas o vértice origem  $s$ , dos vértices de  $G$ , é adicionado a  $G'$  (Figura 3.9). Ele será a origem de caminhos em  $G'$  e os vértices destino serão vértices do tipo  $\langle u, v \rangle$ , que representam os arcos de  $G$ . Cada arco  $(s, v)$  em  $G$  gera ainda um arco  $(s, \langle s, v \rangle)$  em  $G'$  com custo  $c'(s, \langle s, v \rangle) = c(s, v)$  (Figura 3.9).



**Figura 3.9:** Grafo  $G'$  após inclusão do vértice origem e seus arcos.

Até o momento, a modelagem foi definida pressupondo caminhos do tipo  $s \rightsquigarrow x$ , onde  $s$  é o vértice de interesse inicial e  $x$  é todo outro vértice pertencente ao grafo. Para tratar caminhos entre quaisquer pares de vértices, todo vértice  $v \in V$  que é origem de pelo menos um arco deve ser inserido em  $G'$  e seus arcos  $(v, x) \in A$  devem gerar arcos  $(v, \langle v, x \rangle)$

em  $A'$ . Note, consultando a Figura 3.10, que, devido a essa regra para inserção de novos vértices, o vértice  $t$  não aparece em  $G'$  por não ser origem de nenhum arco e, portanto, não iniciar nenhum caminho.



**Figura 3.10:** Grafo  $G'$  com todos os vértices e arcos.

Finalizando a descrição desta modelagem, o processo de encontrar um caminho  $P$  em  $G$ , correspondente a um caminho  $P'$  já encontrado em  $G'$ , é semelhante ao processo realizado na Modelagem 1, com a diferença de que o vértice destino não é um vértice “herdado” de  $G$ , mas sim um vértice  $\langle u, t \rangle$  referente a um arco em  $G$ . O rótulo do vértice destino de  $G'$  conterá o rótulo do vértice destino de  $G$  como um sufixo. Como pode existir mais que um vértice de  $G'$  com o sufixo do vértice destino  $t$ , é necessário selecionar o vértice  $\langle u, t \rangle$ ,  $\forall u \in V$ , que resulta no menor caminho para  $t$ .

### 3.2.1 Análise

A grande maioria dos vértices do grafo  $G'(V', A')$  resultante é formada a partir dos arcos do grafo original  $G(V, A)$ . Seguindo a nomenclatura utilizada na Modelagem 1, esses vértices são os chamados vértices “internos” de  $G'$  e compõem quase que a totalidade dos vértices dos caminhos em  $G'$ , excluindo-se apenas os vértices origem e destino que são vértices “externos”. Mas, nesta Modelagem, os vértices destino também são “internos” – ou do tipo  $\langle v, t \rangle$  – o que diminui a quantidade de vértices no grafo final. Mesmo assim ainda temos que  $|V'| \geq |A|$ . Adicionando a essa conta apenas uma unidade, referente ao vértice  $s$  origem, temos:

$$|V'| = |A| + 1$$

Mas, da mesma forma que ocorre na Modelagem anterior, essa conta leva em consideração apenas o vértice inicial  $s$  do caminho. Para que seja possível iniciar um

caminho de qualquer vértice deve-se adicionar a  $G'$ , inicialmente, todos os vértices de  $G$ . Isso nos leva a:

$$|V'| = |A| + |V|$$

Entretanto uma análise mais cuidadosa pode trazer uma redução desses números baseando-se no fato de que há grafos que possuem vértices com  $d^+(v) = 0$ , ou seja, vértices que não são origem de nenhum arco. Sendo assim tais vértices seriam apenas vértices finais de qualquer caminho neste grafo e sua inclusão no grafo auxiliar não se mostra necessária. Não incluindo esses vértices no grafo auxiliar temos:

$$|V'| \leq |A| + |V|$$

ou, de forma mais específica,

$$|V'| = |A| + \sum \delta(v)$$

$$\text{tal que, } \forall v \in G(V), \delta(v) = \begin{cases} 1 & \text{se } d^+(v) > 0, \\ 0 & \text{se } d^+(v) = 0. \end{cases}$$

Partindo para análise dos arcos de  $G'$ , os primeiros arcos criados são os que unem os vértices do tipo  $\langle u, v \rangle$  de  $G'$ . A ligação de dois vértices  $\langle u, v \rangle$  e  $\langle v, w \rangle$  de  $G'$  corresponde à formação de uma conversão livre  $\langle u, v, w \rangle$  no grafo original  $G$ . Isso garante que  $|A'| \geq |L|$ . Os últimos arcos criados são os que partem do vértice origem  $s$ . Todo arco  $(s, v)$  de  $G$  gera um arco  $(s, \langle s, v \rangle)$  em  $G'$  e são os primeiros arcos a serem percorridos em qualquer caminho do grafo final iniciado em  $s$ . Adicionando essa quantidade de vértices que partem de  $s$  em  $G$  temos:

