UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

DEPARTAMENTO DE INFORMÁTICA

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

THAYSE DOBIS BARROS

OTIMIZAÇÃO NO ESCALONAMENTO DE VEÍCULOS PARA ATENDIMENTO DE PEDIDOS PRÉ-ESTABELECIDOS

TRABALHO DE CONCLUSÃO DE CURSO DE GRADUAÇÃO

PONTA GROSSA

2014

THAYSE DOBIS BARROS

OTIMIZAÇÃO NO ESCALOMENTO DE VEÍCULOS PARA ATENDIMENTO DE PEDIDOS PRÉ-ESTABELECIDOS

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas do Departamento de Informática, da Universidade Tecnológica Federal do Paraná.

Orientadora: Profª. Drª. Sheila Morais de Almeida

PONTA GROSSA

2014

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Ponta Grossa
Nome da Diretoria
Nome da Coordenação
Nome do Curso

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

TERMO DE APROVAÇÃO

OTIMIZAÇÃO NO ESCALONAMENTO DE VEÍCULOS PARA ATENDIMENTO DE PEDIDOS PRÉ-ESTABELECIDOS

por

THAYSE DOBIS BARROS

Este Trabalho de Conclusão de Curso (TCC) foi apresentado em 09 de Junho de 2014 como requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas. A candidata foi arguida pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Sheila Morais de Almeida
Prof.(a) Orientador(a)

Saulo Jorge Beltrão de Queiroz
Membro titular

Willian Massami Watanabe
Membro titular

- O Termo de Aprovação assinado encontra-se na Coordenação do Curso -

Dedico este trabalho a quem o inspirou, e
é a minha fonte de inspiração e exemplo:
meu pai.

AGRADECIMENTOS

Agradeço à minha orientadora, professora Sheila Morais de Almeida, pelo carinho com que me recebeu desde o primeiro contato. Muito obrigada pela atenção e paciência. Seu entusiasmo, dedicação, e sua maneira de transformar o tema mais complexo em um desafio envolvente, são inspirações.

Ao professor Gleifer Vaz Alvez, agradeço pela atenção, interesse ao trabalho e indicação à minha orientadora.

Ao professor Daniel Costa de Paiva, agradeço pelo valioso tempo que dispendeu auxiliando nos testes do algoritmo.

Ao PROGRAD/UTFPR-PG, pela concessão de uma bolsa para que eu realizasse o trabalho.

Aos amigos da UTFPR, agradeço pelo carinho, companheirismo e por tornarem mais leve a rotina.

Agradeço especialmente aos meus pais, Rogério e Silvana, pela vida, pelos preciosos valores que me transmitiram, pela amizade e amor incondicionais. Ao meu irmão Victor, muito obrigada pelos gestos tão maduros de carinho e apoio, que sempre me surpreendem. Ao Lucan, meu amor, agradeço por adotar os meus sonhos e apoiá-los incondicionalmente. A sua presença, mesmo longe, torna os meus dias sempre melhores. Vocês são meu alicerce.

Por fim, agradeço Àquele que foi a minha única companhia em todas as horas de estudo, concedendo-me força, sabedoria e determinação, tanto quanto eu necessitasse. Deus, muito obrigada.

RESUMO

BARROS, Thayse D. **Otimização no escalonamento de veículos para atendimento de pedidos pré-estabelecidos**. 2014. 122. Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas) - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2014.

A gestão de frotas para transporte de cargas envolve grande número de decisões, tais como: dimensionamento, especificação de equipamentos, roteirização, custos e manutenção. Este trabalho é voltado para a atividade de roteirização. Seu objetivo consiste em desenvolver um método eficiente para realizar o atendimento de pedidos previamente estabelecidos. Considera-se que a carga de um pedido preenche totalmente o veículo, a capacidade de todos os veículos é a mesma, o produto deve ser carregado em uma origem e descarregado em um destino predefinidos e que cabe ao método desenvolvido decidir qual o próximo pedido a ser atendido. O potencial da ferramenta é verificado pela redução de custos e pelo tempo de execução, que possibilitam ampla aplicabilidade.

Palavras-chave: *Branch and Bound*. Problema do Caixeiro Viajante. Roteamento.

ABSTRACT

BARROS, Thayse D. Optimizing the scheduling of vehicles for attendance of pre-established orders. 2014. 122. Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas) - Federal Technology University - Parana. Ponta Grossa, 2014.

Fleet management to cargo transportation involves many decisions, such as: design, the equipment specification, routing, and maintenance costs. This work is focused on routing activity. The objective is to develop an efficient method to perform the scheduling of vehicles for attending pre-established orders. It is considered that the cargo order completely fills the vehicle, the capacity of all vehicles is the same, the source and destination where the cargo must be loaded and unloaded should be known beforehand. To decide whitch request should be handled next the developed method should be applied. The potential of the tool is verified by reducing costs and by its runtime, which enables wide applicability.

Keywords: Branch and Bound. Traveling Salesman Problem. Routing.

SUMÁRIO

1 INTRODUÇÃO

O transporte de cargas pelo sistema rodoviário no Brasil tem uma estrutura respeitável, sendo o principal meio de transporte de cargas no País e desempenhando um papel vital para a economia da Nação. O termo “gestão de frotas” representa a atividade de gerenciar um conjunto de veículos pertencentes a uma mesma empresa. Esta tarefa tem uma abrangência ampla e envolve serviços como: dimensionamento, especificação de equipamentos, roteirização, custos, manutenção, entre outros (NOVAES, 2004).

Segundo Chopra e Meindl (2011), as dificuldades de maximização da eficiência e racionalização nos processos de gestão de frotas levam à adoção de procedimentos empíricos ou intuitivos que, muitas vezes, estão distantes do ótimo. Assim, avanços na área de informática e engenharia estão sendo absorvidos pelos transportadores, com o intuito de otimizar seu processo produtivo.

Este trabalho visa propor um método eficiente para resolução do problema de escalonamento de veículos no contexto da gestão de frotas. O método deverá receber como entrada as informações de custos monetário e/ou temporal para todos os percursos entre as origens e destinos envolvidos nos pedidos de transporte de carga, os pedidos em si, o número de veículos disponíveis e a posição inicial de cada um deles. Quando o custo for temporal, os custos de tempo para carga nas origens e descarga nos destinos também deverão ser considerados. Como resultado, determina-se o somatório dos custos de transporte e a sequência de pedidos a serem atendidos por cada um dos veículos disponíveis, de forma que todos os pedidos sejam atendidos e que o custo do processo seja minimizado.

1.1 MODELAGEM DO PROBLEMA

A modelagem do problema por meio de grafos orientados faz com que cada nó corresponda a um pedido e cada aresta orientada, de v para w, corresponda ao atendimento do pedido v, seguido imediatamente pelo pedido w. Um caminho orientado v_1, v_2, v_3 ... v_q indica que os pedidos v_1, v_2, v_3 ... v_q devem ser atendidos na ordem: primeiro v_1, depois v_2, e assim sucessivamente até v_q.

Como v_1 é o primeiro a ser atendido, então, não existe aresta v_q-v_1 na solução procurada, pois ao atender ao pedido v_q, sabe-se que o pedido v_1 já foi atendido anteriormente.

A definição do problema não impõe nenhuma restrição na ordem de atendimento dos pedidos, sendo qualquer ordem válida. Assim, todas as alterativas devem ser representadas no grafo, que é completo com orientação de v_i para v_j e vice-versa entre todos os pares de vértices.

Como o vértice v_i representa o atendimento do pedido P_i, as ações representadas por v_i são:

- O carregamento da carga na origem do pedido i;
- O transporte da carga da origem i até o destino do pedido i;
- O descarregamento ou entrega da carga do pedido i.

A aresta v_i-v_j representa o deslocamento de um veículo, após o atendimento do pedido *i*, para a origem do pedido *j*, aonde o veículo deverá ser carregado para atendimento do pedido *j*. Este custo visa ser minimizado neste trabalho, e é denominado custo variável.

Por sua vez, o custo de deslocamento do veículo carregado entre a origem e o destino de um mesmo pedido não são representados no grafo. Essa decisão foi tomada considerando-se que o custo desse deslocamento é inevitável, visto que o problema considera que todos os pedidos serão obrigatoriamente atendidos. Este peso foi denominado de custo fixo.

Enquanto a aresta v_i-v_j representa o deslocamento do destino do pedido *i* (um cliente) para a origem do pedido *j* (um ponto de carregamento), a aresta v_j-v_i representa o deslocamento do destino do pedido *j* para a origem do pedido *i*. Como o destino do pedido *i* e o destino do pedido *j* não coincidem necessariamente, bem como as origens de ambos os pedidos, as arestas v_i-v_j e v_j-v_i podem ter pesos diferentes e, portanto, o grafo é assimétrico.

Quando a carga de um pedido for maior que a capacidade do veículo, o modelo adotado divide esse pedido *Pi* em pedidos menores que possam ser atendidos em uma viagem, aonde *k* vezes a capacidade do veículo é o tamanho da carga do pedido P_i. Todos os pedidos $P_{i,j}$ com *i* fixo e $1 \leq j \leq k$ carregam na mesma origem e descarregam no mesmo destino, podendo ser atendido por veículos diferentes. Mesmo quando dois pedidos são atendidos por um mesmo veículo, esse atendimento não é necessariamente consecutivo.

Considerando o modelo apresentado, deseja-se encontrar um conjunto de caminhos disjuntos nos vértices e tal que todo vértice pertença a algum desses caminhos. Cada caminho será uma sequência de pedidos a ser atendido por um determinado veículo e a soma das arestas desse caminho é a soma dos custos variáveis para atendimento dos pedidos do mesmo. Deseja-se encontrar um conjunto de caminhos que tenha a menor soma possível do custo das arestas.

Durante o levantamento bibliográfico, identificou-se um amplo conjunto de problemas de otimização famosos cuja modelagem como problema de programação linear é bem conhecida, como: Problema da Designação, do Transporte, do Escalonamento de Tarefas e do Caixeiro Viajante. As definições e modelagens de todos esses problemas podem ser vistas em Camponogara (2006). Dentre todos os modelos estudados, o que se mostrou mais compatível com o problema apresentado foi o Problema do Caixeiro Viajante (PCV). Para Goldbarg (2005), o Problema do Caixeiro Viajante é um dos mais tradicionais problemas de programação matemática, de importância indiscutível, tanto sob aspecto prático como teórico. Nele, um caixeiro viajante tem por objetivo sair de sua cidade e percorrer todas as cidades de um mapa, passando por cada uma delas exatamente uma vez, gastando o mínimo possível no trajeto e retornando à sua cidade ao final do percurso. O problema tem modelos teóricos na Teoria dos Grafos e na Programação Linear. Sua aplicação, além da definição clássica da visita em cidades, estende-se a problemas de sequenciamento de produção, roteirização, manufatura de circuitos integrados, problemas de distribuição, separação de pedidos em armazéns, etc.

A Figura 1 apresenta um exemplo de grafo orientado que contém todos os possíveis trechos a serem escolhidos pelo caixeiro viajante e os respectivos custos para percorrê-los, no caso de 4 cidades.

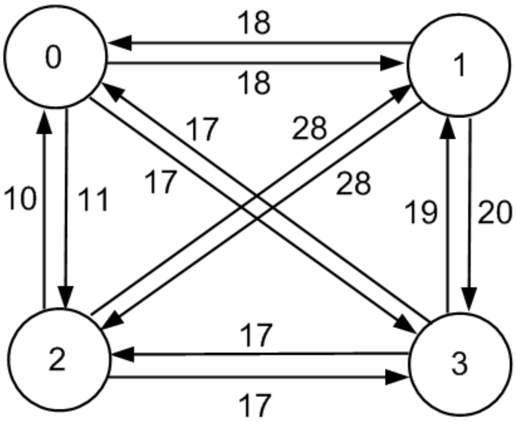


Figura 1 - Exemplo de possiblidades de trechos do PCV de 4 cidades
Fonte: Autoria Própria (2014)

A modelagem do Problema do Caixeiro Viajante como um problema de Programação Linear impõe duas restrições importantes. A primeira delas é que sejam utilizadas exatamente duas estradas para cada cidade: uma para se chegar a ela e outra para sair da mesma. Ao considerar apenas essa restrição, a solução do problema pode ser composta de um conjunto de circuitos disjuntos, como mostra a Figura 2.

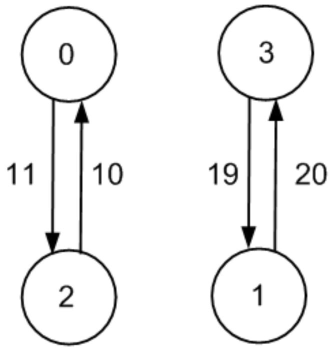


Figura 2 - Solução com circuitos disjuntos
Fonte: Autoria Própria (2014)

Como a Figura 2 não é uma resposta válida para o caixeiro viajante, impõe-se a segunda restrição, que garante que todas as cidades estão no mesmo circuito, tal como exemplifica a Figura 3.

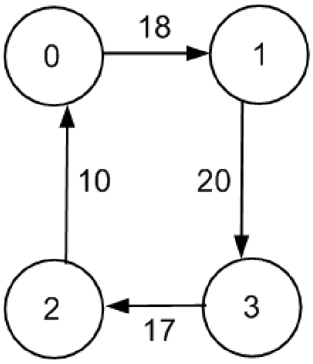


Figura 3 - Solução da designação com um circuito (solução do PCV)
Fonte: Autoria Própria (2014)

A diferença entre o PCV e o problema em estudo é que o PCV exige o retorno ao ponto de origem. Além disso, o PCV exige um único ciclo, enquanto o problema tratado permite a existência de mais de um caminho.

A solução do PCV em uma instância do problema de atendimento de pedidos pré-estabelecidos equivale a atender todos os pedidos com um único veículo e retornar à origem (ponto de carregamento) do primeiro pedido atendido.

Para resolver o problema proposto, será feita uma relaxação do Problema do Caixeiro Viajante que elimina a restrição que faz a eliminação de subcircuitos e o iguala ao chamado Problema da Designação. Este problema torna-se interessante pelo fato de que soluções com vários ciclos na prática permitem que cada um desses ciclos (sequência de pedidos) sela atendido por um veículo diferente. Sendo veículos diferentes, esse atendimento pode ocorrer concomitantemente, diminuindo o tempo total de atendimento.

Entretanto, como as soluções do problema da designação são ciclos disjuntos nos vértices e, como o problema tratado exige como solução um conjunto de caminhos, serão realizadas alterações na solução para que a mesma esteja em conformidade com as restrições do problema estudado.

1.1.1 Exemplo de Solução do Problema

Para exemplificar, considera-se que determinada transportadora possui os três pedidos a atender, conforme mostra a Tabela 1:

Tabela 1 - Exemplo de pedidos a atender

Cód. do Pedido	Tipo de Carga	Origem	Destino	Quantidade (t)
1	A	O1	D1	120
2	B	O2	D2	30
3	C	O3	D3	60

Fonte: Autoria Própria (2014)

Supondo-se que a mesma informe que tem 4 veículos disponíveis, todos com capacidade de transporte de 30t por carregamento. A relação completa de carregamentos é ilustrada na Tabela 2:

Tabela 2 - Exemplo de carregamentos a atender

Cód. Pedido	Tipo de Carga	Origem	Destino	Quantidade (t)
1 – 1	A	O_P1	D_P1	30
1 – 2	A	O_P2	D_P2	30
1 – 3	A	O_P3	D_P3	30
1 – 4	A	O_P4	D_P4	30
2 – 1	B	O_P5	D_P5	30
3 – 1	C	O_P6	D_P6	30
3 – 2	C	O_P7	D_P7	30

Fonte: Autoria Própria (2014)

Aonde os quatro primeiros carregamentos (1 – 1, 1 – 2, 1 – 3, 1 – 4) devem ser realizados para atendimento do pedido 1, cuja demanda é 4 vezes maior (120t) que a capacidade máxima dos veículos (30t). Para o pedido 2, apenas um carregamento (2 – 1) atende à demanda de 30t. Para o pedido 3, são necessários dois carregamentos para completar o total de 60t solicitadas.

Como foi dito, os trechos entre as origens e os destinos de cada pedido serão percorridos obrigatoriamente quando os veículos estão fazendo a entrega, carregados. Este trabalho denomina o somatório de custos decorrentes de tais percursos obrigatórios de *Custo_fixo*, A Tabela 3 mostra a matriz de custos fixos (em valores monetários) do exemplo, denominada *M_fixo*.

Tabela 3 - Exemplo de matriz de custos fixos

	D1	D2	D3
O1	466	122	289
O2	333	418	382
O3	600	130	354

Fonte: Autoria Própria (2014)

Para o exemplo, o custo fixo total será:

$$Custo\_fixo = \sum_{i=1}^{N} M\_fixo\left[O\_P_i\right]\left[D\_P_i\right]$$

$$Custo\_fixo = 466 + 466 + 466 + 466 + 418 + 354 + 354$$

$$Custo\_fixo = 2613,00$$

Sendo N = 7, o número de pedidos da Tabela 2.

Uma vez estipulado que há 4 veículos disponíveis para cumprir o total de 7 pedidos, o escalonamento ótimo obrigatoriamente atribuirá mais de um pedido à mesma rota. Serão construídas 4 rotas, uma para cada veículo.

Supondo que o veículo 1 seja designado para atender, nesta ordem, aos pedidos 2, 4 e 5, da Tabela 2. Além dos trechos fixos O2→D2, O4→D4 e O5→D5, o veículo teria de se deslocar, descarregado, de D2 para O4 e de D4 para O5. A rota final percorrida por este veículo é5 apresentada na Figura 4:

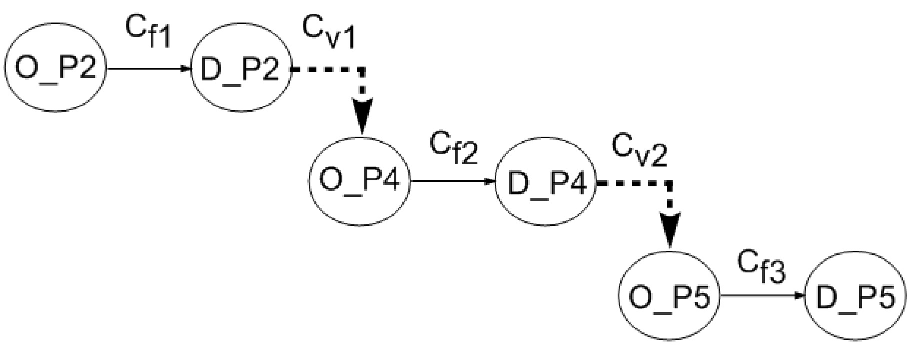


Figura 4 - Rota suposta para veículo 1
Fonte: Autoria Própria (2014)

Observa-se que, além dos custos fixos denotados por C_{f1}, C_{f2} e C_{f3}, há os custos C_{v1} e C_{v2}, representados pelas retas tracejadas, que correspondem aos custos referentes ao deslocamento do veículo descarregado entre o destino de um pedido e a origem do próximo pedido. Tais custos são denominados de custos variáveis, já que o seu valor depende da próxima origem a ser escolhida, de forma que quanto menor a distância entre o destino atual e a próxima origem, menor o custo. Assim pode-se construir uma matriz com os custos de viagem com veículo descarregado de cada possível destino (onde um pedido se encerra) para cada possível origem (onde um novo pedido será carregado). Essa matriz é chamada de matriz de custo variável e seus dados são os principais determinantes da ordem em que os pedidos devem ser atendidos. A Tabela 4 apresenta a matriz de custo variável (*M_variavel*), dada para o exemplo.

Tabela 4 - Exemplo de matriz de custos variáveis

	O1	**O2**	**O3**
D1	440	100	209
D2	323	408	302
D3	520	100	324

Fonte: Autoria Própria (2014)

Os custos da matriz de custo variável podem ser determinados de diversas formas. Por exemplo, verificou-se em um estudo de caso que quando os trechos de O_i para D_i são fisicamente os mesmos e os custos sejam decorrentes, dentre outros fatores, do combustível, esses tendem a ser em torno de 30% mais baratos quando o veículo está descarregado.

Os custos podem também representar custos de tempo. Neste caso, o valor da célula (D1, O1), na Tabela 4, constitui a soma dos custos de tempo de descarga em D1, o tempo de deslocamento de D1 a O1 e o tempo de carregamento em O1. Uma vez construída a tabela, as etapas seguintes são as mesmas tanto para custos monetários, quanto para custo temporal.

Em seguida, a matriz (*M*), cujas linhas são os destinos dos pedidos de 1 até N, as colunas são as origens de 1 até N e o valor das células é o custo variável do transporte, pode ser estruturada conforme a Tabela 5:

Tabela 5 - Exemplo de matriz de custos variáveis para pedidos

	O_P1	O_P2	O_P3	O_P4	O_P5	O_P6
D_P1	∞	440	440	100	209	209
D_P2	440	∞	440	100	209	209
D_P3	440	440	∞	100	209	209
D_P4	323	323	323	∞	302	302
D_P5	520	520	520	100	∞	324
D_P6	520	520	520	100	324	∞

Fonte: Autoria Própria (2014)

Por meio desta, objetiva-se realizar a designação do destino de cada pedido à origem de exatamente um próximo pedido a atender e que o destino de cada pedido seja atendido por exatamente uma origem, de forma que o somatório dos custos variáveis seja mínimo. Verifica-se que todos os valores da diagonal principal da matriz recebem o valor infinito. Isto é feito por que não é possível designar-se o destino de um pedido à sua própria origem. Caso o valor zero fosse atribuído aos elementos da diagonal principal, a designação ótima seria sempre formada por estes elementos. Do ponto de vista teórico, tal designação seria totalmente correta. Do ponto de vista prático, entretanto, não seria interessante para uma transportadora com 100 pedidos ter de possuir 100 veículos para atendê-los. Assim, optou-se por atribuir o valor infinito à diagonal principal e excluir tal possibilidade.

Para o exemplo, a designação ótima é dada pela escolha das células cujas destacadas. O somatório dos custos variáveis é :

$$Custo\_variavel = 209 + 209 + 440 + 323 + 100 + 520$$
$$Custo\_variavel = 1801$$

A interpretação da designação da Tabela 5 por meio de grafos é dada na Figura 5, aonde os vértices numerados são os pedidos, as arestas orientadas indicam a sequência a seguir em cada rota e o peso de cada aresta é o valor do custo variável para o pedido. Observa-se que houve a formação de 2 rotas.

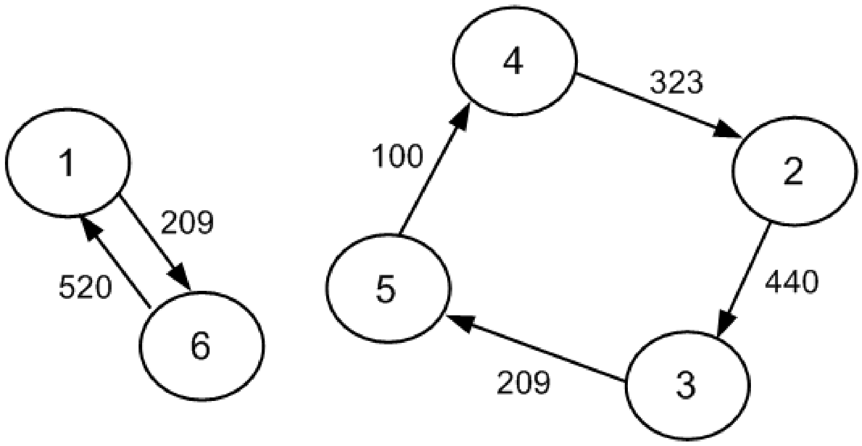


Figura 5 – Grafo resultante da designação
Fonte: Autoria Própria (2014)

Como foi dito, um número de veículos disponíveis (*Num_veículos*) é informado pelo usuário ao sistema. Este número e o número de rotas (*Num_rotas*) do grafo acima são pontos chave do algoritmo. Caso eles sejam diferentes, algoritmos que serão detalhados posteriormente serão executados até que eles sejam igualados, mesmo que isso acarrete no aumento do valor do *Custo_variavel.*

Essa medida foi tomada considerando-se que pode ser importante atender aos pedidos o mais rápido possível, mesmo que se tenha que usar mais veículos e se sujeitar a custos mais altos, por questões como prazos, por exemplo.

Após a execução dos algoritmos que alteram o *Num_rotas*, outro algoritmo é executado para escolher o primeiro pedido a atender em cada rota e designar um veículo a cada rota.

O resultado final (*Custo_total)* consiste no somatório de: custos variáveis, custos fixo e custo com deslocamento (*Custo_desloc*) da posição inicial do veículo (informada pelo usuário) até a posição da origem do primeiro pedido a ser atendido na rota designada ao mesmo.

$$Custo\_total = Custo\_variavel + Custo\_desloc + Custo\_fixo$$

1.2 DELIMITAÇÃO DO TEMA

Considerando o problema apresentado e a sua modelagem, as seguintes delimitações podem ser citadas:

- Carga completa: para utilização do algoritmo desenvolvido, é necessário que os carregamentos não tenham carga consolidada. Isto é, todo carregamento terá apenas a carga de um mesmo pedido, mesmo que esta não atinja a capacidade máxima de carga. Não é possível aplicar o algoritmo a carregamentos que tenham diferentes tipos de carga na mesma viajem;
- É obrigatório o atendimento de todos os pedidos exatamente uma vez;
- O custo de deslocamento do veículo carregado é considerado inevitável, e, portanto, não pode ser minimizado. A única forma de reduzi-lo seria eliminando alguns pedidos, o que não é a intenção de uma empresa transportadora, por exemplo;

1.3 COMPLEXIDADE

O Problema do Caixeiro Viajante é *NP*-Completo (CORMEN *et al*., 2009). A complexidade de um problema é determinada pelo número de operações que um algoritmo executa para resolvê-lo. Os problemas da Classe *P* são aqueles em que os algoritmos executam um número de instruções que é polinomial em função do tamanho da entrada (diz-se que o algoritmo executa em tempo polinomial). A classe *NP* é a classe dos problemas em que o número de soluções possíveis é, em geral, exponencial e a única forma conhecida de se encontrar a melhor solução, atualmente, é percorrendo todo o espaço de soluções realizando comparações entre os mesmos. Essa técnica é chamada de forma bruta. Porém, comparar todas as possíveis soluções exige um número de instruções computacionais exponencial quando se aplica a técnica de força bruta.

Quando se pode fazer uma redução em tempo polinomial de qualquer problema da classe *NP* a um dado problema *A*, diz-se que *A* é um problema *NP*-Completo. Assim, a classe dos problemas *NP*-Completos é o conjunto dos problemas aos quais qualquer problema de *NP* pode ser reduzido em tempo polinomial. Para esses, não se conhece nenhum algoritmo com tempo de execução polinomial. Entretanto, também não se provou que o mesmo não existe. Sabe-se também que, caso seja apresentado um algoritmo polinomial para algum dos problemas *NP*-Completo, então todos eles possuem um algoritmo polinomial e, portanto, pertenceriam à classe *P*. A questão *P* = *NP*? É uma importante questão ainda em aberto para a ciência, uma vez que encontrar uma solução polinomial de um *NP*-Completo acarretaria na possibilidade de solução polinomial de todos os *NP*, o que engloba uma vasta gama de aplicações (CORMEN *et al*., 2009).

Em relação ao Problema do Caixeiro Viajante, devido ao caráter altamente combinatório do problema, usualmente são utilizados métodos exatos (aqueles que garantem encontrar a solução ótima) para a solução de problemas de menor porte e métodos heurísticos para aqueles de maior porte. Os métodos heurísticos são técnicas que procuram boas soluções em tempos razoáveis, sem que haja a garantia de otimalidade.

1.4 METODOLOGIA

Para resolver o problema do escalonamento de veículos neste trabalho será seguido o método denominado *Branch and Bound.* Neste, será feita uma relaxação do Problema do Caixeiro Viajante que omite a restrição responsável pela eliminação de subcircuitos e o equipara a outro conhecido problema chamado Problema da Designação, pertencente à classe *P*. Esta abordagem permite que a solução ótima do problema seja composta por vários ciclos. A solução é bastante atraente para o problema em estudo, pois os circuitos podem ser tratados em paralelo e considerados como rotas a serem percorridas cada uma por um veículo. Assim, aproveita-se a possibilidade da existência de mais de um veículo disponível para redução da complexidade do algoritmo. Na solução, as cidades do PCV são os pedidos a serem atendidos e as arestas orientadas que as unem constituem a ordem de atendimento dos pedidos.

Para resolver o Problema da Designação será utilizado o Método Húngaro. O Método Húngaro, apresentado por Harold Kuhn em 1955, é um algoritmo polinomial frequentemente utilizado para resolver o Problema da Designação (KUHN, 1955). Caso o número de circuitos (rotas) da solução ótima do Problema da Designação seja equivalente ao número de veículos disponíveis informado pelo usuário, a atribuição de rotas de menor custo terá sido encontrada. Caso o valor seja diferentes, heurísticas para ambas as possibilidades (número de rotas maior ou menor que o de veículos) serão utilizadas para fornecer limites ao algoritmo exato de *Branch and Bound* (B&B), que irá buscar rotas que minimizam o custo e utilizam todos os veículos disponíveis.

O algoritmo *Branch and Bound* é exato. Nele, relaxam-se uma ou mais restrições do problema e constrói-se uma árvore onde cada nó é uma possível solução para o problema relaxado. Além disso, os filhos de um nó são soluções com pequena variação em relação ao seu pai e com custo maior ou igual ao do pai. A árvore é construída de forma a considerar as possíveis soluções do problema relaxado. O algoritmo realiza uma busca na árvore a partir da raiz e avalia as soluções, continuando em cada ramo da árvore sempre que a restrição relaxada ainda não foi satisfeita e há possibilidade de se encontrar uma solução com custo menor que a melhor solução vista até o momento. Quando o custo da solução é muito alto, o ramo é cortado, indicando que as soluções daquela subárvore não são boas o suficiente para serem consideradas.

O procedimento é citado por Taha (2008), que afirma que a resolução do PCV por meio do B&B inicia-se com a solução ótima do Problema da Designação associado. Se a solução for um circuito, o processo termina. Se não, são impostas restrições para eliminar os circuitos indesejados. Todos os procedimentos que alteram o número de rotas resultam em custos maiores ou iguais ao determinado pelo Problema da Designação, uma vez que este encontra o número de rotas cíclicas de custo mínimo.

O procedimento que determinará se o número de rotas da solução é equivalente ou não ao número de veículos disponíveis é uma busca em grafos. Trata-se de um tipo de busca que examina sistematicamente os vértices e arcos de um grafo por meio de uma pilha (busca em profundidade) ou de uma fila (busca em largura), que contém todos os vértices visitados cujos vizinhos ainda não foram todos visitados (SEDGEWICK, 2002).

Destaca-se ainda que o problema a ser resolvido no presente trabalho deve apresentar rotas que são caminhos, e não ciclos. São apresentadas duas heurísticas para escolher qual será o primeiro pedido a ser atendido em cada rota. É também necessário determinar qual veículo deve atender a qual rota de forma que o custo de deslocamento do mesmo até a origem do primeiro pedido seja mínimo. A medida de desempenho será dada pelo tempo de execução do método e pelo somatório de custos envolvidos, que devem ser minimizados.

1.5 OBJETIVOS

1.5.1 Objetivo Geral

Modelar, implementar e solucionar computacionalmente o problema de otimização no escalonamento de veículos para atendimento de um conjunto de pedidos pré-estabelecidos para um dado conjunto de recursos.

1.5.2 Objetivos Específicos

Os objetivos específicos desse Trabalho de Conclusão de Curso são:

- Caracterizar o problema de escalonamento de veículos no contexto da gestão de frotas;
- Estudar os problemas de programação linear disponíveis na literatura que se assemelham ao problema estudado;
- Ampliar os conhecimentos na linguagem de programação C através da implementação de uma solução para o problema nesta linguagem.
- Estudar algoritmos em grafos.
- Aprender técnicas de solução de problemas de otimização e desenvolver a habilidade de aplicá-las a problemas reais.

1.6 JUSTIFICATIVA

Segundo a o Portal da ANTT (Agência Nacional de Transportes Terrestres) (2013), atualmente o Brasil possui ao todo 836.784 entidades transportadoras de carga (por meio do sistema rodoviário) registradas, sendo estas divididas entre autônomos, empresas e cooperativas.

Através de pesquisas em *web sites*, verificou-se que tais empresas contam com uma gama de opções de softwares de gestão que incluem, de modo geral, os módulos: Fretes, Veículos, Pneus, Manutenções, Acertos Com Motoristas, Financeiro, Despesas, Cobrança Eletrônica, Relatórios, entre outros.

Apesar de tais módulos serem de fundamental importância para a administração das empresas do ramo, nota-se a ausência de módulos que auxiliem os administradores a distribuir seus veículos de maneira ótima. Verificou-se que não há módulos que realizem o planejamento da produção de forma a garantir que o somatório dos custos envolvidos seja o mínimo possível, ao mesmo tempo em que todas as restrições de demanda, prazo, mão de obra, etc. sejam atendidas.

Uma vez que tais problemas, quando modelados matematicamente, são de elevada complexidade e requerem um grande volume de processamento de cálculos, sua resolução manual torna-se inviável.

Deste modo, verifica-se que a implementação de um algoritmo capaz de otimizar a designação de veículos na gestão de frotas é uma ferramenta de potencial eficiência para redução de custos do processo, com larga aplicabilidade, bem como uma oportunidade para que as empresas transportadoras melhorem sua competitividade.