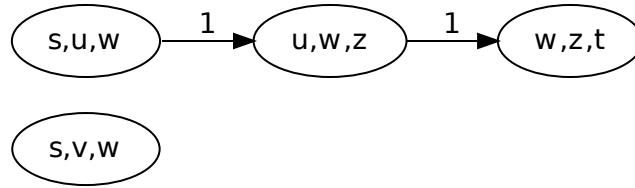
$$|A'| = |L| + d^+(s)$$

Como pode-se notar, a análise acima leva em consideração apenas caminhos que iniciam no vértice  $s$ . Para que sejam formados caminhos que iniciem em qualquer vértice de  $G$  devemos adicionar a  $G'$  todos os arcos de  $v \in V(G)$  da mesma forma que os arcos de  $s$  foram adicionados. Somando os arcos adicionados temos  $d^+(s) + \sum d^+(v) \forall v \in V(G) - \{s\} \Rightarrow \sum d^+(v) \forall v \in V(G)$  que, na verdade, é o conjunto completo de arcos no grafo original. Assim, considerando caminhos que iniciam em qualquer vértice de  $G$ , temos:

$$|A'| = |L| + |A|$$

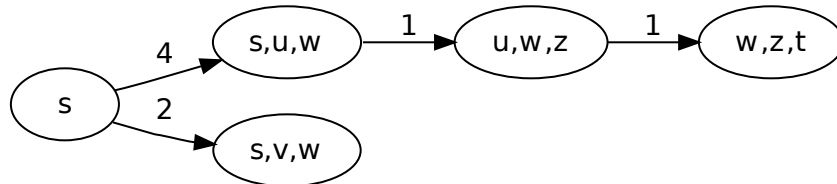
### 3.3 Modelagem 3

Diferentemente do proposto nas Modelagens anteriores, na Modelagem 3, oriunda de [13], os vértices iniciais serão formados a partir das próprias conversões livres de  $G$ . Assim, toda conversão livre  $(u, v, w)$  pertencente a  $L(G)$  gerará um vértice  $\langle u, v, w \rangle$  em  $G'$  e basta a informação contida no próprio vértice para saber qual conversão livre ele representa – o que não acontece nas outras Modelagens. Cada par de conversões livres  $(u, v, w)$  e  $(v, w, z)$  de  $L(G)$  gera um arco  $(\langle u, v, w \rangle, \langle v, w, z \rangle)$  em  $G'$ , e o seu custo  $c'$  é dado por  $c'(\langle u, v, w \rangle, \langle v, w, z \rangle) = c(w, z)$ , o que, em  $G$ , seria dizer que o custo é equivalente ao custo do arco entre os dois últimos vértices da última conversão livre. Tomando o grafo da Figura 3.3 como grafo original e aplicando estes passos sobre o mesmo, o grafo da Figura 3.11 exibe como ficaria o grafo resultante após esta etapa inicial da modelagem.



**Figura 3.11:** Criação dos vértices iniciais e custos de seus arcos.

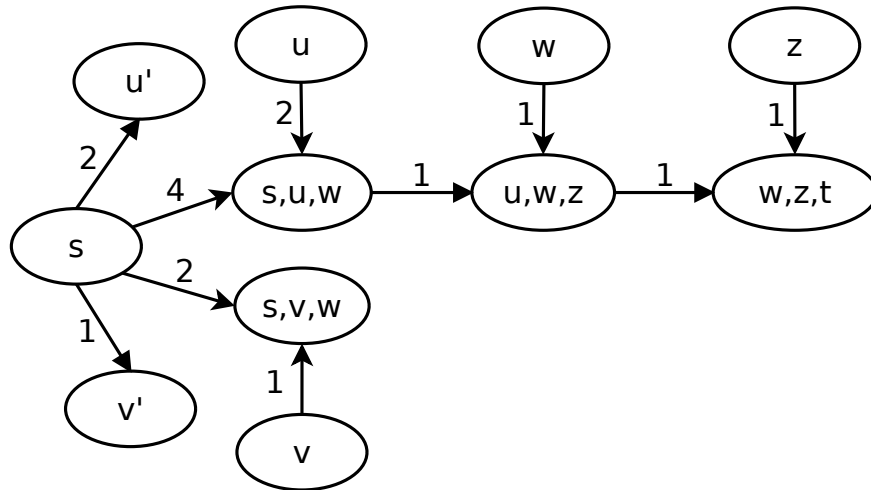
Todo vértice  $s$  de  $G$  que é origem de arcos mas não é destino de nenhum também pertence a  $G'$  e é gerado um arco  $(s, \langle s, u, v \rangle)$  para cada conversão livre  $(s, u, v)$  em  $G$  na qual  $s$  é o vértice inicial, com custo  $c'$  definido por  $c'(s, \langle s, u, v \rangle) = c(s, u) + c(u, v)$  (vide exemplo na Figura 3.12).



**Figura 3.12:** Adicionando o vértice origem e custo de suas arestas.



Para inserção dos outros vértices é considerado o caso no qual o caminho em  $G$  possui tamanho menor que dois. Devido ao fato de os vértices  $\langle u, v, w \rangle$  em  $G'$  serem formados a partir de conversões livres de  $G$ , os caminhos em  $G$  menores que o tamanho de uma conversão livre – ou seja, caminhos menores que dois – não estariam presentes em  $G'$  caso todos os vértices de  $G$  fossem criados no grafo auxiliar conforme descrito no parágrafo anterior. Para resolver esta questão é necessário, para todo  $v \in V$  que é origem e destino de arcos, inserir  $v$  em  $G'$  e criar arcos  $(v, \langle u, v, w \rangle)$  para todo vértice de  $G'$  que possui  $v$  como identificador central, atribuindo seu custo como  $c'(v, \langle u, v, w \rangle) = c(v, w)$ . Para os vértices  $v$  de  $G$  que não são vértices centrais de nenhuma conversão livre, é necessário criar vértices auxiliares  $w'$  de seus vizinhos e atribuir arcos que os unam, sendo a função de custo  $c'(v, w') = c(v, w)$ . Este é o caso do vértice  $s$  de nosso grafo original de exemplo, e o grafo  $G'$  final é visto na Figura 3.13.



**Figura 3.13:** Grafo auxiliar final.

Para encontrar um caminho não trivial <sup>2</sup>  $P' = s \rightsquigarrow t$  em um grafo  $G'$  já construído, basta iniciar a busca a partir do vértice  $s$  e percorrer o grafo até encontrar o vértice na forma  $\langle u, v, t \rangle$  que possui  $t$  como sufixo. Utilizemos o grafo da Figura 3.13 para encontrar um caminho de  $s$  a  $t$ , ambos pertencentes a  $G$ . Iniciando no vértice  $s$  escolhemos a aresta que nos leva ao vértice  $\langle s, u, w \rangle$ , e deste escolhemos a aresta que nos leva ao vértice  $\langle u, w, z \rangle$ . A próxima aresta escolhida nos leva ao vértice  $\langle w, z, t \rangle$  onde podemos observar o sufixo  $t$ . Como  $t$  é o vértice final do caminho em  $G$  e é o sufixo do vértice  $\langle w, z, t \rangle$  em  $G'$  constatamos que este é o vértice destino em  $G'$  e o caminho está construído neste grafo.

<sup>2</sup>Para este caso, consideramos um caminho trivial  $u \rightsquigarrow v$  em  $G'$  aquele formado apenas por um arco do tipo  $(u, v')$ .

Dado um caminho  $P'$  em  $G'$ , partindo de  $s$  e terminando em um vértice  $\langle u, v, t \rangle$ , é possível construir um caminho correspondente  $P$  em  $G$  que inicia em  $s$  e termina em  $t$ , mas essa operação é um pouco mais complicada que a da Modelagem 1. Na Modelagem 1 os vértices origem e destino de qualquer caminho em  $G$  e  $G'$  são tanto vértices de  $G$  quanto de  $G'$ , facilitando a construção de caminhos correspondentes de um grafo para o outro. Isso não acontece nesta Modelagem 3, pois os vértices destino em  $G'$  são do tipo  $\langle u, v, t \rangle$ , enquanto os vértices destino em  $G$  são vértices originais, do tipo  $t$ .

Desta forma, para obter  $P$  de  $P'$ , devemos considerar, de  $P'$ , o seu vértice inicial, o(s) índice(s) à direita do identificador do vértice inicial no primeiro vértice do tipo  $\langle u, v, w \rangle$  e apenas o último índice de todos os outros vértices do caminho. Seguindo o grafo da Figura 3.13 e tomando o caminho  $P'$  como  $s, \langle s, u, w \rangle, \langle u, w, z \rangle, \langle w, z, t \rangle$ , o caminho  $P$  em  $G$  é formado pelo vértice inicial  $s$ , seguido por  $u$  e  $w$  que são os dois últimos índices do primeiro vértice diferente da origem. O próximo vértice do caminho em  $G$  é  $z$ , pois é o sufixo de  $\langle u, w, z \rangle$ , seguido por  $t$ , sufixo de  $\langle w, z, t \rangle$ . Assim, o caminho correspondente a  $P'$  em  $G$  é  $s, u, w, z, t$  e pode ser verificado no grafo da Figura 3.3.

### 3.3.1 Análise

É simples iniciar o cálculo da quantidade de vértices  $|V'|$  de  $G'$ , uma vez que a maioria dos vértices gerados são procedentes do conjunto  $L$  de conversões livres de  $G$ , adicionando a eles o vértice inicial  $s$  do caminho. Assim, temos que:

$$|V'| \geq |L| + 1$$

Mas como, geralmente, não se tem apenas um possível vértice inicial em um grafo  $G$ , esse número deve ser estendido, adicionando um vértice em  $G'$  para todos os vértices em  $G$  que são origem de pelo menos um arco. A inclusão de tais vértices nos leva a:

$$|V'| = |L| + \sum \delta(v)$$

$$\text{tal que, } \forall v \in G(V), \delta(v) = \begin{cases} 1 & \text{se } d^+ > 0, \\ 0 & \text{se } d^+ = 0. \end{cases}$$

Continuando cálculo, temos que incluir os vértices auxiliares para aqueles que não são vértices centrais de nenhuma conversão livre, caso onde o caminho mínimo é formado apenas por um arco. Assim, chegamos a:

$$|V'| = |L| + \sum \delta(v) + \sum \theta(v)$$

$$\text{tal que, } \forall v \in G(V), \theta(v) = \begin{cases} 0 & \text{se } \exists(u, v, w) \in C, \\ d^+(v) & \text{se } \nexists(u, v, w) \in C. \end{cases}$$

Essa conta pode ser simplificada se consideramos que o grafo  $G$  a ser analisado é um mapeamento de uma rede de trânsito urbana, no qual todos os vértices são tanto origens de arcos quanto elementos centrais em conversões livres. Devido à essas características, a quantidade de vértices para este tipo de grafo é obtida, simplesmente, através do seguinte cálculo:

$$|V'| = |L| + |V|$$

A quantidade de arcos também não é tão simples de ser encontrada, pois ela depende diretamente da quantidade de pares de conversões livres consecutivas, no formato  $(u, v, w)$  e  $(v, w, z)$ , e tal quantidade não pode ser calculada de um modo genérico como feito nas modelagens anteriores. Adicionalmente devem ser levados em consideração também os arcos que unem os vértices iniciais às conversões livres e os arcos dos vértices iniciais que não são elementos centrais de nenhuma conversão livre. Assim:

$$|A'| = |L_c| + |L| + \sum \delta(v)$$

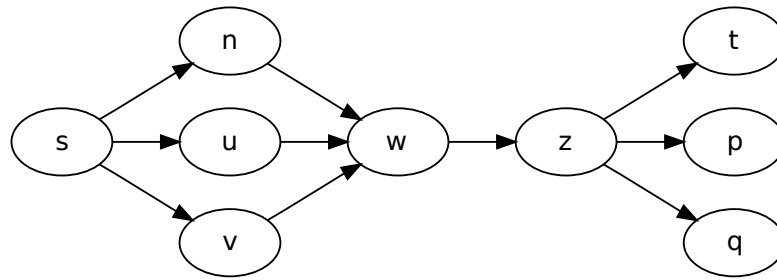
$$\text{tal que, } \forall v \in G(V), \delta(v) = \begin{cases} 0 & \text{se } \exists(u, v, w) \in L(G), \\ d^+(v) + |L(v)| & \text{se } \nexists(u, v, w) \in L(G). \end{cases}$$