1.7 PROCEDIMENTOS METODOLÓGICOS

Para alcançar os objetivos mencionados anteriormente, buscou-se inicialmente estruturar as características do problema a ser resolvido por meio da identificação das necessidades em cenários reais. Esta ocorreu por meio de entrevistas com dois gestores de empresas de transporte de cargas rodoviárias, leitura de periódicos específicos, revisão bibliográfica sobre logística e gestão de frotas, consulta em *web sites* e uso de versão de demonstração de sistema comercial de roteamento de veículos.

Posteriormente, efetuou-se uma ampla revisão bibliográfica nas áreas de Programação Linear, Algoritmos, Análise Combinatória e Teoria dos Grafos. Problemas de Programação Linear e da Teoria dos Grafos como: Fluxo em Rede, Transporte, Roteamento de Veículos, Alocação de Berços, Escalonamento, Designação, Carteiro Chinês, Caixeiro Viajante, entre outros, foram analisados buscando-se a adaptação das características ao problema proposto.

Determinou-se que o Problema do Caixeiro Viajante com a relaxação da restrição que elimina os subcircuitos (resultando no Problema da Designação) foi aquele ao qual o problema de escalonamento de veículos com atendimento de pedidos pré-estabelecidos melhor se adaptou. Assim, buscou-se a determinação do estado da arte dos métodos de resolução deste algoritmo clássico. A literatura neste contexto é vasta e salienta a importância de estudar-se o problema, a fim de contribuir com a redução da complexidade computacional de sua resolução.

A linguagem de programação escolhida para a implementação do algoritmo foi a linguagem C. Iniciou-se o desenvolvimento do sistema com a especificação de quais seriam os dados de entrada. A partir de tais dados, implementou-se o Algoritmo Húngaro para encontrar a solução relaxada do problema, aquela que preocupa-se apenas com a minimização do custo total, ignorando a restrição quanto ao número de veículos disponíveis. Uma busca em largura foi desenvolvida para encontrar qual é o número de veículos dados pela solução do Método Húngaro. Para casos em que o número de veículos do problema relaxado é diferente do número de veículos disponibilizados para transporte, desenvolveu-se o algoritmo *Branch and Bound*. Este consiste em um algoritmo de busca de soluções em uma árvore de decisão binária, na qual são analisados apenas os ramos cujas soluções estão em um intervalo de valores pré-determinado. Este intervalo tem como limite inferior a solução do Algoritmo Húngaro (solução de menor custo, mas não necessariamente com o número de veículos solicitado) e como limite superior a solução encontrada heuristicamente (solução viável que contém o número de veículos desejado, mas não necessariamente apresenta o menor custo).

A etapa final consistiu na realização de testes e análises dos resultados. Os testes foram realizados em máquina com a seguinte configuração: Processador Intel(R) Core(TM) i3 – 2370M com 2.40 GHz e 4GB de memória RAM.

O projeto do presente trabalho foi iniciado no mês de Maio de 2012. Durante o período de desenvolvimento foram realizadas reuniões semanais com a professora orientadora, as quais eram utilizadas para discussão dos assuntos, esclarecimento de dúvidas, desenvolvimento de pseudocódigos, planejamento das atividades, entre outros temas.

2 CONTEXTUALIZAÇÃO TEÓRICA

2.1 LOGÍSTICA

De acordo com Tacla (2003), apesar de a guerra trazer à sociedade muitos malefícios e prejuízos, a mesma exige a aceleração do desenvolvimento de algumas áreas e tecnologias, como ocorreu com a logística. Durante as operações militares da Segunda Guerra Mundial, a chamada “equipe de logística” era responsável pelo suporte às tropas que se deslocariam para a batalha, devendo providenciar munição, alimentos, equipamentos, suprimentos médicos, etc. Com o fim da guerra, uma analogia entre as necessidades da tropa e as de um processo produtivo permitiu uma expansão dos estudos e aplicação da área, que vem evoluindo a cada ano.

Para Novaes (2004), atualmente a definição mais aceita é a dada pelo CLM – *Council os Logistics Management*, o Concelho Norte-Americano de Logística, de 1998:

> “Logística é a parcela do processo da cadeia de suprimentos que planeja, implanta e controla de forma eficiente e eficaz, o fluxo de matérias-primas do estoque em processo, produtos acabados e informações relacionadas, desde seu ponto de origem até o ponto de consumo, com o propósito de atender aos requisitos dos clientes/consumidores.”

Segundo Novaes (2004), a logística moderna busca incorporar: prazos previamente acertados e cumpridos integralmente durante toda a cadeia de suprimento; integração entre todos os setores da empresa; integração com fornecedores e clientes por meio de parcerias; busca da otimização global (envolvendo racionalização dos processos e redução de custos) e satisfação do cliente. Para o autor, apesar de o conceito de “logística” estar bastante difundido atualmente, ainda se percebe uma lacuna pouco explorada quando se trata de métodos quantitativos utilizados para auxiliar na gestão de suas operações. Tais métodos são fundamentais pois permitem o racionamento no uso de recursos, aumento de produtividade, otimização do sistema e, consequentemente, redução de custos.

Segundo Chopra e Meindl (2011), o transporte refere-se ao movimento de um produto de um local para outro enquanto prossegue do início de uma cadeia até o cliente final. Assim, concentra a maior parte de custos logísticos e influencia diretamente no sucesso de toda a cadeia de suprimentos. Do ponto de vista financeiro, Ballou (2001) afirma que o transporte normalmente é o elemento mais importante em termos de custos logísticos para inúmeras empresas, sendo que a movimentação de cargas absorve de um a dois terços dos custos logísticos totais.

Para o autor, um serviço de transporte incorre em uma série de custos, tais como: mão de obra, combustível, manutenção, terminais de carga e descarga, rodovias e administrativos. Por tais motivos, deve-se buscar realizar o transporte com o máximo de eficiência.

2.1.1 Gestão de Frotas

Para Novaes (2008) o termo “gestão de frotas” representa a atividade de reger, administrar ou gerenciar um conjunto de veículos pertencentes a uma mesma empresa. Esta tarefa tem ampla abrangência e envolve serviços como: dimensionamento, especificação de equipamentos, roteirização, custos, manutenção e renovação de veículos, entre outros. Além disso, exige a constante absorção de novas tecnologias que permitam melhorar a eficiência e nível de serviços oferecidos. O autor classifica as empresas transportadoras da seguinte forma: autônomos, empresas transportadoras de carga (ETC), transportadores de carga própria (TCP) e empresas locadoras de veículos. Segundo ele, os autônomos detêm boa parte dos caminhões em circulação no Brasil e exercem papel de fundamental importância econômica. Contudo, constituem a parte mais frágil do sistema, carecendo de planejamento operacional e apoio governamental.

As ETC, apesar de mais organizadas e estruturadas, sofrem também muitas dificuldades, principalmente devido à condição inadequada das estradas, falta de planejamento para o setor por parte dos órgãos governamentais e devido a aspectos operacionais. As transportadoras de carga própria, por sua vez, têm operações semelhantes às ETC, uma vez que realizam: manutenção, treinamento de recursos humanos, renovação de frotas e equipamentos, especificação de veículos, dimensionamento e controle de frotas, entre outros. Além disso, as TCP diferenciamse por aspectos operacionais como: receitas, *marketing*, clientela, logística, etc. As empresas locadoras de veículos, por sua vez, dispõem de frotas de caminhões para locação. Elas são utilizadas para suprir, em diversos casos, as necessidades esporádicas ou sistemáticas por veículos, facilitando a gestão de frotas de outras empresas.

Diante de tais empresas com diferentes necessidades, verifica-se uma carência de ferramentas ou sistemas computacionais capazes de, a um custo acessível, ajudar as transportadoras a planejar e a executar suas operações. Por outro lado, verifica-se que os avanços na área de informática, telecomunicação, sensoriamento remoto, etc. são relativamente recentes e estão sendo absorvidos lentamente pelos transportadores. Estes, por falta de conhecimento, muitas vezes não utilizam nenhum tipo de ferramenta que auxilie na sua gestão de frotas. Em outros casos, existe insegurança ou resistência para incluir alterações em uma sistemática de trabalho que vem sendo adotada, com resultados, há certo tempo. De qualquer modo, sabe-se que o gerenciamento de frotas em tais empresas possui naturalmente elevada complexidade devido ao número de variáveis envolvidas.

2.2 PESQUISA OPERACIONAL

A Pesquisa Operacional (PO) é a ciência que trata de problemas de decisão por meio de modelos matemáticos que visam representar problemas reais (ARENALES *et al.,* 2007). Segundo Hillier (2006), a PO envolve a “pesquisa sobre operações”, sendo aplicada a problemas das mais distintas áreas, como: manufatura, transportes, construção, telecomunicações, planejamento financeiro, assistência médica, militar, serviços públicos e outros.

Segundo Arelanes *et al.* (2007), a origem da PO se deu no início das atividades militares da Segunda Guerra Mundial. Em 1934, ocorreu na Inglaterra a invenção do radar, e o Ministério da Aviação criou uma estação de trabalho para estudar como esta tecnologia poderia ser utilizada para interceptar aviões inimigos. Em 1941, foi inaugurada a Seção de Pesquisa Operacional da Força Aérea de Combate com equipes envolvidas em problemas de operações de guerra, como: manutenção e inspeção de aviões, escolha do tipo de avião para uma missão, melhoria na probabilidade de destruição de submarinos, dimensionamento de comboios de frotas, entre outros.

Com o fim da guerra, a área continuou evoluindo na Inglaterra e EUA. O rápido crescimento ocorrido desde então é atribuído por Arenales *et al.* (2007) a dois fatores: o aprimoramento técnico resultante das intensas pesquisas realizadas durante a guerra e o desenvolvimento dos computadores, que passaram a ter maior capacidade de processamento para resolver problemas complexos.

Hillier (2006), Arelanes *et al*. (2007) e Lachtermacher (2009) apresentam em seus trabalhos algumas fases similares para a resolução de problemas através da PO. Uma adaptação de tais modelos pode ser resumida pela Figura 6. Como se trata de um processo cíclico, pode-se retroceder a níveis anteriores quando algum problema for detectado (LACHTERMACHER, 2009).

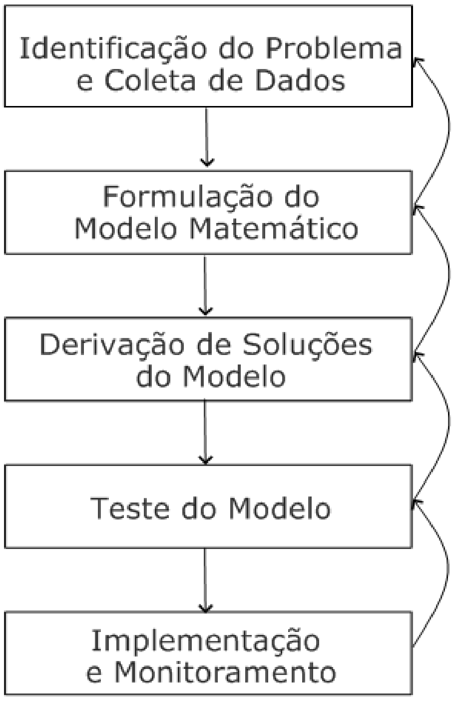


Figura 6: Fases do Processo de Resolução de Problemas de PO.
Fonte: Adaptado de Hillier (2006), Arelanes *et al*. (2007) e Lachtermacher (2009).

Para os autores, a primeira etapa é a mais crítica, uma vez que sua má definição acarretará em falhas até o final do processo. Segundo Hillier (2006), ela deve deixar claro quais são os objetivos, restrições, qual a relação entre a área a ser estudada e as outras áreas da organização, quais os possíveis limites alternativos, limites de tempo para tomada de decisão, etc.

Para Arelanes *et al*. (2007), a fase de formulação do modelo matemático consiste na formulação do problema em uma forma conveniente para análise. É comum que esta etapa exija simplificações do problema real para que seja tratável pelos métodos de resolução conhecidos. Entretanto, é necessário que tais simplificações não afetem a captação dos elementos essenciais do problema. Segundo o autor, a qualidade da solução depende da precisão com o que o modelo representa o problema e da qualidade dos dados.

Na fase de derivação de soluções do modelo objetiva-se encontrar uma solução para o modelo construído. Usualmente, desenvolve-se um procedimento computacional para tal, devido à necessidade de grande volume de processamento. Na maioria dos casos a solução buscada é aquela chamada ótima, que consiste na melhor solução para o modelo. Entretanto, deve-se analisar a viabilidade da busca por tal solução ótima. Quando o custo ou o tempo necessários para encontrá-la são demasiadamente grandes, são usados procedimentos heurísticos que fornecem soluções aproximadas (HILLIER, 2006).

O processo de teste do modelo visa aumentar a validade do modelo por meio da verificação de compatibilidade entre as soluções apresentadas e a realidade. Para Arelanes *et al.* (2007), deve-se verificar o resultado de soluções alternativas, ou seja, aquelas em que alteram-se dados do problema como: demanda, custos, etc.

Por fim, a fase de implementação e monitoramento consiste de uma fase crítica que visa garantir que a solução do modelo seja traduzida precisamente em um procedimento operacional, retificando quaisquer falhas por meio do monitoramento. Arelanes *et al.* (2007) salientam que os modelos apoiam os tomadores de decisão mas não os substituem, uma vez que, em geral, diversos outros fatores não quantificáveis também devem ser levados em consideração para a decisão final.

Segundo Goldbarg (2005), o processo de modelagem matemática utilizado no processo de resolução de problemas sofre poucas variações, entretanto as técnicas de solução de tais modelos costumam variar bastante. Algumas destas técnicas são: Programação linear, programação linear inteira, programação em redes e programação não linear. Este trabalho é voltado para um modelo matemático de programação linear inteira.

2.3 PROGAMAÇÃO LINEAR INTEIRA

Segundo Hillier (2006) em um modelo de programação linear (PL), é dado um conjunto de variáveis, chamadas de variáveis de decisão, às quais objetiva-se atribuir valores reais de modo que satisfaçam um conjunto de equações e/ou inequações lineares (restrições), visando otimizar determinada função objetivo linear. Neste sentido, o termo otimizar é utilizado para representar as possibilidades de maximizar ou minimizar a função objetivo.

Para o autor, embora a alocação de recursos para atividades seja o tipo de problema mais comum, qualquer problema cujo modelo matemático se encaixe no formato genérico para o modelo de programação linear é um problema deste tipo. Assim, a abrangência de aplicações e o desenvolvimento da computação auxiliaram na classificação da PL como um dos mais importantes avanços científicos do século XX.

Há exemplos de problemas que podem ser formulados como um problema de otimização linear nas mais variadas áreas. Alguns deles são: escala de funcionários, caminho mais curto, árvore de custo mínimo, fluxo de caixa multiperíodo, problema da mistura, produção e estoque, escolha da carteira de investimentos, composição de: rações, ligas metálicas e areias para filtros, *mix* de produção, caixeiro viajante, designação, etc (HILLIER, 2006; ARELANES *et al.,* 2007, LACHTERMACHER, 2009).

Segundo Goldbarg (2005), todo modelo de programação linear deve possuir as seguintes características:

- Proporcionalidade: a quantidade do recurso consumida por uma dada atividade deve ser proporcional ao nível dessa atividade na solução final do problema, bem como ao custo desta.
- Não negatividade: o desenvolvimento de atividades e o consumo de recursos devem ocorrer sempre em níveis não negativos.
- Aditividade: o custo total consiste na soma das parcelas associadas a cada atividade.
- Separabilidade: pode-se identificar de forma separada o custo (ou consumo de recurso) específico das operações de cada atividade.

A programação linear inteira (PLI) herda todas as características mencionadas da programação linear, porém com a restrição de que algumas ou todas as variáveis devem ter valores inteiros. Define-se o problema como programação inteira mista quando apenas algumas variáveis devem ser inteiras e programação inteira pura quando a restrição é para todas as variáveis (TAHA, 2007).

A solução ótima de problemas de PL pode ser obtida por um método iterativo baseado em ferramentas da álgebra linear, chamado de algoritmo *simplex*. Segundo Goldbarg (2005), o algoritmo é simples e eficiente. A sua ideia geral consiste em partir de uma solução viável do sistema de equações que constituem as restrições do problema e utilizar critérios de escolha para identificar novas soluções viáveis, determinando se estas são ou não uma solução ótima.

O presente trabalho embasou-se nos conhecidos problemas de programação linear do Caixeiro Viajante e da Designação para resolução da otimização do escalonamento de veículos para atendimento de pedidos pré-estabelecidos.