Se considerarmos, novamente, que o grafo  $G$  a ser analisado é um mapeamento de uma rede de trânsito urbana, o último fator do cálculo anterior terá sempre o valor zero, uma vez que, neste tipo de grafo, todo vértice é elemento central de ao menos uma conversão livre, ou seja,  $\forall v \in V, \exists(u, v, w) \in L$ . Assim, temos:

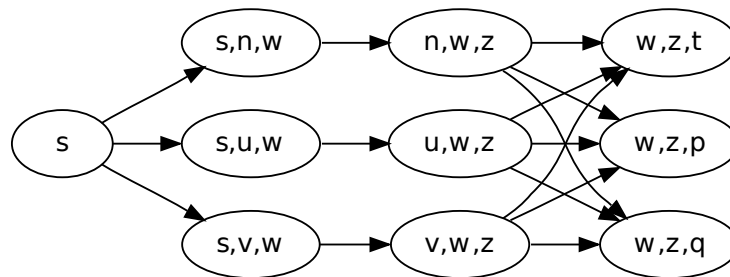
$$|A'| = |L_c| + |L|$$

Essa dependência de classes específicas de conversões livres pode resultar em uma quantidade de arcos de  $G'$  menor que a quantidade de arcos de  $G$ . Esse é o caso do

grafo da Figura 3.3, onde  $|A'| = 4$  e  $|A| = 6$ . Verifique esse fato comparando aquele grafo com o da Figura 3.12. Mas em outros casos,  $|A'|$  pode ser maior que  $|A|$ , como no grafo  $H$  da Figura 3.14, formado apenas por caminhos livres, e no seu grafo  $H'$  resultante<sup>3</sup>, na Figura 3.15, onde  $|A(H)| = 10$  e  $|A(H')| = 15$ .



**Figura 3.14:** Grafo  $H$ .



**Figura 3.15:** Grafo  $H'$ , com uma quantidade maior de arcos que  $H$ .

### 3.4 Modelagem 4

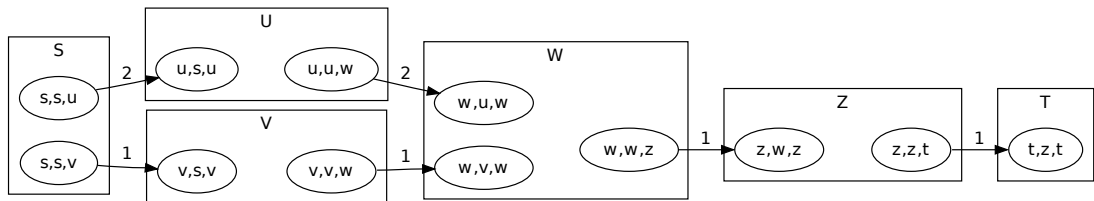
A última modelagem apresentada neste projeto mostra uma abordagem diferente das modelagens anteriores, e que surgiu de [10]. Nestas eram feitas transformações de alguns arcos do grafo original em vértices no grafo resultante. Nas duas primeiras modelagens eram criados vértices  $\langle u, v \rangle$  em  $G'$  representando arcos  $(u, v)$  de  $G$ , e na terceira modelagem eram criados vértices  $\langle u, v, w \rangle$  em  $G'$  representando conversões livres  $(u, v, w)$

<sup>3</sup>Este grafo resultante apresenta apenas  $s$  como vértice inicial a fim de facilitar a visualização.

em  $G$ . Diferentemente destas três modelagens anteriores, onde eram feitas transformações de arcos de  $G$  para vértices de  $G'$ , a modelagem desta Seção estende o grafo original a fim de separar cada possível fluxo nele encontrado. A forma encontrada para que cada fluxo seja facilmente identificada foi dividir os vértices, na proporção de um vértice em  $G'$  para cada destino de arco nos vértices de  $G$ . O mesmo foi feito com os vértices que são origem de arcos. Assim sendo, cada arco em  $G$  criará um arco e dois vértices em  $G'$ , um em cada extremidade. Chamemos estes arcos de arcos inter-vértices, pois unem dois vértices distintos de  $G$ .

São utilizados três campos para dar nome a um vértice em  $G'$ , de forma semelhante ao que é apresentado na Modelagem 3. O primeiro campo identifica qual vértice de  $G$  o vértice de  $G'$  representa. O segundo campo indica de qual vértice de  $G$  partiu o arco associado ao vértice de  $G'$ , e da mesma forma o terceiro campo indica o destino do arco. Complementando, o custo associado a cada arco de  $G$  adicionado a  $G'$  nesta etapa se mantém o mesmo. Como exemplo, dados  $u, z \in V(G)$  e  $(u, z) \in A(G)$  será gerado  $\langle u, u, z \rangle, \langle z, u, z \rangle \in V(G')$  e um arco  $(\langle u, u, z \rangle, \langle z, u, z \rangle) \in A(G')$  unindo os dois vértices com custo  $c'(\langle u, u, z \rangle, \langle z, u, z \rangle) = c(u, z)$ . Embora um pouco confuso à primeira vista, esta forma de nomear os vértices facilita a compreensão da transformação de  $G$  em  $G'$ , e nada impede que em uma implementação seja feito um mapeamento para nomes mais fáceis de serem tratados.