2.3.1 Problema da Designação

Um tipo real e comum de problema de programação linear é o conhecido problema de transporte. Segundo Lachtermacher (2009), o problema de transporte básico é aquele em que objetivo é determinar a maneira de distribuir um produto que resultará no menor custo de transporte entre as fábricas e os centros de distribuição. Uma adaptação do problema geral representado em uma rede por Taha (2007) é mostrada na Figura 7. Nela, há m origens e n destinos, cada um representado por um nó. Os arcos representam as rotas que ligam as origens aos destinos. O arco (i, j), que liga a origem i ao destino j, contém as informações de custo de transporte por unidade (c_{ij}) e quantidade enviada (x_{ij}). A quantidade de oferta na origem i é a_i e a quantidade de demanda no destino j é b_j. O objetivo do problema é determinar as incógnitas x_{ij} que minimizarão o custo total de transporte, ao mesmo tempo em que satisfarão todas as restrições de suprimentos e demanda.

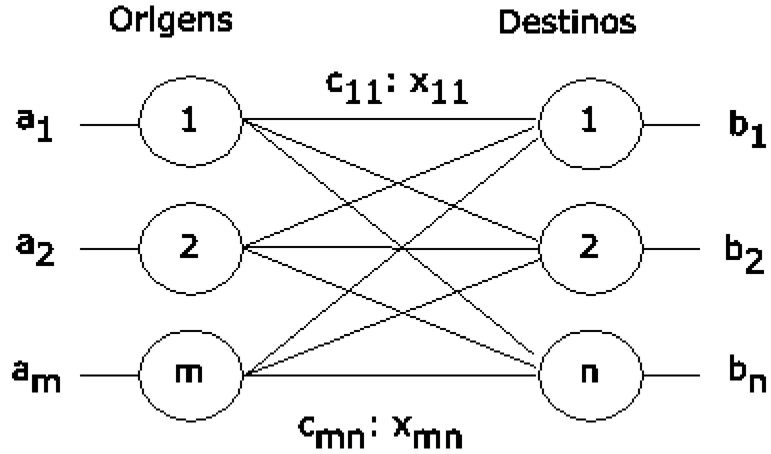


Figura 7: Representação do problema de transporte com nós e arcos
Fonte: Adaptado de Taha (2007)

O Problema da Designação , também conhecido como problema da alocação ou atribuição, é um dos casos mais importantes do problema de transporte, diferenciando-se da formulação original deste por possuir ofertas e demandas unitárias. Para Hillier (2006), o Problema da Designação consiste em um tipo especial de problema de programação linear em que os designados são indicados para a realização de tarefas. Assim, o problema geral de designação com *n* trabalhadores e *n* tarefas é representado na Figura 8. Nela, o elemento c_{ij} representa o custo de designar o trabalhador *i* à tarefa *j* (*i*, *j* = 1, 2, ..., *n*).

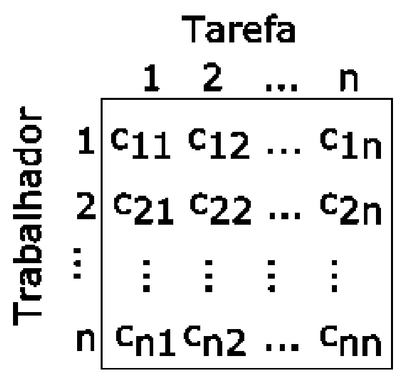


Figura 8: Problema geral da designação
Fonte: Adaptado de Taha (2007)

Entretanto, segundo Hillier (2006), as designações não precisam necessariamente ser de trabalhadores a tarefas. Há casos de designação de empregados, máquinas, veículos, fábricas ou períodos de tempo a serem destinados a recursos. Em qualquer um destes casos, para que o problema seja definido como um Problema da Designação é necessário que satisfaça às seguintes hipóteses:

- O número de designados e tarefas é o mesmo;
- Deve-se atribuir a cada designado exatamente uma tarefa;
- Cada tarefa deve ser realizada exatamente por um designado;
- Há um custo associado ao designado i (i = 1, 2, ... n) executando a tarefa j (j = 1, 2, ... n) denotado por c_{ij};
- O objetivo é determinar como todas as n designações devem ser feitas para minimizar o custo total.

Há casos de aplicações que não satisfazem completamente todas as hipóteses. Nestes, é possível reformular o problema utilizando recursos como a criação de “designados fantasmas” ou “tarefas fantasmas” com custos inviáveis para designação (HILLIER, 2006).

O modelo matemático para o Problema da Designação , dado por Arenales *et al.* (2011) é:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{1}$$

Sujeito a:

$$\sum_{i=1}^{n} x_{ij} = 1, j = 1, 2, \ldots, n \tag{2}$$

$$\sum_{j=1}^{n} x_{ij} = 1, i = 1, 2, \ldots, n \tag{3}$$

$$x_{ij} \in \{0, 1\} \tag{4}$$

A função objetivo (1) minimiza o custo total de designação de tarefas a agentes. As restrições (2) e (3) garantem que cada tarefa j é executada por um único agente, e que cada agente i executa exatamente uma tarefa. A restrição (4) indica que as variáveis são do tipo binário (as variáveis aceitam apenas os valores 0 ou 1).

Qualquer problema que satisfaça todas essas hipóteses pode ser resolvido com eficiência por algoritmos desenhados especificamente para o Problema da Designação (HILLIER, 2006). O fato de todas as quantidades fornecidas e demandadas serem sempre unitárias levou ao desenvolvimento de um algoritmo de solução simples denominado Algoritmo Húngaro (TAHA, 2007), que será apresentado na seção 2.4.

2.3.2 Problema do Caixeiro Viajante

O Problema do Caixeiro Viajante (PCV) consiste em um problema que visa encontrar uma rota que inicie em uma cidade, percorra todas as cidades de um conjunto *n* exatamente uma vez, e retorne à cidade inicial, dispendendo para tal uma distância total mínima. A rota resultante é denominada ciclo Hamiltoniano de custo mínimo (MURTY, 2008).

De acordo com Goldbard (2005), a origem do ciclo hamiltoniano é datada de 1857, ano em que Willian Rowan Hamilton propôs o jogo *Around the World*. Nele, havia um dodecaedro em que cada vértice estava associado com uma cidade importante da época. O desafio do jogo era encontrar uma rota através dos vértices do dodecaedro que iniciasse e terminasse em uma mesma cidade sem nunca repetir uma visita. Em homenagem a Hamilton, uma solução do seu jogo passou a ser denominada de ciclo hamiltoniano.

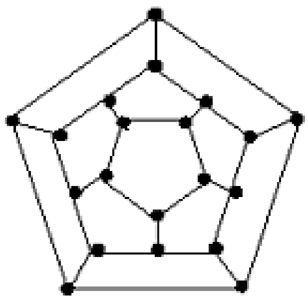


Figura 9: Dodecaedro do jogo de Hamilton
Fonte: GANHOTO (2003)

Atualmente, o Problema do Caixeiro Viajante se estende além da definição clássica de visitas a cidades, sendo aplicado a problemas de programação de operações de máquinas de manufatura, problemas de roteamento de veículos, cortes em chapas de aço e vidro, testes citológicos, fabricação de circuitos integrados, entre outros (GOLDBARG, 2005).

Além da larga aplicabilidade, observa-se que, na literatura, há um elevado número de trabalhos que buscam obter melhores resultados na resolução do PCV devido ao fato de que este é um problema *NP*-Completo.

Segundo Goldbarg (2005), existem várias formulações para o PCV, sendo muitas delas consideradas canônicas, tanto pela vasta difusão como pela forma peculiar como caracterizam o problema. Dentre elas, destaca-se na literatura a formulação de Dantizig-Fulkerson-Johnson (DFJ), como segue:

Objetiva-se:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} \tag{5}$$

Sujeito a:

$$\sum_{j=1}^{n} x_{ij} = 1, i = 1, 2, \ldots, n \qquad \forall i \in N \tag{6}$$

$$\sum_{i=1}^{n} x_{ij} = 1, j = 1, 2, \ldots, n \qquad \forall j \in N \tag{7}$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1 \qquad \forall S \subset N \tag{8}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i, j \in N \tag{9}$$

Aonde a variável binária x_{ij} assume valor 1, se o arco $(i, j) \in A$ for escolhido para integrar a solução, e O caso contrário. S é um subgrafo de $G = (N, A)$, em que $|S|$ representa o número de vértices desse subcircuito.

Observa-se que a função objetivo (5) e as restrições (6), (7) e (9) definem um modelo de designação normal. Já a restrição (8) determina a eliminação dos subcircuitos. Para cada subcircuitos possível é necessário uma restrição do tipo (8), gerando-se assim o número de $O(2^n)$ restrições, destacando a natureza combinatória do problema.

Para exemplificar, consideram-se 5 cidades ligadas entre si por rotas de duas vias, representadas pelos arcos:

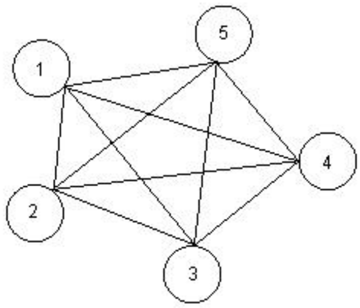


Figura 10: Exemplo de problema de designação com 5 cidades
Fonte: Adaptado de Taha (2008)

Se a solução ótima do modelo da designação produzir um circuito, como o da Figura 11, então esta também será uma solução para o PCV. Caso contrário, se a solução da designação for a mostrada na Figura 12, então a restrição (8) deverá ser considerada e outro procedimento deverá ser executado para eliminar os subcircuitos. Usualmente, tal procedimento é o algoritmo *Branch and Bound* (B&B). Nele, são impostas restrições até que todos os subcircuitos sejam eliminados. O algoritmo B&B será detalhado na seção 2.5.

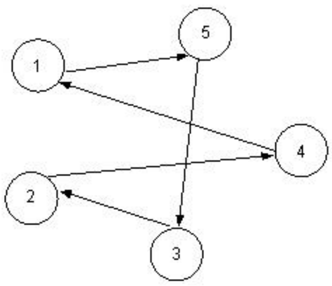


Figura 11: Exemplo de solução da designação com circuito
Fonte: Adaptado de Taha (2008)

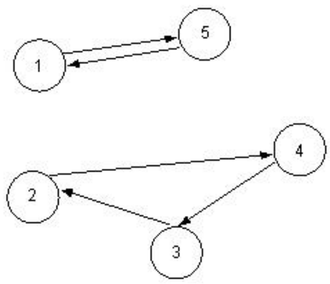


Figura 12: Exemplo de solução da designação com subcircuitos
Fonte: Adaptado de Taha (2008)

Outra observação pode ser feita a respeito do exemplo da Figura 10. Nela, considerou-se que as cidades estão ligadas entre si por rotas de duas vias. Isso significa que o mesmo arco pode ser utilizado para representar o trajeto da cidade 1 para a cidade 2 e da cidade 2 para a cidade 1, por exemplo. Casos em que é possível afirmar que $c_{ij} = c_{ji}$ para todo *i* e todo *j*, são chamados de problemas simétricos. Tais problemas são comuns em áreas como: manufatura de circuitos, programação da produção, telecomunicação e sequenciamento de DNA (SOUSA, 2008).

Quando se verifica que a matriz de custos apresenta $c_{ij} \neq c_{ji}$, denomina-se o problema como Caixeiro Viajante Assimétrico, como é o caso do problema a ser tratado no presente trabalho de conclusão de curso, já que os custos do veículo carregado e vazio são diferentes.

Goldbarg (2005) descreve ainda outros seguintes problemas correlatos: PCV com janelas de tempo, PCV com múltiplos caixeiros viajantes, PCV com coleta, PCV com bônus, PCV seletivo, PCV estocásticos, PCV com custos dependentes do tempo, entre outros.

2.4 ALGORITMO HÚNGARO

O Método Húngaro é um método de otimização utilizado para resolução de diversos problemas práticos de alocação de tarefas. Com ele, encontra-se de forma eficiente uma solução ótima para o Problema da Designação, sem precisar fazer uma comparação direta de cada opção (RENDER, 2010). Para a resolução do Algoritmo Húngaro é necessária uma matriz com os custos associado à designação de uma tarefa a um recurso. Trata-se de uma matriz quadrada *n* x *n*, em que cada elemento c_{ij} consiste no custo de atribuir a tarefa i ao agente j.

Silva *et al.*(1998) e Render(2010) definem as principais etapas do Método Húngaro por:

- Etapa 1: Em cada linha, subtrair o menor elemento dela de todos os números da mesma. Fazer o mesmo com as colunas. Assim, em cada linha e cada coluna haverá no mínimo um elemento nulo.
- Etapa 2: Testar a tabela resultante da Etapa 1 para verificar se uma atribuição ótima pode ser feita. Para tal, traça-se o número mínimo de retas horizontais ou verticais necessárias para cobrir todos os zeros na tabela. Se o número de retas for igual ao número de linhas ou colunas na

tabela, uma atribuição ótima pode ser realizada. Se o número de retas for menor que o número de linhas ou colunas, deve-se prosseguir para a Etapa 3.

- Etapa 3: Deve-se subtrair o menor número não coberto de todos os outros números não cobertos. Esse menor número é também adicionado a todos os números que estão na interseção das retas horizontais e verticais. Depois, retorna-se à Etapa 2 e repete-se o ciclo até que uma atribuição ótima seja possível.

Um exemplo de aplicação do Algoritmo Húngaro é dado quando se tem uma matriz de custos como a da Tabela 6 (conjunto de j = 1, 2, 3, 4 origens que devem ser alcançadas a partir de i = 1, 2, 3, 4 destinos) e objetiva-se minimizar o somatório dos custos c_{ij}, 1 ≤ *i*, *j* ≤ 4, para que cada origem seja alcançada por exatamente um destino e que cada destino alcance exatamente uma origem.

Tabela 6 - Exemplo de matriz inicial para Algoritmo Húngaro

	D1	**D2**	**D3**	**D4**
O1	10	20	15	16
O2	14	12	13	18
O3	10	16	19	15
O4	14	12	13	15

Fonte: Adaptado de Silva (1998)

Após a subtração do menor elemento de cada linha obtém-se a Tabela 7:

Tabela 7 - Matriz após subtrair menor elemento de cada linha

	D1	**D2**	**D3**	**D4**
O1	0	10	5	6
O2	2	0	1	6
O3	0	6	9	5
O4	2	0	1	3

Fonte: Adaptado de Silva (1998)

Após a subtração do menor elemento de cada coluna obtém-se a Tabela 8:

Tabela 8 - Matriz após subtrair menor elemento de cada coluna

	D1	D2	D3	D4
O1	0	10	4	3
O2	2	0	0	3
O3	0	6	8	2
O4	2	0	0	0

Fonte: Adaptado de Silva (1998)

Após traçar número mínimo de retas para cobrir todos os zeros obtém-se a Tabela 9:

Tabela 9 - Matriz após Etapa 2 do Algoritmo Húngaro

	D1	D2	D3	D4
O1	0	10	4	3
O2	2	0	0	3
O3	0	6	8	2
O4	2	0	0	0

Fonte: Adaptado de Silva (1998)

Após a Etapa 2, verifica-se que o número de retas traçadas é menor que o número de linhas, então deverá ser executada a Etapa 3. Verifica-se que o menor elemento não coberto é o 2. Assim, a matriz ao final da Etapa 3 é:

Tabela 10 - Matriz após Etapa 3 do Algoritmo Húngaro

	D1	D2	D3	D4
O1	0	8	2	1
O2	4	0	0	3
O3	0	4	6	0
O4	4	0	0	0

Fonte: Adaptado de Silva (1998)

Com a atualização da matriz, verifica-se que o número mínimo de retas a serem traçadas para cobrir todos os zeros é 4, equivalente ao número de linhas da matriz, como mostra a Tabela 11.

Tabela 11 – Executando-se novamente a Etapa 2 do Algoritmo Húngaro

	D1	D2	D3	D4
O1	0	8	2	1
O2	4	0	0	3
O3	0	4	6	0
O4	4	0	0	0

Fonte: Adaptado de Silva (1998)

Assim, a seguinte designação ótima poderá ser feita, conforme mostra a Tabela 12:

Tabela 12 - Matriz resultante do Algoritmo Húngaro

	D1	D2	D3	D4
O1	0	8	2	1
O2	4	0	0	3
O3	0	4	6	0
O4	4	0	0	0

Fonte: Adaptado de Silva (1998)

O custo mínimo total será dado pela soma dos custos de cada designação:

$$10+12+13+15=50$$

Nota-se que a designação ótima é dada pela escolha das células de valor zero de forma de exatamente uma célula seja escolhida em cada linha e em cada coluna.

A designação ótima será:

D1 é alcançada por O1;
D2 é alcançada por O2;
D3 é alcançada por O4;
D4 é alcançada por O3.