Levando em consideração o grafo da Figura 3.3 o início do grafo  $G'$  se dá como visto na Figura 3.16. Esta primeira figura agrupa os vértices de  $G'$  criados a partir de um mesmo vértice de  $G$  para facilitar o entendimento de como foram criados esses vértices e da nomenclatura dos mesmos.

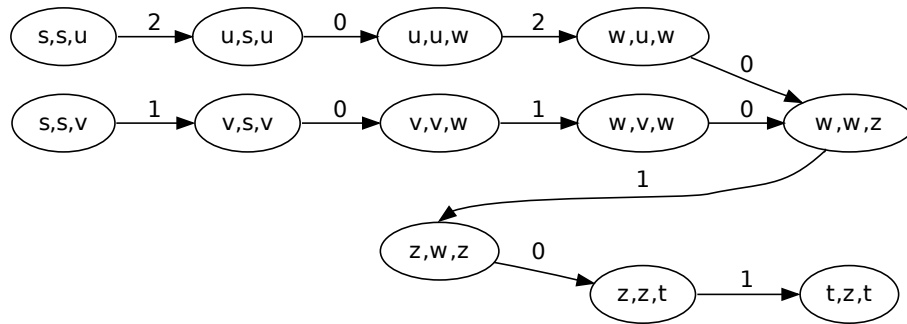


**Figura 3.16:** Criação dos vértices e dos arcos inter-vértices.

Com os vértices e os arcos inter-vértices criados incluímos os arcos que representam as “ligações internas” de um mesmo vértice de  $G$ , e chamemos estes arcos de intra-vértices. A inclusão de um destes arcos em  $G'$  permite a ligação entre três vértices consecutivos de  $G$  (do vértice anterior, do vértice atual e do próximo vértice). Em outras palavras, significa que aquele caminho é uma conversão livre em  $G$ . Portanto para criarmos uma conversão livre basta utilizar um arco intra-vértice do segundo vértice da

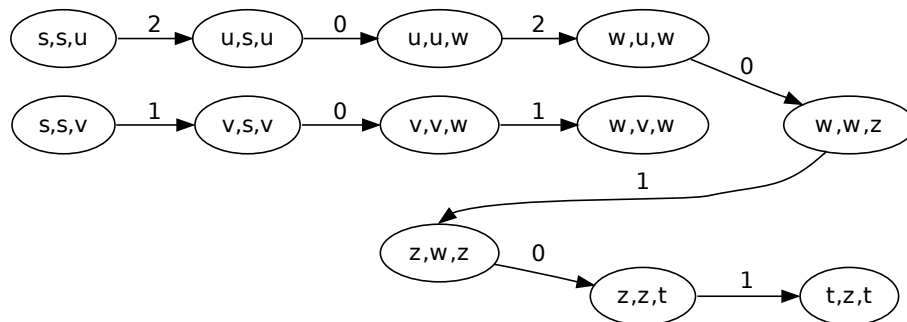
conversão livre, e para inibir que um caminho proibido seja percorrido basta eliminar o arco intra-vértice correspondente no segundo vértice de tal caminho.

Como não há nenhum ônus quando se entra ou sai de um vértice em  $G$ , os arcos intra-vértices em  $G'$  não possuem custo, podendo ser percorridos sem o consumo de valor algum. O grafo da Figura 3.17 apresenta o grafo  $G'$  com todos os vértices e arcos. É representado de uma forma diferente da Figura 3.16 a fim de melhor se ajustar aos limites da página e facilitar a sua visualização.



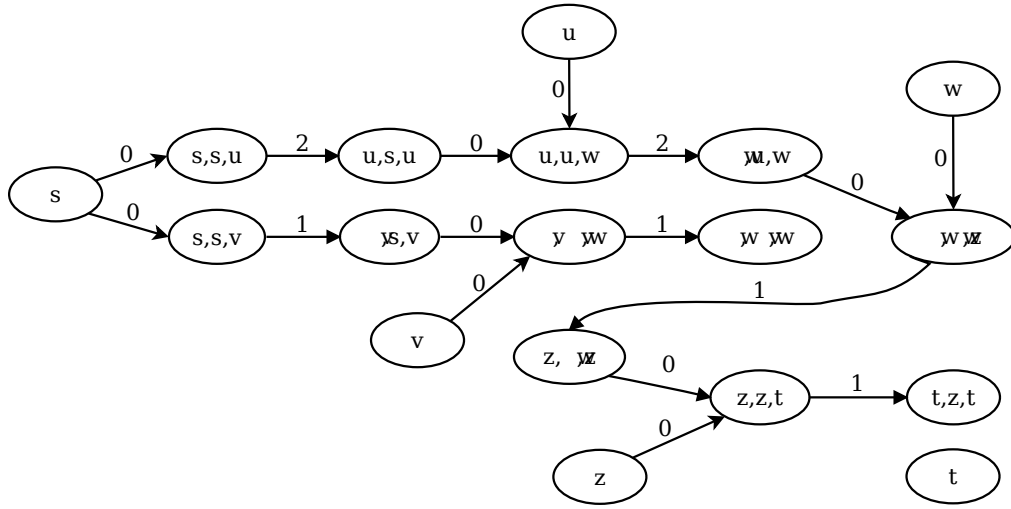
**Figura 3.17:** Grafo  $G'$  completo.

No entanto, o grafo anterior ainda possui caminhos proibidos de tamanho igual a dois. A remoção destes caminhos é feita de uma forma mais prática que nas abordagens anteriores, bastando apenas remover o arco intra-vértice, em  $G'$ , do vértice correspondente ao segundo vértice do caminho em  $G$ . No grafo original tomado como exemplo em todas as modelagens, o conjunto de caminhos proibidos é  $F(G) = \{(v, w, z)\}$ . Para eliminar este caminho proibido de  $G'$  basta remover o arco intra-vértice de  $w$  que une os vértices  $v, w$  e  $z$  de  $G$ , ou seja, remover  $(\langle w, v, w \rangle, \langle w, w, z \rangle)$  de  $A(G')$ . A Figura 3.18 mostra o grafo  $G'$  já sem a presença de caminhos proibidos.



**Figura 3.18:** Grafo  $G'$  já sem caminhos proibidos.

Agora resta apenas a inclusão dos vértices iniciais, que serão origens de qualquer caminho em  $G'$ . Para isso, cada vértice  $v$  de  $G$  gera um vértice  $v$  em  $G'$  e, para cada um dos vértices adicionados, deve-se criar arcos  $(v, \langle v, v, w \rangle)$  para todo vértice  $\langle v, v, w \rangle$  de  $V'$ . Ou seja, inserimos os vértices iniciais em  $G'$  e criamos arcos unindo estes novos vértices aos vértices que são origem de arcos inter-vértices de  $G'$ . O custo dos novos arcos é zero e o grafo final, que permite caminhos entre qualquer par de vértices, é visto na Figura 3.19.



**Figura 3.19:** Grafo  $G'$  final.

Uma vez encontrado um caminho mínimo  $P'$  em  $G'$  partindo de  $s$  e chegando a  $t$ , a construção de um caminho  $P$  correspondente em  $G$  se dá através da seleção dos prefixos dos nomes dos vértices de  $P'$ , de forma que os prefixos que aparecem duas vezes consecutivas em  $P'$  sejam selecionados apenas uma vez. Desta forma, dado o caminho mínimo  $P' = \{s, \langle s, s, u \rangle, \langle u, s, u \rangle, \langle u, u, w \rangle, \langle w, u, w \rangle, \langle w, w, z \rangle, \langle z, w, z \rangle, \langle z, z, t \rangle, \langle t, z, t \rangle\}$ , os prefixos selecionados segundo o critério mencionado anteriormente são os sublinhados em  $\{\underline{s}, \langle s, s, u \rangle, \langle \underline{u}, s, u \rangle, \langle u, u, w \rangle, \langle w, u, w \rangle, \langle w, w, z \rangle, \langle \underline{z}, w, z \rangle, \langle z, z, t \rangle, \langle \underline{t}, z, t \rangle\}$ , o que nos leva ao caminho  $P = \{s, u, w, z, t\}$  em  $G$ .

### 3.4.1 Análise

A quantidade de vértices do grafo  $G'$  resultante, como descrito no início desta Seção 3.4, é obtida através do número de arcos e de vértices em  $G$ . Cada arco neste grafo gera um par de vértices em  $G'$  e cada vértice de  $G$  é repetido em  $G'$  para serem os vértices iniciais, o que nos leva a inferir a quantidade de vértices do grafo resultante como sendo:

$$|V'| = |V| + 2|A|$$

Levando em consideração que existe um arco unindo cada par de vértices criados em  $G'$  a partir dos arcos de  $G$  sabemos, inicialmente, que a quantidade de arcos em  $G'$  será  $|A'| \geq |A|$ . Outros arcos incluídos em  $G'$  são os arcos intra-vértices, que, uma vez criados, simbolizam as conversões livres em  $G$ . Por último, temos os arcos que partem dos vértices iniciais. Assim chegamos a:

$$|A'| = 2|A| + |L|$$



## Conclusão

---

Foram descritas neste documento abordagens que apresentam soluções para o SPPFP. Algumas tratam do caso geral do problema e facilmente podem ser utilizadas para a resolução do subcaso aqui apresentado, algumas vezes necessitando de leves alterações em sua descrição. Outras foram modeladas especificamente para tratar caminhos proibidos de comprimento dois, sendo que seu uso para o caso geral – que possui também caminhos proibidos de comprimento maior que dois – geralmente não resulta em soluções aceitáveis.

Modelagens projetadas para o caso geral procuram encontrar soluções que aumentem o mínimo possível o tamanho do grafo resultante. Em [22], por exemplo, os caminhos proibidos são evitados através da utilização da abordagem de desvio de caminho de Martins. Esta abordagem de desvio de caminho acrescenta dois vértices e pelo menos um arco no grafo, se o caminho proibido tiver um comprimento maior que dois, e, no caso especial onde o caminho proibido possui tamanho exatamente igual a dois, a quantidade de vértices é incrementada em uma unidade e a quantidade de arcos pode tanto permanecer inalterada quanto ser incrementada em várias unidades. Esta modelagem resolve o problema e apresenta bons resultados para caminhos proibidos de qualquer tamanho. Adicionalmente, permite que o grafo seja alterado em tempo de execução, tornando possível a adição de novos vértices a qualquer momento. Mas como, neste projeto, será utilizada apenas para tratar caminhos proibidos de tamanho dois, traz a desvantagem de sempre aumentar o tamanho do grafo toda vez que este é modificado para evitar que um caminho proibido de tamanho dois seja percorrido. Além disso, a replicação de um vértice para eliminação de um caminho proibido pode duplicar a ocorrência de outros caminhos proibidos que passam pelo mesmo vértice, fazendo com que um único caminho proibido no grafo original apareça várias vezes no grafo resultante.