2.5 ALGORITMO *BRANCH AND BOUND*

A classe de algoritmos do tipo *Branch and Bound* (B&B), também denominada de ramificação e avaliação progressiva, baseia-se, segundo Machado (2013), em enumerar todas as soluções viáveis de um problema em uma estrutura de árvore de decisão, percorrendo os ramos desta de forma sistemática e evitando aqueles que levam a soluções inviáveis ou piores que as já encontradas. Para Goldbarg (2005), o termo *branch* refere-se ao fato de que são efetuadas partições (ramificações) no espaço das soluções. Caso apenas a etapa de *branch* fosse seguida, para *n* variáveis binárias, 2^n ramos deveriam ser explorados. Assim, o tempo computacional para problemas de grande porte seria inviável. Desta forma, utiliza-se a estratégia de poda (*bound*), que avalia se o ramo será ou não ramificado novamente por meio da determinação de limites para os valores das soluções de cada ramo e opção, a cada iteração, pelo ramo que ofereceu a melhor solução. A busca realizada na árvore de decisão pode ser do tipo largura ou profundidade, o valor do limite superior é obtido por soluções heurísticas para o problema e o limite inferior é encontrado por soluções do problema relaxado.

Para um problema de minimização, se um ramo tiver o valor do limite inferior maior que o valor de uma solução heurística conhecida, então a solução ótima do problema não poderá estar em um descendente desse nó, e este ramo não sofrerá mais ramificações. O método deixa, então, de examinar grande parte das soluções e contribui para acelerar a obtenção da solução ótima. (MACHADO, 2013).

Para Machado (2013), dentre outros exemplos de ramificação, há aquele em que, para cada aresta $e \in E$, consideram-se dois subproblemas: um em que a aresta está na solução ($x_e = 1$) e outro em que ela não está ($x_e = 0$). Esse método é o que foi utilizado no presente trabalho.

É comum encontrar na literatura a aplicação deste método em problemas de programação linear inteira. Nestes, mesmo exigindo-se que todas as variáveis de decisões recebam respostas inteiras, o método começa com a solução do problema relaxado na qual são permitidos valores não inteiros como resultado. Se as variáveis são inteiras, essa solução é ótima também para o problema inteiro. Se as variáveis não são inteiras, a região viável é dividida, e adicionam-se restrições que limitam o valor de uma das variáveis que não era inteira, originando dois novos ramos. Com estes, são feitos testes de avaliação que escolhem um dos ramos para continuar dividindo, enquanto o outro é descartado (RENDER, 2010). Um exemplo pode ser dado para o seguinte problema de programação linear inteira:

$$\begin{aligned}&\textit{Maximizar } Z = 5x_1 + 4x_2\\&\textit{Sujeito a}:\\&\qquad x_1 + x_2 \le 5\\&\qquad 10x_1 + 6x_2 \le 45\\&\qquad x_j \in \mathbb{Z}\end{aligned}$$

A solução para o problema relaxado (que elimina a restrição $x_j \in \mathbb{Z}$) é:

Z = 23,75
x_1 = 3,75
x_2 = 1,5

Verificando-se que as duas variáveis de decisão apresentaram valores não inteiros, pode-se escolher arbitrariamente a variável x_1 para ramificar. Assim, para o valor de 3,75, criam-se dois novos ramos. No primeiro deles (ramo 1), adiciona-se ao problema relaxado a restrição: $x_1 \le 3$. Ao segundo (ramo 2), adiciona-se ao problema relaxado a restrição: $x_1 \ge 4$. O ramo 1 apresenta a seguinte solução viável (solução que atende a todas as restrições), e não será ramificado novamente:

Z = 23
x_1 = 3
x_2 = 2

O ramo 2, por sua vez, apresenta a solução:

Z = 23,33
x_1 = 4
x_2 = 0,833

Como o ramo 1 encontrou uma solução viável, ele não será ramificado. O ramo 2, por sua vez, deverá ser ramificado pois apresentou valor para Z maior que o ramo 1 (para problemas de minimização escolhe-se o Z de menor valor). Para o valor de 0,833, criam-se dois novos ramos. No primeiro deles (ramo 3), adiciona-se a restrição: $x_2 \le 0$. Ao segundo (ramo 4), adiciona-se a restrição: $x_2 \ge 1$. Desta forma, novas avaliações e ramificações serão efetuadas com o objetivo de encontrar uma solução melhor que a solução viável já encontrada. Dentre todas as soluções viáveis encontradas, aquela que apresente o maior valor (para problemas de maximização) para Z, será a solução ótima.

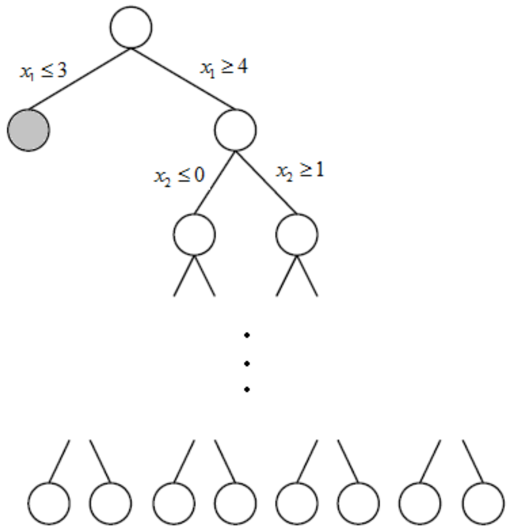


Figura 13: Exemplo de árvore de decisão do algoritmo B&B
Fonte: Adaptado de Machado (2013)

2.6 TEORIA DOS GRAFOS

Para Gusmão (2003), todo conjunto de elementos entre os quais existam relacionamentos binários de qualquer natureza pode ser considerado um grafo. Segundo Goldbarg (2000), tal conjunto visa a representação de uma estrutura de abstração. A notação matemática tipicamente emprega os símbolos G = (V,E) para representar um grafo simples G constituído de um conjunto V de vértices e um conjunto E de arestas. O número de vértices do grafo é normalmente denotado por $n = |V|$, e o de arestas por $m = |E|$. Os símbolos v_i, $1 \leq i \leq n$ designam, usualmente, os vértices, enquanto os pares (v_i, v_j) descrevem as arestas (GUSMÃO, 2003).

Os grafos estudados no desenvolvimento deste trabalho apresentaram as seguintes características:

a. Completo: Para Nascimento (2010), se houver arestas entre quaisquer dois vértices de um grafo, então este será completo.
b. Orientado: Segundo Gusmão (2003), grafos cujas arestas possuem orientação bem definida, representada por setas, são chamados de digrafos. Nestes, cada aresta é dita aresta de saída de seu vértice-origem e de entrada com relação a seu vértice-destino. Caso a orientação não seja definida, denomina-se de grafo simples. Neste, cada

aresta é simplesmente incidente a dois vértices e ambas têm a função de representar sua interdependência.

c. Ponderado: Segundo Nascimento (2010), grafos ponderados são grafos que possuem um valor associado a cada aresta.
d. Fortemente Conexo: Segundo Harary *apud* Gusmão (2003), um grafo direcionado (digrafo) é fortemente conexo se existe um caminho de *u* para *v* e também um caminho de *v* para *u,* para todo par de vértices *u, v*.
e. *Subgrafos:* Manic (2006) define que um grafo *H* é um *subgrafo* de um grafo G se $V_H \subseteq V_G$ e $E_H \subseteq E_G$.

2.6.1 Busca em Grafos

Busca em grafos é um procedimento metódico que visa explorar todos os vértices e arestas de um grafo.

Para Figueiredo e Figueiredo (2010), a busca em grafos visa descobrir se existe caminho (aresta) entre dois vértices. Para tal, evita explorar vértices já explorados, identificando-os como: *descoberto* ou *explorado*. É considerado vértice *descoberto* aquele que está sendo visitado pela primeira vez, e *explorado* aquele no qual todas as arestas incidentes já foram exploradas e vizinhos descobertos.

Existem basicamente duas formas de estabelecer a ordem de exploração de um grafo: a busca em profundidade e a busca em largura.

A busca em largura é um algoritmo simples, sem camadas, de complexidade linear (mesma ordem de grandeza do tempo necessário para ler o grafo), que percorre um grafo utilizando o princípio de uma fila, ou FIFO (*first in, first out*). Para tal, os vértices adjacentes são definidos em uma matriz de adjacência. Segundo Tenenbaum (1995) a matriz de adjacência consiste de uma matriz M de ordem $|V| \times |V|$, dado um grafo $G = (V, E)$, tal que:

$M[i, j] = 1,$ se existir aresta de i a j.

$M[i, j] = 0,$ se não existir aresta de i a j.

O seguinte pseudocódigo é proposto por Figueiredo e Figueiredo (2010) para a busca em largura:

1. Desmarcar todos os vértices;

2. Definir uma fila *Q* vazia;
3. Marcar *s* e inserir *s* na fila *Q*;
4. Enquanto *Q* não estiver vazia:
   a. Retirar *v* de *Q*;
   b. Para todo vizinho *w* de *v* faça:
      i. Se *w* não estiver marcado;
      ii. Marcar *w;*
      iii. Inserir *w* em *Q*.

Por exemplo, dada a seguinte matriz de adjacências *M[5][5]:*

Tabela 13 - Matriz de adjacências

	1	**2**	**3**	**4**	**5**
1	0	1	0	0	0
2	1	0	1	0	1
3	0	0	0	1	0
4	1	0	0	0	1
5	0	1	0	0	0

Fonte: Autoria Própria (2014)

A partir desta, o grafo da Figura 14 pode ser estruturado. Nas Figuras 15 a 19, os vértices *descobertos* são sombreados em azul e os *explorados* em verde.

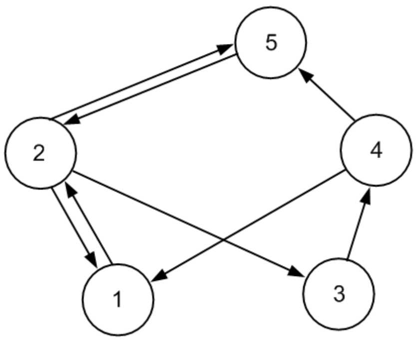


Figura 14: Exemplo de grafo para busca em largura
Fonte: Autoria Própria (2014)

Considerando o vértice 1 como a raiz, o mesmo terá todos os seus vizinhos mapeados. O único vizinho do vértice 1 é o vértice 2. Assim, o vértice 1 passa a ser do tipo *explorado* e o vértice 2 do tipo *descoberto*, como mostra a Figura 15*.*

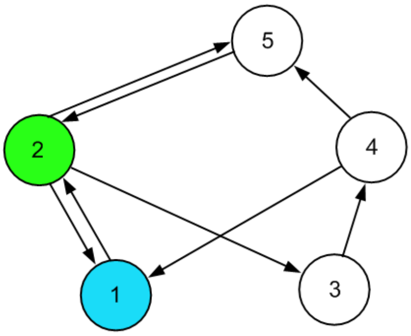


Figura 15: Exemplo de busca em largura (vértice 1 *explorado)*
Fonte: Autoria Própria (2014)

Os vizinhos do vértice 2 são os vértices: 1, 3 e 5. Como mostra a Figura 16, Após o mapeamento dos vizinhos, o vértice 2 é considerado *explorado* e o seu vizinho, vértice 3, é o novo *descoberto*, uma vez que o vértice 1 não pode ser escolhido, pois é *explorado*.

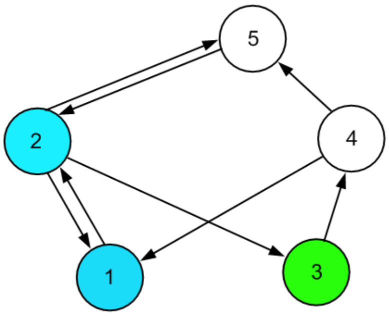


Figura 16: Exemplo de busca em largura (vértice 2 explorado)
Fonte: Autoria Própria (2014)

O vértice 3 contém apenas uma aresta que vai para o vértice 4, ainda não explorado. Sendo assim, o vértice 3 passa a ser um *explorado* e analisam-se os vizinhos do vértice 4, o novo *descoberto*. O resultado pode ser visualizado na Figura 17.

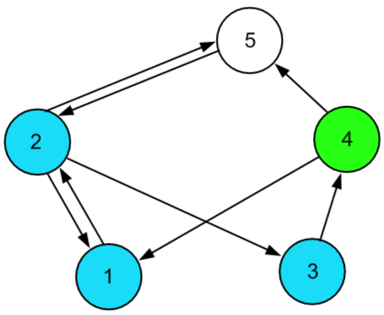


Figura 17: Exemplo de busca em largura (vértice 3 *explorado*)
Fonte: Autoria Própria (2014)

Os vizinhos do vértice 4 são o vértice 1 e 5. Como o primeiro já é um vértice *explorado*, escolhe-se o vértice 5 como *descoberto*, como mostra a Figura 18.

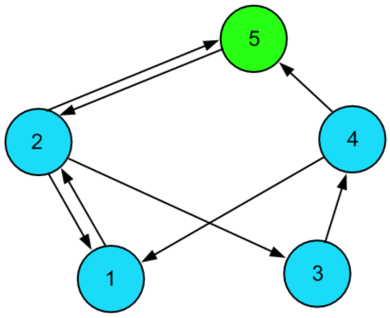


Figura 18: Exemplo de busca em largura (vértice 4 *explorado*)
Fonte: Autoria Própria (2014)

O vértice 5 tem como único vizinho o vértice 2, que já foi explorado, e nada pode ser feito. Uma vez que todos os vizinhos do vértice 5 são vértices *explorados*, este também passa a ser deste tipo e o algoritmo é finalizado, como mostra a Figura 19.

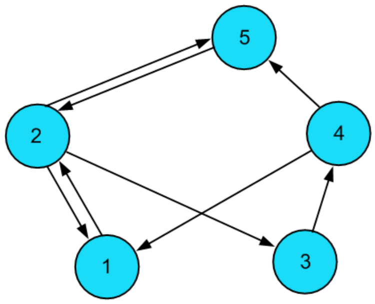


Figura 19: Exemplo de busca em largura (vértice 5 *explorado*)
Fonte: Autoria Própria (2014)

O algoritmo da busca em profundidade é bastante similar ao da busca em largura, entretanto, no percurso em profundidade, cada nó visitado é colocado em uma pilha. Assim, o último nó visitado é o primeiro nó cujos sucessores serão visitados (TENENBAUM, 1995).

3 RESOLUÇÃO DO PROBLEMA DE ESCALONAMENTO DE VEÍCULOS PARA PEDIDOS PRÉ-ESTABELECIDOS

A otimização a ser apresentada neste trabalho consiste na solução do problema de escalonamento de veículos para atendimento de pedidos pré-estabelecidos com a minimização da variável *Custo_total*.

Para tal, inicialmente definiu-se que o objetivo do trabalho seria desenvolver um algoritmo que encontrasse o custo mínimo para se designar *n* veículos a *m* pedidos. Idealizou-se que poderia ser utilizado o conhecido Problema da Designação. Este, entretanto, possui uma restrição que exige que *n* = *m*. Para o problema a ser tratado, verificou-se que, na maioria das aplicações reais, o valor de *m* seria superior ao de *n*. Para solucionar o não atendimento de tal restrição, alguns autores indicam a criação de veículos fictícios (neste caso, em que o elemento a ser atribuído é o veículo) com custos de designação consideravelmente elevados, o que faria com que os mesmos não fossem escolhidos para integrar a solução ótima. Verificou-se que tal resolução seria inviável, uma vez que, principalmente para o transporte rodoviário, muitas vezes as empresas necessitam que uma mesma carreta atenda a vários pedidos, o que faz com que *m* seja muito superior a *n*. Assim, seria necessária a construção de uma matriz de ordem elevada que dispendesse grande volume de memória computacional, sendo que a maior parte desta seria ocupada com valores fictícios cuja única função seria eliminar as designações da solução ótima.

Assim, uma nova abordagem passou a ser investigada. Verificou-se que, para um pedido *m*, havia uma origem O_m e um destino D_m, e que o percurso entre eles seria obrigatoriamente realizado. Uma vez que, após cumprir o pedido *m*, o veículo estará em D_m, a decisão a ser feita consiste em designar uma origem O_{m+1} a ser alcançada pelo veículo que está em D_m. Note que a escolha do próximo pedido a ser atendido é a única variável do problema e que é a responsável pelas diferenças no custo final.

Realizou-se então a relaxação do Problema do Caixeiro Viajante Assimétrico, transformando-o em um Problema da Designação cujo enunciado é:

<u>Enunciado 1</u>: “Dado um conjunto de *j* origens que devem ser alcançadas a partir de *i* destinos, objetiva-se minimizar o somatório dos custos c_{ij} para que cada

origem seja alcançada por exatamente um destino e que cada destino alcance exatamente uma origem”.