As modelagens criadas especificamente para o subcaso do SPPFP, onde todos os caminhos proibidos são de tamanho igual a dois, são projetadas pensando em um crescimento ainda menor do grafo resultante, sendo que, em alguns casos, este grafo é menor que o grafo original. A Modelagem 1 (Seção 3.1) e a Modelagem 2 (Seção 3.2) são bem semelhantes, com a principal diferença de que a segunda é um melhoramento da primeira,

<i>Espaços</i>	$ V' $	$ A' $
<b>Modelagem 1</b>	$2 V  +  A $	$2 A  +  L $
<b>Modelagem 2</b>	$ V  +  A $ <sup>1</sup>	$ A  +  L $
<b>Modelagem 3</b>	$ V  +  L $ <sup>1</sup>	$ Lc  +  L $
<b>Modelagem 4</b>	$ V  + 2 A $	$2 A  +  L $

<sup>1</sup> No pior caso.

**Tabela 4.1:** Complexidades espaciais das modelagens.

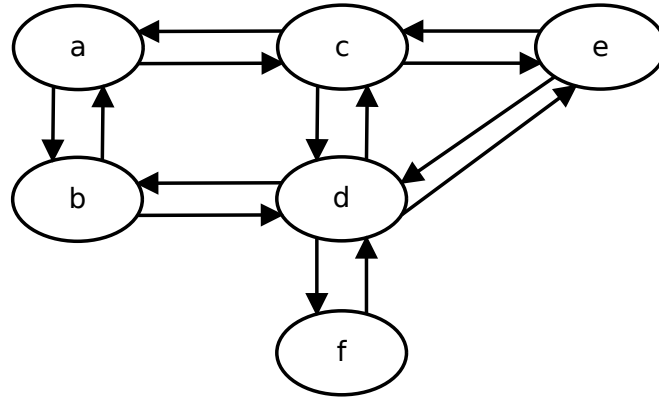
contendo um número menor de vértices e arcos (utilize a Tabela 4.1 para consulta <sup>1</sup>). A primeira apresenta uma discreta vantagem na visualização de seus caminhos por utilizar os vértices do grafo original como vértices origem e destino de seus caminhos. A segunda apresenta-se mais viável de utilização no subcaso do que a primeira, pois possui menor quantidade de vértices e arcos no grafo resultante, o que implica diretamente em um menor tempo de execução. A Modelagem 3 (Seção 3.3) é diferente das duas anteriores por transformar as conversões livres do grafo original em vértices no grafo resultante. Contudo, este grafo apresenta tamanho semelhante, de um modo geral, ao da Modelagem 2, significando que o tempo necessário para encontrar um caminho mínimo em cada grafo resultante pode ser bem parecido. Vale ressaltar que, em alguns casos específicos, é percebida uma diferença maior no tamanho final dos grafos – como no exemplo dos grafos das Figuras 3.14 e 3.15 – exigindo uma maior atenção na comparação dos grafos finais. A Modelagem 4 (Seção 3.4), última apresentada neste projeto, é a que, aparentemente, apresenta um número maior de vértices no grafo resultante. Mas, mesmo assim, a quantidade de arestas não é tão elevada, comparando-se com a da Modelagem 2. Um fator diferencial, visto apenas nesta Modelagem 4, é a facilidade com que se exclui e inclui um caminho proibido no grafo final: basta apenas inserir ou eliminar um arco intra-vértice do grafo final, sem qualquer outra alteração. Em todas as outras modelagens seria necessário realizar alterações também no conjunto de vértices.

## 4.1 Comparações com Resultados de um Caso de Teste

As conclusões expostas acima nos dão uma ideia geral das diferenças das modelagens, mas não nos traz números com os quais podemos selecionar a abordagem mais adequada para nossas necessidades. Esta Seção apresenta os números obtidos nos testes e que irão auxiliar na escolha.

<sup>1</sup> A Tabela 4.1 foi montada utilizando as notações apresentadas no Capítulo 3, página 22.

O grafo tido como base é o apresentado na Figura 4.1. Ele apresenta características encontradas nas malhas de trânsito urbanas, como ciclos com quatro vértices  $(a, b, d, c, a)$ , três vértices  $(c, d, e, c)$  e dois vértices  $(d, f, d)$ , além de combinações entre eles. Ainda pensando como uma malha urbana, o conjunto  $F$  de caminhos proibidos é formado pelas conversões à esquerda que possuem  $d$  como vértice central  $(\{(f, d, b), (b, d, c), (c, d, e), (e, d, f)\})$  e pelos retornos, exceto  $(d, f, d)$ .



**Figura 4.1:** Grafo  $G$  utilizado para comparações entre as modelagens.

Com o grafo  $G(V,A)$  definido, bem como o conjunto  $F$  de caminhos proibidos, foram gerados os grafos  $G'$ s utilizando as quatro modelagens do Capítulo 3. A Tabela 4.2 mostra as quantidades de vértices e de arcos obtidos ao final da criação dos grafos auxiliares.

<i>Espaços</i>	$ V $	$ A $	$ V  +  A $
<b>Grafo original</b>	6	14	20
<b>Modelagem 1</b>	26	49	75
<b>Modelagem 2</b>	20	35	55
<b>Modelagem 3</b>	27	51	78
<b>Modelagem 4</b>	34	49	83

**Tabela 4.2:** Tamanho dos grafos resultantes medidos.

A partir destes números, fica claro que, para o grafo  $G$  da Figura 4.1, a Modelagem 3 – utilizada por Walid em [13] – é a que apresenta o maior número de arcos, o segundo maior número de vértices e a segunda maior soma de arcos e vértices de  $G'$ . Comparando com a abordagem que apresenta os menores números, a Modelagem 3 apresenta aumentos de 35% em  $|V'|$ , 45% em  $|A'|$  e 41% na soma dos vértices e arcos. Isso mostra que esta modelagem não é a mais indicada, em termos de economia de espaço, para re-

resolver o SPPFP tendo como entrada o grafo  $G$  e seu conjunto  $F$  de caminhos proibidos utilizados nos testes.