A formulação matemática do mesmo tem a seguinte função objetivo:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{10}$$

Sujeito às restrições:

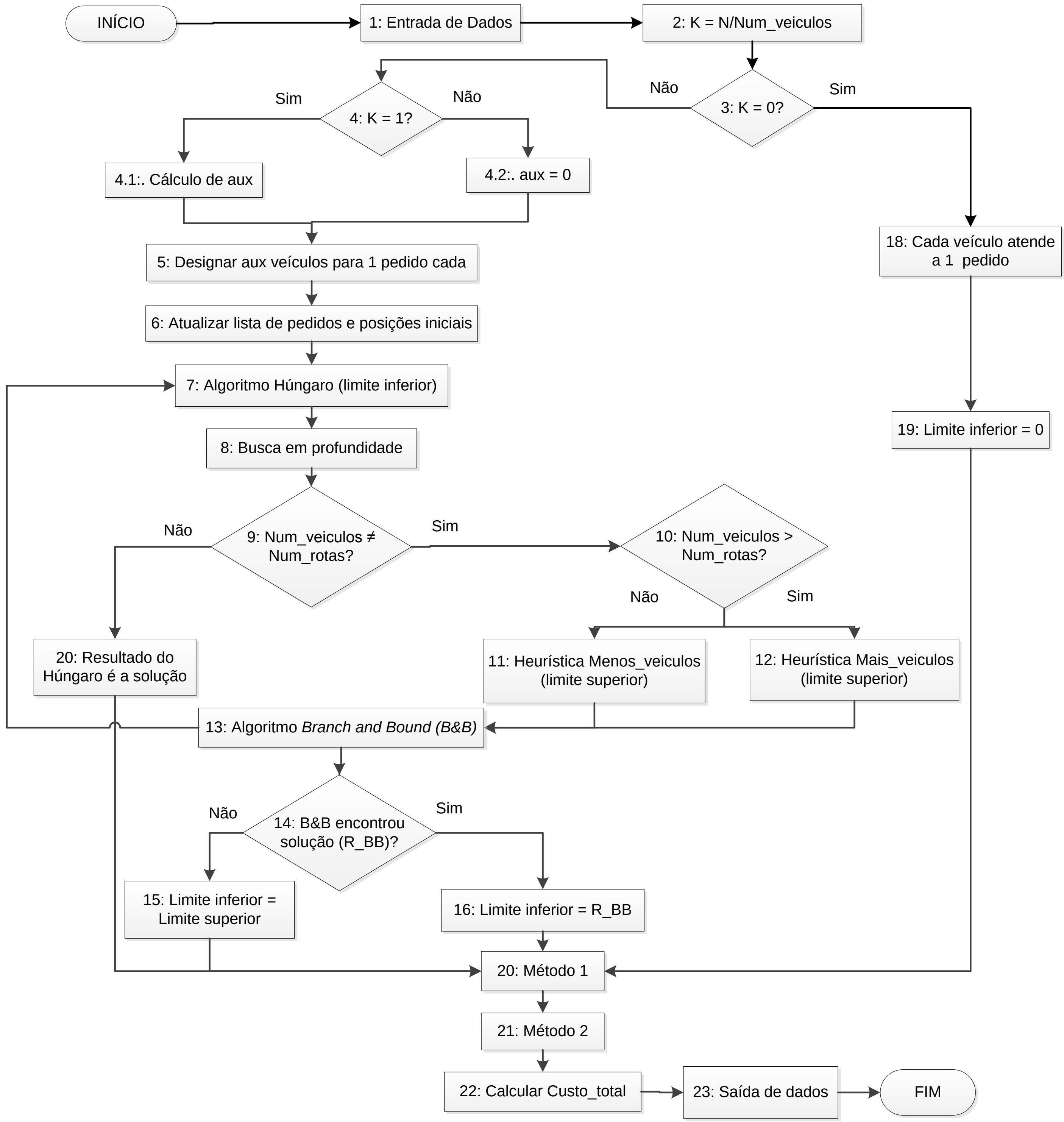
$$\sum_{i=1}^{n} x_{ij} = 1, j = 1,2, \ldots, n \tag{11}$$

$$\sum_{j=1}^{n} x_{ij} = 1, i = 1,2, \ldots, n \tag{12}$$

$$x_{ij} \in \{0,1\} \tag{13}$$

Aonde a variável binária da restrição (13) indica se a designação será realizada (recebe valor 1) ou não (recebe valor 0). A restrição (11) evita que o mesmo destino seja designado a mais de uma origem e a restrição (12) evita que a mesma origem seja alcançada por mais de um destino.

As etapas de resolução adotadas no algoritmo implementado são apresentadas no Fluxograma 1. A seguir, os passos de maior importância apresentados no Fluxograma 1 serão detalhados, para melhor compreensão do método desenvolvido.



Fluxograma 1: Etapas de resolução do algoritmo
Fonte: Autoria Própria (2014)

1. Entrada de Dados:

Sabe-se que a linguagem de programação C não contém nenhum comando de E/S, sendo todas as operações deste tipo decorrentes de chamadas a funções de bibliotecas padrão da linguagem. Tal construção torna-a altamente poderosa e flexível, permitindo-se ler e escrever qualquer tipo de dados. O sistema de E/S em C provê um nível de abstração entre o programador e o dispositivo que é utilizado para E/S. Tal abstração denomina-se *stream* e o dispositivo real é o arquivo. Um *stream* pode ser do tipo binário ou texto (SCHILDT, 1995).

O presente trabalho utilizará como entrada de dados dois arquivos do tipo texto. O primeiro deles é denominado “pedidos.txt” e tem a estrutura apresentada na Figura 20:

```
10
3 11
1 7
1 11
2 16
0 8
1 19
0 5
1 4
2 19
2 5
4
8 6 2 1
```

Figura 20 - Exemplo da entrada de dados "pedidos.txt"
Fonte: Autoria Própria (2014)

Neste, o primeiro valor apresentado (10) indica o número de pedidos a serem atendidos. A matriz de ordem 10 x 2 que inicia na próxima linha contém os 10 pares de valores que representam as origens e destinos de cada pedido. O primeiro deles, por exemplo, é o pedido cuja origem é a 3 e o destino é o 11. O valor 4, visualizado abaixo da matriz, representa o número de veículos disponíveis (*Num_veiculos*). O vetor abaixo dele é um vetor que contém a posição inicial de cada um dos veículos, correspondente a algum ponto de destino de pedidos anteriores já atendidos, onde o veículo concluiu uma rota.

A outra entrada é a “custos.txt”. A figura a seguir exemplifica o formato da entrada:

```
4 10
62 60 56 98 40 10 70 39 63 51
72 19 67 72 29 37 44 59 99 20
105 23 75 44 33 106 43 67 20 93
46 81 30 33 81 36 43 74 40 75
10 14 43 10
29 14 50 53
76 28 31 71
20 11 27 38
66 15 53 60
54 28 44 54
22 33 53 18
19 76 18 64
54 75 74 24
73 43 62 52
```

Figura 21 - Exemplo da entrada de dados "custos.txt"
Fonte: Autoria Própria (2014)

O primeiro par de valores informado nesta entrada corresponde ao número de origens (4) e de destinos (10) da transportadora. A matriz de dimensões 4 x 10 que segue possui como linhas as origens dos pedidos e como colunas os destinos destes. Assim, cada elemento corresponde ao custo para percorrer, com o veículo carregado, os trechos entre as origens e destinos dos pedidos. Esta matriz é denominada de *M_cheio.* A próxima matriz, de dimensões 10 x 4, é denominada *M_vazio* e contém os custos para percorrer, com o veículo descarregado, os trechos entre o destino de um pedido até a origem do próximo pedido, para todos os pedidos, origens e destinos.

O Quadro 1 mostra as variáveis resultantes da leitura da entrada de dados:

Descrição	Variável
Número de pedidos	*N*
Pedidos	*Pedidos [N][2]*
Número de destinos	*Num_destinos*
Número de origens	*Num_origens*
Custos de veículo cheio	*M_cheio[Num_origens][Num_destinos]*
Custos de veículo vazio	*M_vazio[Num_destinos][Num_origens]*
Número de veículos	*Num_veiculos*
Posição inicial dos veículos	*Pos_veiculos [Num_veiculos]*

Quadro 1 - Variáveis Resultantes da leitura da entrada de dados
Fonte: Autoria Própria (2014).

2. Cálculo de K:

Inicialmente, utiliza-se a aritmética de inteiros em linguagem C para calcular o valor inteiro resultante da razão entre o número de pedidos e o número de veículos disponíveis. Tal resultado é armazenado na variável *K.* Este cálculo é necessário para que se designe veículos para atender um único pedido nos casos em que não há pedidos suficientes para que cada veículo atenda pelo menos 2 pedidos.

3. Teste de K = 0?:

O primeiro teste realizado refere-se à variável *K* calculada na etapa anterior. Caso *K* seja zero, determina-se que cada veículo deverá atender a um único pedido (18) e o limite inferior será zero (19).

4. Teste de K = 1?:

Caso *K* seja igual a 1, significa que o sistema deverá obrigar alguns caminhões a atender um único pedido pois, do contrário, a heurística *Mais_veículos* terá de utilizar uma designação de custo infinito para conseguir cumprir todos os pedidos. Assim, será calculado o valor da variável *aux*, que indicará o número de veículos que atenderão a um único pedido.

5. Designar *aux* veículos para 1 pedido cada:

Caso o valor de K seja maior que 1, então nenhum veículo atenderá a apenas um pedido. Caso contrário, a variável *aux* será calculada por:

$$aux = Num\_veiculos - \left(n \,\%Num\_veiculos\right)$$

Aonde a variável *aux* corresponde ao número de veículos que atenderá ao apenas um pedido.

7. Algoritmo Húngaro (limite inferior):

O Algoritmo Húngaro foi utilizado para determinar a designação ótima de origens a destinos com uma relaxação do problema real, pois não há nenhuma restrição quanto ao número de rotas da solução ótima. No problema real o número de rotas deverá ser igual ao número de veículos.

Um exemplo de solução dada pelo Algoritmo Húngaro é representado em um grafo por meio da Figura 22, na qual as setas direcionadas das origens i para os destinos j indicam que deverá ser percorrido o trecho do destino i para a origem j. Observa-se que este grafo possui 3 componentes ou *subgrafos*. Determinou-se que cada componente encontrada no grafo da solução ótima dada pelo Algoritmo Húngaro será considerada uma rota a ser seguida por um veículo. Cada veículo percorrerá exatamente uma rota e cada rota será percorrida por exatamente um veículo.

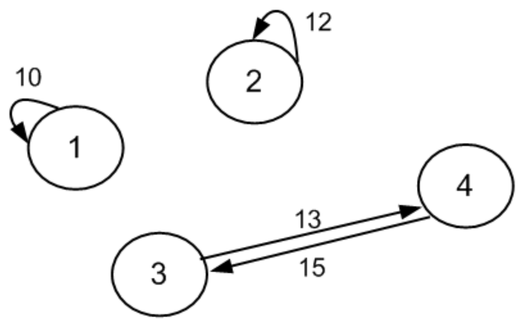


Figura 22 - Exemplo de grafo que representa a resposta do Algoritmo Húngaro
Fonte: Autoria Própria (2014)

A solução exemplificada é originária do grafo completo da Figura 23. Este grafo é do tipo completo, orientado, ponderado e fortemente conexo.

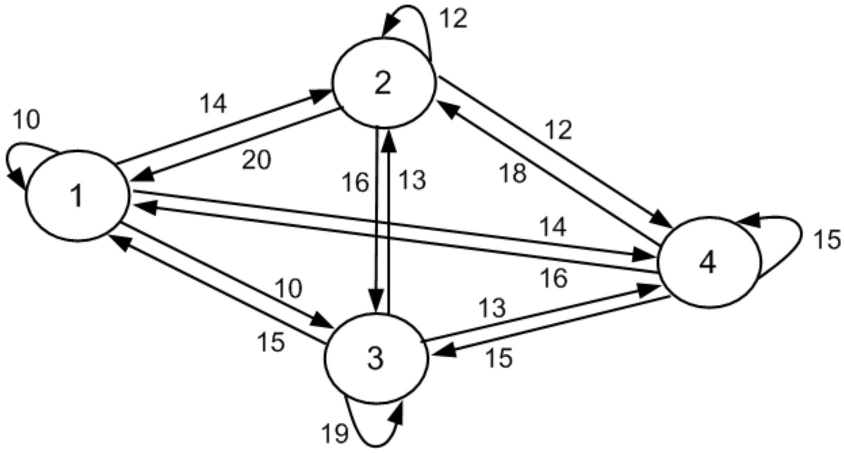


Figura 23 - Grafo que representa matriz de entrada do Algoritmo Húngaro
Fonte: Autoria Própria (2014)

A solução representada na Figura 22 define que, para o exemplo, o custo mínimo para que cada origem seja alcançada por exatamente um destino e que cada destino alcance exatamente uma origem ocorre quando são utilizados 3 veículos. O primeiro dos veículos deverá atender ao pedido 1 e depois voltar para a origem do mesmo. O segundo veículo fará o mesmo para o pedido 2. O terceiro, atenderá ao pedido 3, irá do destino do pedido 3 para a origem do pedido 4, atenderá ao pedido 4 e depois retornará à origem do pedido 3. O presente trabalho, entretanto, determina que os veículos não retornem à origem do primeiro pedido atendido, o que será tratado mais adiante.

Pode-se, então, calcular o custo variável da mesma designação por meio do somatório de custos da matriz *M [i] [Result [i]]*, com *i* variando de 1 a *N*. Este custo será utilizado como limite inferior no algoritmo *branch and bound* caso ele seja executado.

8. Busca em profundidade:

Note que o número de rotas dadas na resposta do Algoritmo Húngaro correspondente ao número de componentes fortemente conexas de um grafo.

Para que o algoritmo desenvolvido fosse capaz que reconhecer o número de componentes fortemente conexas do grafo simples resultante da execução do Algoritmo Húngaro, implementou-se um algoritmo de busca em grafos do tipo busca em profundidade utilizando um vetor de inteiros que é resultado do húngaro.

Denominou-se tal vetor de *VS[N]*, aonde *N* é o número de pedidos, como indica o Quadro 1. Para o exemplo de resultado da Figura 22, o vetor *VS* seria escrito da seguinte maneira:

VS[i]	1	2	4	3
I	1	2	3	4

Figura 24: Vetor resultante do Algoritmo Húngaro
Fonte: Autoria Própria (2014)

Ou seja, se *i* é o pedido 1, *VS [1]* é o próximo pedido a ser atendido após o pedido 1. Quando a busca selecionar o pedido 1, ela sinalizará em outro vetor denominado *visitados,* de tamanho *N,* que este pedido já foi atendido. Como na rota 1 tem apenas o pedido 1, quando *VS[1]* for analisado, será verificado que seu valor é o próprio 1, o qual já foi visitado. Assim, indica que o componente chegou ao fim. Em seguida, as variáveis *Num_rotas* e *i* são incrementadas. Caso *i* ainda seja menor ou igual a *N* e o vetor *visitados* ainda tenha alguma posição não sinalizada, o mesmo será feito para o próximo veículo. Quando a busca atingir *i* = 3, o próximo pedido é o 4 (*VS[3]* = 4). Neste caso, *i* passa a ser 4 e verifica-se qual é o valor de *VS[4]*. Como este é 3 e 3 já foi sinalizado no vetor *visitados[3]*, então a componente foi totalmente atendida. A variável *Num_rotas* será novamente incrementada e o seu valor final corresponderá ao número de rotas da designação ótima de origens a destinos.

9. Teste: Num_veículos ≠ Num_rotas?

Caso o número de veículos seja igual ao número de rotas dado pelo Algoritmo Húngaro significa que a solução deste é a solução ótima. Caso contrário, deverá ser verificada a relação entre eles.

10. Teste: Num_veículos > Num_rotas?

Caso o número de veículos seja maior que o número de rotas do Húngaro, a heurística chamada de *Mais_veiculos* será executada, caso contrário, será executada a heurística *Menos_veiculos*. A função de ambas é encontrar uma solução viável para o problema (de forma que *Num_veiculos* = *Num_rotas.* Não se pode afirmar que o somatório dos custos é mínimo, pois quando se altera o número de veículos da solução ótima, apresentada pelo Método Húngaro, o custo pode aumentar. Mas, se mesmo assim, o usuário deseja utilizar uma quantidade de veículos diferente da informada pelo Método Húngaro, então heurísticas e um algoritmo de *Branch and Bound* serão aplicados para que essa restrição seja atendida com custos baixos (mesmo que não mínimos). O somatório de custos dado pelas heurísticas será o limite superior do algoritmo *Branch and Bound,* uma vez que a solução ótima para o número de veículos informado possui somatório de custos menor ou igual ao da mesma.

11. Heurística *Menos_Veiculos*:

A heurística aplicada caso *Num_veiculos* < *N* e *Num_veiculos* < *Num_rotas,* consiste em eliminar quantos números de trechos de maior custo forem necessários e forçar a ligação entre as rotas visando à redução do número das mesmas. Dado o seguinte exemplo, com 3 componentes encontradas pelo Método Húngaro:

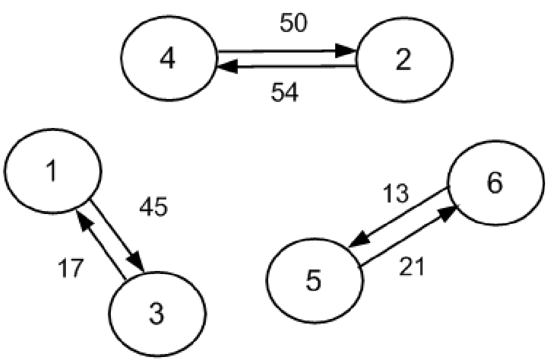


Figura 25 – Exemplo de componentes para heurística *Menos_Veiculos*
Fonte: Autoria Própria (2014)

Supondo-se que para o caso acima o número de veículos disponíveis é 2 (*Num_veiculos* = 2), os componentes do grafo da Figura 25 devem ser unidos até que o grafo final tenha apenas 2 componentes.

Para tal, utilizou-se novamente a busca pelo vetor *VS.* O vetor que representa a solução é:

VS [i]	3	4	1	2	6	5
I	1	2	3	4	5	6

Figura 26: Exemplo de vetor inicial para *Menos_Veiculos*
Fonte: Autoria Própria (2014)

Criou-se uma variável denominada *custo* que é inicializada com o valor da matriz de custos *M[0][VS[0]],* que é o elemento *M[0][3]* para o exemplo. A variável *prox* foi inicializada com *VS[0]* = 3 e outra variável *e1,* que armazena o maior custo do primeiro componente percorrido, foi inicializada com zero. A busca irá percorrer todo o componente ao qual pertence o pedido 3 (apenas os pedidos 3 e 1) e salvar em *e1* o valor do *prox* cujo custo *M[prox][VS[prox]]* foi o maior. Quando *prox = 3, M[3][VS [3]]* = 17, já quando *prox* = 1, *M[1][VS [1]]* = 45. Assim, *e1* = 1. Todos os pedidos percorridos são sinalizados no vetor *visitados.* Em seguida, caso *i* ainda não seja igual a *N,* ele será incrementado até que se encontre um pedido ainda não visitado. Quando encontrado, é sinalizado como visitado e armazenado na variável *e2.* Atribui-se *M[i][VS[i]]* a *custo* e *VS[i]* a *prox* e, da mesma forma que para o primeiro componente, encontra-se o valor de *prox* para o qual o *M[prox][VS[prox]]* é o maior, mas agora armazena-se *prox* em *e2.* Para o exemplo, *e2 = 2.*

Tendo-se encontrado *e1* e *e2,* atribui-se *VS[e1]* a *VS[e2]* e *VS[e2]* a *VS[e1]* utilizando-se outra variável auxiliar. O novo vetor VS será:

VS [i]	4	3	1	2	6	5
i	1	2	3	4	5	6

Figura 27: Exemplo de vetor resultando para *Menos_Veiculos*
Fonte: Autoria Própria (2014)

E pode ser representado pelo grafo da

Figura 28:

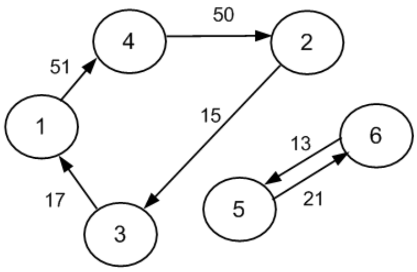


Figura 28 – Exemplo de componentes resultantes para *Menos_Veiculos*
Fonte: Autoria Própria (2014)

Em seguida, o *Num_rotas* é decrementado. O procedimento descrito será executado enquanto *Num_rotas* > *Num_veiculos.*

12. Heurística *Mais_Veículos*:

Esta heurística busca aumentar o número de rotas da solução da designação ótima até que ela seja igualada ao número de veículos disponíveis. Para tanto, o mesmo princípio lógico da busca no vetor *VS* será seguido, porém novas análises serão exigidas.

Supondo-se que o grafo abaixo é uma solução do Algoritmo Húngaro e que *Num_veículos* = 2. A heurística será aplicada para transformar o componente em dois componentes disjuntos.

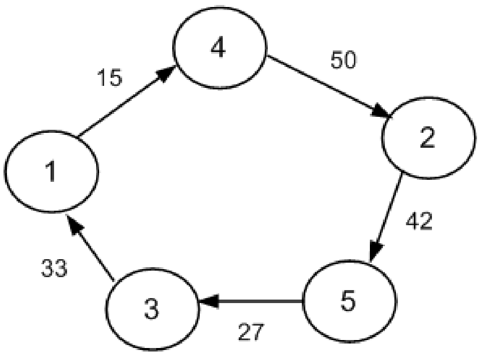


Figura 29 – Exemplo de ciclo resultante do Algoritmo Húngaro
Fonte: Autoria Própria (2014)

O vetor *VS* que representa esta solução é:

VS[i]	4	5	1	2	3
i	1	2	3	4	5

Figura 30: Exemplo de vetor inicial para *Mais_Veiculos*
Fonte: Autoria Própria (2014)

Para a variável *i* de 1 até *N e* enquanto o vetor *visitados* ainda não estiver completamente sinalizado, determina-se que a variável *start* inicialmente recebe o valor de *i.* As variáveis *ant_prox* e *prox* recebem *VS [start]* e *VS [VS [start]],* respectivamente. Para o exemplo: *start* = 1, *ant_prox* = 4 e *prox* = 2. Assim, o seguinte custo de percorrer os componentes da Figura 31 pode ser calculado por:

$$custo = M[start][prox] + M[ant\_prox][VS[start]]$$
$$-M[start][VS[start]] - M[ant\_prox][prox]$$
$$custo = M[1][2] + M[4][4] - M[1][4] - M[4][2]$$

Para o exemplo, custo = 134.

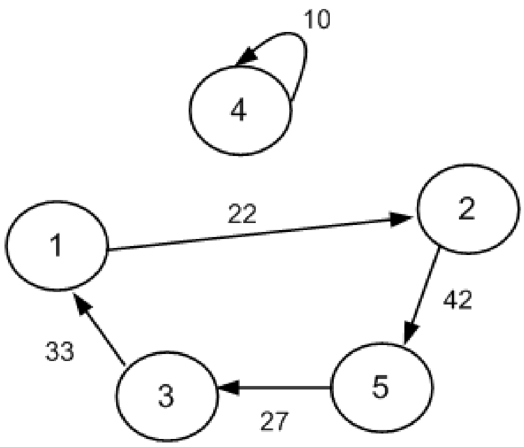


Figura 31 – Exemplo 1 de componentes resultantes da *Mais_Veiculos*
Fonte: Autoria Própria (2014)

Em seguida, ant_prox recebe o valor de prox e prox recebe o valor de VS[prox], para calcular-se novamente o custo e analisar-se se este é melhor que o(s) outro(s) custo(s) já calculado(s). Para o exemplo, o segundo e terceiro conjunto de componentes avaliado será o da

Figura **32** e Figura 33, respectivamente:

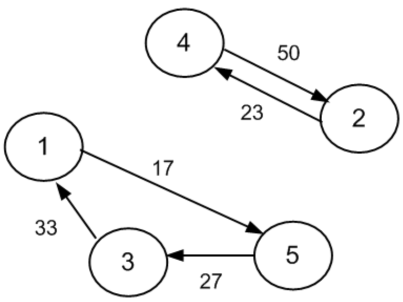


Figura 32 – Exemplo 2 de componentes resultantes da *Mais_Veiculos*
Fonte: Autoria Própria (2014)

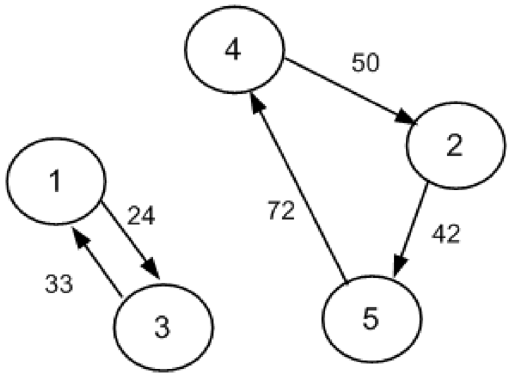


Figura 33 – Exemplo 3 de componentes resultantes da *Mais_Veiculos*
Fonte: Autoria Própria (2014)

Ao final da formação da terceira possibilidade de componentes verifica-se que *VS[prox] = start*, o que indica que todas as possibilidades com o *start* em 1 foram verificadas. Assim, o vetor *visitados* na posição *start* (1) é sinalizado e o novo *start* passa a ser *VS [start]* que, para o exemplo, é 4. O mesmo procedimento será realizado para *start* = 2, 5 e 3. Posteriormente, a solução escolhida será aquela que apresentar menor custo e o *Num_rotas* será incrementado. Todo o procedimento será executado novamente enquanto *Num_rotas != Num_veiculos.*

13. Algoritmo *Branch and Bound* (B&B):

O algoritmo *Branch and Bound* implementado neste trabalho realiza a ramificação da árvore de soluções ao escolher aleatoriamente uma aresta para ser eliminada da solução. Verifica-se qual é o comportamento do somatório de custos neste caso e determina-se se a mesma deve ser fixada para compor a solução ótima ou não. Realiza-se este procedimento para todas as arestas da solução da pelo Algoritmo Húngaro.

Inicialmente, atribui-se valor 0 aos elementos de uma matriz quadrada de ordem N denominada *e_fixo*. Esta matriz será sinalizada sempre que for forçada a existência de uma aresta na solução. O algoritmo entrará em um laço que irá executar a lógica do *branch and bound* enquanto houverem arestas a serem testadas (número de sinalizações na matriz de custos é menor que N).

Em seguida, remove-se uma das arestas da solução (atribuindo a ela custo infinito) e executa-se o Algoritmo Húngaro para se obter uma solução ótima diante do novo contexto (sem uma aresta da solução anterior). Caso esta contenha número de componentes igual ao número de veículos disponíveis informado, ela será armazenada como uma solução viável. Caso contrário, será verificado se o custo da solução não-viável ainda é menor que o custo da melhor solução viável já encontrada. Em caso afirmativo, uma nova ramificação será feita a partir desta solução, uma vez que é possível que se encontre uma solução viável de custo menor que o melhor já encontrado, o que acarreta em atualizar a melhor solução já encontrada. Entretanto, se o custo da solução não-viável for maior que o da melhor solução viável já encontrada, o ramo pode ser descartado, sem novas ramificações.

O passo seguinte consiste em reestruturar a matriz de custos colocando novamente a aresta escolhida na solução e realizando novamente os mesmos testes citados no parágrafo anterior.

O processo de retirar um trecho da solução para avaliar o comportamento da solução sem ele, reestabelecer as mudanças que causam resultados ruins, e manter aquelas que apresentam melhores resultados é repetido recursivamente até que não haja mais opções de ramos para avaliação.

14. Teste: B&B encontrou solução (R_BB)?

Ao final da execução do *Branch and Bound* será testado se o mesmo encontrou uma solução melhor do que aquela dada por uma das heurísticas ou não. Em caso afirmativo, tal resposta será atribuída ao limite inferior (15), que será a resposta. Caso contrário, a resposta receberá o valor do limite superior, que é o resultado da heurística que foi executada (16).

Após determinar-se qual é o valor do(s) ciclo(s) de custo mínimo (limite inferior), deve-se transformar tal(is) ciclo(s) em caminho(s), escolhendo-se qual será o primeiro pedido a ser atendido e eliminando o trecho que chega ao mesmo. Deve-se também determinar quais veículos atenderão a quais rotas, levando-se em consideração sua posição inicial. Tais finalizações serão realizadas por dois métodos distintos, detalhados nos itens 20 e 21, respectivamente.

20. Método 1:

Dada a matriz de custos:

Tabela 14 - Matriz de custos para exemplo de heurística para primeiro pedido

	O0	O1	O2	O3	O4	O5	O6	O7	O8	O9
D0	4	67	34	0	69	24	78	58	62	64
D1	5	4	81	27	61	91	95	42	27	36
D2	91	4	2	3	92	82	21	16	18	95
D3	47	6	71	38	69	12	7	9	35	94
D4	3	11	22	33	73	64	41	1	53	68
D5	47	2	62	7	37	59	23	41	29	78
D6	16	35	90	2	88	6	80	4	4	48
D7	46	5	90	2	70	50	6	1	93	48
D8	2	23	84	54	6	40	66	76	31	8
D9	44	39	26	23	37	38	18	82	29	41

Fonte: Autoria Própria (2014)

Para a seguinte designação ótima cujo somatório dos custos é 77:

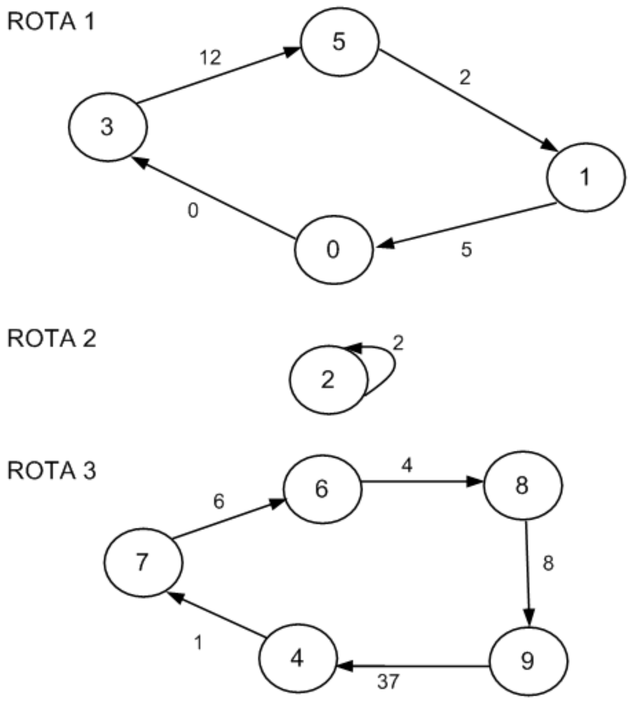


Figura 34 - Exemplo de designação para a heurística para primeiro pedido
Fonte: Autoria Própria

Considerando que a posição inicial dos veículos, armazenada no vetor *Pos_veiculos [Num_veiculos], é*:

Veículo 1 em 2;

Veículo 2 em 7;

Veículo 3 em 4.

Neste método escolhe-se o primeiro pedido a ser atendido de forma que este seja o vértice de chegada da aresta de maior custo. Esta aresta será descartada, uma vez que este trabalho considera que não ocorre o retorno para a origem da rota. Para o exemplo, a adaptação faria com que o custo total passasse a ser 25 e a rota 1 fosse iniciada em 5, a rota 2 em 2 e a rota 3 em 4, como ilustra a figura a seguir:

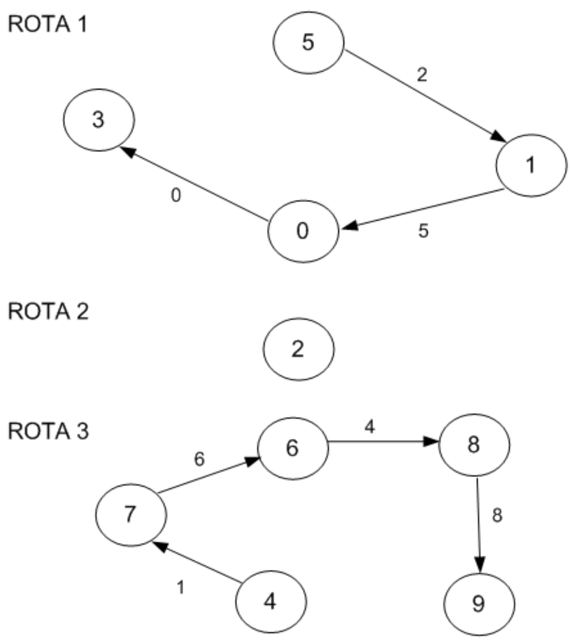


Figura 35 - Escolha do primeiro pedido pelo Método 1
Fonte: Autoria Própria (2014)

Conhecendo-se os primeiros pedidos de cada rota, um novo Problema da Designação é proposto. Neste, deve-se designar n veículos a m rotas. Cada rota deve ser atendida exatamente por um veículo e cada veículo deve ser atribuído a exatamente uma rota. Para o exemplo seguido, a matriz de custos resultante seria:

Tabela 15 - Matriz de custos para o Método 1

	5	**2**	**4**
2	82	2	92
7	50	90	70
4	64	22	73

Fonte: Autoria Própria (2014)

Nesta, os valores das linhas indicam a posição inicial dos veículos (*Pos_veiculos*) e as colunas indicam os primeiros pedidos a serem atendidos quando se elimina o trecho de maior custo em cada rota (ao posicioná-lo como retorno à origem após atendimento de todos os pedidos). Os valores são determinados pela Tabela 14.