Por outro lado, a Modelagem 2 é a que possui os menores números. Ela apresenta menor quantidade de vértices, de arcos e, conseqüentemente, menor soma de arcos e vértices, aparecendo como a melhor opção em economia de espaço. Com tais informações, temos que esta modelagem é a mais indicada para resolver a instância do SPPFP apresentada nesta Seção.

## 4.2 Trabalhos Futuros

Mesmo com as comparações apresentadas na Seção 4.1, é interessante a realização de testes com implementações das abordagens, para que se possa fazer análises práticas das informações obtidas, levando em consideração tanto o espaço utilizado por cada grafo resultante quanto o tempo exigido para criação do mesmo. Com o resultado das análises será possível escolher a abordagem que traz os melhores resultados para cada tipo de caso.

Os próximos trabalhos incluem implementações das abordagens deste projeto, realização de conjuntos de testes com as implementações e apresentação de conclusões a respeito dos resultados, e devem ser publicados na forma de artigo(s).

As quatro abordagens apresentadas neste documento serão implementadas conforme foram descritas e todas receberão os mesmos conjuntos de testes. Como este projeto é voltado para a malha de trânsito urbana de Goiânia, os testes conterão dados reais das vias desta cidade e poderão conter subgrafos divididos por setor, região e/ou outras áreas de interesse. Com os resultados dos testes, serão feitas comparações entre as abordagens, distinguindo os aspectos positivos e negativos de cada abordagem para cada conjunto de testes. Assim será possível fazer a escolha de qual implementação deve ser utilizada para cada configuração dos dados.

---

## Referências Bibliográficas

---

- [1] AHMED, M.; LUBIW, A. **Shortest paths avoiding forbidden subpaths**. In: Albers, S.; Marion, J.-Y., editors, *26th International Symposium on Theoretical Aspects of Computer Science (STACS 2009)*, volume 3 de **Leibniz International Proceedings in Informatics (LIPIcs)**, p. 63–74, Dagstuhl, Germany, 2009. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
- [2] AHO, A. V.; CORASICK, M. J. **Efficient String Matching: an Aid to Bibliographic Search**. *Communications of the ACM*, 18(6):333–340, June 1975.
- [3] BELLMAN, R. **On a routing problem**. *Quarterly of Applied Mathematics*, 16(1):87–90, 1958.
- [4] CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. **Introduction to Algorithms**. McGraw-Hill Science, second edition, September 2001.
- [5] DIJKSTRA, E. W. **A note on two problems in connexion with graphs**. *Numerische Mathematik*, 1(1):269–271, December 1959.
- [6] DO NASCIMENTO, H. A. D. **Produção informal**, 2008.
- [7] DROR, M. **Note on the Complexity of the Shortest Path Models for Column Generation in VRPTW**. *Operations Research*, 42(5):977–978, 1994.
- [8] FLOYD, R. W. **Algorithm 97: Shortest path**. *Communications of the ACM*, 5(6):345, June 1962.
- [9] FORD JR., L. R. **Network flow theory**. Paper P-923, The RAND Corporation, Santa Monica, California, August 1956.
- [10] FOULDS, L. R. **Produção informal**, 2010.
- [11] GABOW, H. N.; MAHESHWARI, S. N.; OSTERWEIL, L. J. **On two problems in the generation of program test paths**. *IEEE Transactions on Software Engineering*, 2(3):227–231, May 1976.

- [12] GAREY, M. R.; JOHNSON, D. S. **Computers and Intractability: A Guide to the Theory of NP-Completeness**. W. H. Freeman & Company, New York, NY, USA, 1979.
- [13] JRADI, W. A. R. **Uma arquitetura de software interativo para apoio à decisão na modelagem e análise do tráfego urbano**. Master's thesis, Instituto de Informática – Universidade Federal de Goiás, December 2008.
- [14] KJERRSTRÖM, J. **The resource constrained shortest path problem**. *Resources*, June 2001.
- [15] KOLMAN, P.; PANGRÁC, O. **On the Complexity of Paths Avoiding Forbidden Pairs**. *Discrete Applied Mathematics*, 157(13):2871–2876, July 2009.
- [16] MACHADO, P.; TAVARES, J.; PEREIRA, F. B.; COSTA, E. **Vehicle routing problem: Doing it the evolutionary way**. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2002*, p. 690, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.
- [17] MARTINS, E. Q. V. **An algorithm for ranking paths that may contain cycles**. *European Journal of Operational Research*, 18(1):123–130, October 1984.
- [18] PECIN, D. G. **Uso de rotas elementares no cvrp**. Master's thesis, Universidade Federal de Goiás, 2010.
- [19] SHIMA, R. B. **Caminhos mínimos sob restrições: Uma revisão e aplicações**. Master's thesis, Universidade Federal de Santa Catarina, June 2006.
- [20] SZEIDER, S. **Finding paths in graphs avoiding forbidden transitions**. *Discrete Applied Mathematics*, 126(2-3):261–273, March 2003.
- [21] UEHARA, R.; UNO, Y. **Efficient algorithms for the longest path problem**. In: Fleischer, R.; Trippen, G., editors, *Algorithms and Computation*, volume 3341 de **Lecture Notes in Computer Science**, p. 1547–1553. Springer Berlin/Heidelberg, 2005.
- [22] VILLENEUVE, D.; DESAULNIERS, G. **The shortest path problem with forbidden paths**. *European Journal of Operational Research*, 165(1):97–107, 2005.
- [23] WEST, D. B. **Introduction to Graph Theory**. Prentice Hall, second edition, 2001.
- [24] ZIEGELMANN, M. **Constrained Shortest Paths and Related Problems**. PhD thesis, Universität des Saarlandes, July 2001.