Executando-se o mesmo Algoritmo Húngaro já descrito neste trabalho, a seguinte designação é obtida:

Veículo 1 atenderá à rota 2;

Veículo 2 atenderá à rota 1;

Veículo 3 atenderá à rota 3.

O custo de deslocamento dos veículos da sua posição inicial até a origem do primeiro pedido a ser atendido é $2+50+73=125$, enquanto o somatório de custos variáveis é 26.

21. Método 2:

Neste método, escolhe-se o primeiro pedido a ser atendido em cada rota e também qual veículo atenderá qual rota alocando-se os veículos para os pedidos cuja origem é a mais próxima possível (com deslocamento de menor custo).

Para o exemplo citado na etapa 20, a solução seria designar o veículo 1 à rota 2 (cujo custo de deslocamento seria 2 ao pedido 2), o veículo 2 à rota 3 (cujo deslocamento seria 1 ao pedido 7) e o veículo 3 à rota 1 (com custo de deslocamento 3 ao pedido 0). O somatório de custos de deslocamento é 6, enquanto o somatório de custos variáveis é 69.

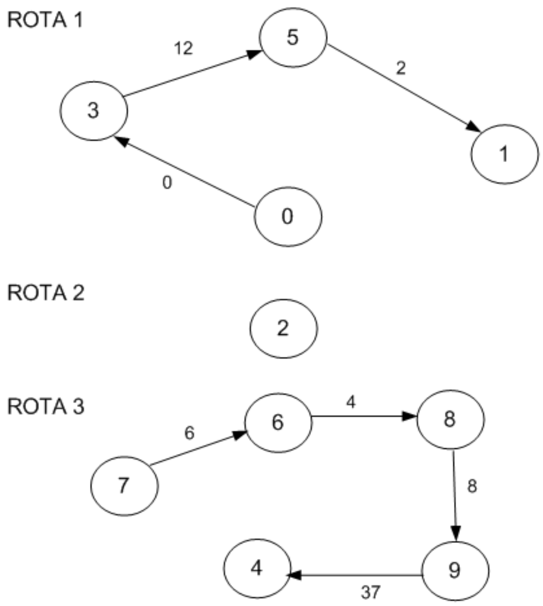


Figura 36 - Escolha do primeiro pedido pelo Método 2
Fonte: Autoria Própria (2014)

22. Calcular *Custo_total*:

O valor do *Custo_total* diferente nos métodos, uma vez que é a soma dos custos de deslocamento, varável e fixo. Os dois primeiros já foram calculados. Os custos fixo, por sua vez, são calculados pelo somatório dos elementos da matriz *M_fixo[pedido[i][0]][pedido[i][1]]* com *i* variando de 1 a *N.*

23. Saída de dados:

A última etapa consiste em escrever em um arquivo de texto o roteamento de cada veículo, bem como o custo resultante de cada método, tal como exemplifica a figura a seguir:

saida1.txt - Bloco de notas

Arquivo Editar Formatar Exibir Ajuda

```
   Ciclo do hungaro: 13143
Componentes do hungaro: 15
   Ciclo da H_Menos: 13143
   Ciclo do Branch and Bound: 13143

Tempo para solucao otima (s) 10.5760

****Primeiro Metodo****

ROTEAMENTO:
   Veiculo 1: 45 43 46 44 41 34 32 31 30 20 23 11 14 16 18 21 13 15 17 19 7 10 5 9 3 8
   12 60 56 54 52 2 62 58 55 53 51 50 49 47 48 0 64 61 57 1 63 59 6 4 22 24 26 25 29 27 28 36 33 40 38 37 35 42 39
CUSTOS FINAIS:
   Custo Fixo = 20224
   Custo Deslocamento = 289
   Custo Variavel = 12761
   TOTAL = 33274

****Segundo Metodo****

ROTEAMENTO:
   Veiculo 1: 3 8 12 60 56 54 52 2 62 58 55 53 51 50 49 47 48 0 64 61 57 1 63 59 6 4 22
   24 26 25 29 27 28 36 33 40 38 37 35 42 39 45 43 46 44 41 34 32 31 30 20 23 11 14 16 18 21 13 15 17 19 7 10 5 9
CUSTOS FINAIS:
   Custo Fixo = 20224
   Custo Deslocamento = 122
   Custo Variavel = 12783
   TOTAL = 33129
```

Figura 37: Exemplo de arquivo de saída
Fonte: Autoria Própria (2014)

4 RESULTADOS

4.1 SIMULAÇÕES

O método proposto foi inicialmente aplicado e analisado para 30 casos simulados. As entradas necessárias para tal foram geradas por um algoritmo desenvolvido na mesma linguagem, o qual recebe como entrada o número de destinos, o número de origens, o valor que será o custo máximo das matrizes de custo, o número de pedidos e o número de veículos disponíveis. Com elas, constroem-se os arquivos “custos.txt” e “pedidos.txt”, detalhados na Seção 3.

No arquivo de custos, os custos da matriz referente ao deslocamento do veículo carregado (*M_cheio*) são proporcionais aos custos de deslocamento do veículo vazio (*M_vazio)*. Os valores da segunda correspondem a 70% dos valores da primeira, sendo esta uma estimativa do valor que a carga agrega ao consumo de combustível e deterioração do veículo.

O valor do custo máximo informado é utilizado pela função *rand(),* definida na biblioteca *stdlib* da linguagem de programação C. A função irá gerar números aleatórios de zero ao valor máximo estipulado, o qual será denominado de amplitude máxima. A matriz de pedidos também utiliza esta função, porém o valor máximo na primeira coluna será o número de origens e na segunda coluna será o número de destinos. O vetor de posições iniciais é gerado da mesma forma, porém com valores entre zero e o número de destinos informado.

O Quadro 2 apresenta as características dos arquivos de testes:

(continua)

Teste	Pedidos	Veículos	Origens	Destinos	Amplitude
0	50	1	10	10	500
1	50	2	10	10	500
2	50	3	10	10	500
3	50	4	10	10	500
4	50	5	10	10	500
5	50	6	10	10	500
6	50	7	10	10	500
7	50	16	10	10	500
8	50	17	10	10	500
9	50	20	10	10	500

(conclusão)

Teste	Pedidos	Veículos	Origens	Destinos	Amplitude
10	60	5	40	40	500
11	60	5	20	20	500
12	60	5	5	50	500
13	60	5	50	5	500
14	75	2	15	15	10
15	75	2	15	15	100
16	75	2	15	15	1000
17	80	1	10	10	300
18	85	1	10	10	300
19	90	1	10	10	300
20	100	1	10	10	300
21	200	1	10	10	300
22	1000	1	10	10	300
23	400	8	10	10	300
24	200	6	10	10	300
25	150	2	10	10	300
26	500	5	10	10	300
27	600	8	10	10	300
28	1000	10	10	10	300
29	2000	10	10	10	300

Quadro 2: Características dos arquivos de testes
Fonte: Autoria própria (2014)

Os testes 0 a 9 utilizaram a mesma matriz de custos variáveis. A solução dada pelo Algoritmo Húngaro para esta matriz solicitou o uso de 5 veículos. Assim, avaliou-se o comportamento do método para a exigência do uso de 1, 2, 3, 4, 5, 6, 7, 16, 17 ou 20 veículos.

Os testes 10 a 13 utilizaram-se de matrizes aleatórias de mesmas dimensões, com mesmo número de pedidos e de veículos disponíveis, porém com quantidades de origens e destinos variando proporcionalmente e também entre si.

Nos testes 14, 15 e 16, o número de pedidos, de veículos disponíveis, de origens e de destinos foram os mesmos, enquanto o valor da amplitude máxima dos custos foi de 10, 100 e 1000, respectivamente.

Os testes 17 a 22 constituem casos do Problema do Caixeiro Viajante Assimétrico, uma vez que há apenas um veículo disponível, o que acarreta na determinação de exatamente uma rota que atenda a todos os pedidos. O número de pedidos para este conjunto de testes variou de 80 a 1000.

Por fim, os testes 23 a 29 apresentaram casos de amplitude, número de origens e número de destinos fixos, bem como valores de pedidos entre 150 a 2000 e quantidade de veículos disponíveis variável.

Vale lembrar que os teste foram realizados em um computador de configuração: Processador: Intel(R) Core(TM) i3 – 2370M, CPU: 2.40 GHz e RAM: 4GB.

Os resultados obtidos são apresentados no Quadro 3. Neste, a coluna *T_BB (s)* apresenta o tempo em que o algoritmo foi finalizado, em segundos. Definiu-se que o algoritmo executasse por no máximo 7200 segundos (2 horas) cada um dos testes. Aqueles que não encontraram a solução neste tempo foram interrompidos e a célula foi preenchida com: INT.

A coluna *Comp_Hung* apresenta o número de componentes encontrado pelo Algoritmo Húngaro para o problema relaxado, ou seja, quando não há restrição para o número de veículos. A coluna *R_Hung* mostra o custo total da solução do Algoritmo Húngaro para o problema relaxado, sendo esta um ciclo. As colunas *R_Mais* e *R_Menos* correspondem ao custo total para executar os ciclos encontrados pelas heurísticas *Mais_Veículos* ou *Menos_Veículos*, respectivamente. A primeira é executada sempre que o número de veículos informado for maior do que o valor da coluna *Comp_Hung* e a segunda quando o número de veículos informado for menor do que o valor da coluna *Comp_Hung.* A coluna *R_BB* apresenta o custo do ciclo encontrado pelo Algoritmo B&B. Esta terá sempre valores que variam entre o resultado do Algoritmo Húngaro e o resultado da heurística (*Mais_Veículos* ou *Menos_Veículos*) que foi executada para o respectivo teste.

(continua)

Teste	Método B&B					
	T_BB (s)	***Comp_Hungg***	***R_Hung***	***R_Mais***	***R_Menos***	***R_BB***
0	20,01	5	3204		3498	3489
1	11,3	5	3204		3450	3204
2	9,313	5	3204		3450	3204
3	13,369	5	3204		3450	3204
4	8,19	5	3204			
5	12,043	5	3204	3204		3204
6	17,534	5	3204	3204		3204
7	INT	5	3204	3245		3245
8	INT	5	3204	3327		3327

(conclusão)

Teste	Método B&B					
	T_BB (s)	***Comp_Hungg***	***R_Hung***	***R_Mais***	***R_Menos***	***R_BB***
9	INT	5	3204	3946		3946
10	14,742	2	1774	1774		1774
11	19,188	7	2075		2269	2075
12	11,653	6	4227		4534	4227
13	15,584	3	4932	4932		4932
14	15,3	3	501		501	501
15	11,6	4	1008		1008	1008
16	12,339	3	6604		7323	6604
17	INT	6	3520		3661	3578
18	14,8	3	3852		3933	3852
19	39,156	3	3511		3569	3511
20	14,664	3	3283		3498	3283
21	INT	3	5477		5563	5543
22	1969,787	6	43709		43918	43918
23	INT	9	12077		12221	12150
24	14,149	7	7845		7957	7845
25	22,76	4	7042		7164	7042
26	72,098	4	23413	23413		23413
27	66,531	4	17169	17169		17169
28	239,628	7	33089	33089		33089
29	6166,824	11	76358		76358	76358

Quadro 3: Matriz de resultados B&B
Fonte: Autoria Própria (2014)

A análise do Quadro 3 permite verificar, por meio dos resultados dos testes 0 a 9, que a heurística *Mais_Veículos* é mais eficiente do que a *Menos_veículos*. Isto é devido ao fato de que, em todos os testes, naquela os valores encontrados foram iguais aos valores dados pelo Algoritmo Húngaro. Ou seja, a heurística *Mais_Veículos* conseguiu, em todos os casos, encontrar uma solução de custo mínimo (custo do problema relaxado) e com o número de rotas igual ao número de veículos informado (solução viável). Já na heurística *Menos_veículos*, verifica-se que, em 84,2% dos testes, o custo da solução viável encontrada é maior do que o custo inicialmente calculado pelo Algoritmo Húngaro.

Os testes 10 a 13 permitem concluir que as quantidades de origens e destinos não influenciam no tempo de execução do algoritmo. Da mesma forma, os resultados dos testes 14, 15 e 16 demostram que a amplitude dos custos não interfere do tempo de execução do algoritmo para testes do porte analisado.

Por meio dos testes 17 a 22, verificou-se a solução caso o problema fosse o PCV com 80, 85, 90, 100, 200 e 1000 pedidos. Salienta-se que a matriz com 1000 pedidos foi solucionada em 1969,787 segundos, tempo eficiente para a complexidade e porte do problema.

Os testes mostraram que a solução é encontrada mais rapidamente quanto menor for a diferença entre o número de veículos dado pelo Húngaro e o número de veículos disponíveis. Verificou-se esta ocorrência para os testes 7 a 9, 17 e 21. Porém, verificou-se também a influência do número de pedidos. Por exemplo, no teste 18 e no teste 21 a diferença supracitada era 2, mas o segundo foi interrompido. Isto se deve ao número de pedidos deste, que é 1000.

Os testes 23 a 29 novamente demonstraram a eficiência do algoritmo desenvolvido para casos de grande porte, nos quais a diferença entre o número de veículos para o problema relaxado e o número de veículos disponíveis é pequena, tal como o teste 29, no qual encontrou-se a solução para o atendimento de 2000 pedidos em 1,71 horas.

Dado que os testes 7, 8, 9, 17 21 e 23 não encontraram a solução em 2 horas de execução, avaliou-se qual seria o comportamento destes quando executados por 21600 segundos (6 horas). Como resultado, nenhum dos testes foi finalizado e nenhuma solução melhor que aquelas relacionadas à interrupção em 7200 segundos foi encontrada.

Os métodos utilizados para cálculo final do custo de deslocamento e variável também foram comparados:

(continua)

Teste	Método 1			Método 2		
	Deslocamento	Variável	Total	Deslocamento	Variável	Total
0	69	3469	3538	20	3267	3287
1	602	3169	3771	126	2990	3116
2	888	3085	3973	267	2867	3134
3	1174	3023	4197	371	2743	3114
4	1392	3042	4434	236	2619	2855
5	1610	2970	4580	184	2522	2706
6	1959	2950	4909	266	2418	2684
7	3728	2414	6142	840	1667	2507

(conclusão)

Teste	Método 1			Método 2		
	Deslocamento	Variável	Total	Deslocamento	Variável	Total
8	4063	2487	6550	951	1570	2521
9	4439	2723	7162	1169	1516	2685
10	1014	1678	2692	167	1505	1672
11	1017	1919	2936	187	1784	1971
12	1063	4117	5180	103	3490	3593
13	1355	4604	5959	360	4242	4602
14	20	487	507	13	485	498
15	183	985	1168	29	909	938
16	1216	6518	7734	86	6196	6282
17	96	3549	3645	29	3482	3511
18	130	3808	3938	44	3747	3791
19	177	3480	3657	31	3410	3441
20	202	3262	3464	16	3209	3225
21	72	5530	5602	29	5457	5486
22	214	43720	43934	12	43906	43918
23	11	12137	12148	11	12066	12077
24	650	7594	8244	256	7315	7571
25	196	6959	7155	155	6821	6976
26	653	23278	23931	335	22990	23325
27	926	16990	17916	225	16742	16967
28	828	32831	33659	234	32574	32808
29	1261	75628	76889	384	76040	76424

Quadro 4: Comparativo entre os métodos
Fonte: Autoria Própria (2014)

Os resultados do Quadro 4 permitem verificar que o Método 2 apresentou resultados de menor custo total em todos os testes, em relação ao Método 1. Ao analisarem-se separadamente os valores de custo de deslocamento e variável, pode-se perceber que as reduções nos custos totais foram devidas a reduções nos custos de deslocamentos e também os variáveis, em todos os testes.

4.2 ESTUDO DE CASO: TRANSPORTE DE MINÉRIOS

Além das simulações supracitadas, verificou-se a possibilidade de resolução do problema em um cenário real. Para tanto, realizou-se um estudo de caso em uma empresa transportadora de minérios por meio do modal rodoviário, sediada na cidade de Campo Largo, Paraná.

Esta empresa possui 4 caminhões de mesma capacidade disponíveis para transporte de minérios entre 3 minas e 3 clientes. Todos os veículos iniciam as rotas na própria cidade aonde a empresa é sediada e o carregamento é sempre com carga completa. Os veículos devem sair da cidade de Campo Largo, direcionar-se a uma das minas para carregamento de minério, descarregar o mesmo no cliente correspondente e, em seguida, escolher em qual mina fará o próximo carregamento.

Os custos para deslocamento dos caminhões carregados são apresentados pela Tabela 16:

Tabela 16: Exemplo de matriz de custos fixos

	Cliente 1	Cliente 2	Cliente 3
Mina 1	466	397	650
Mina 2	146	418	550
Mina 3	333	461	354

Fonte: Barros & Correa Mineração LTDA

Os custos para deslocamento dos caminhões vazios são apresentados pela Tabela 17:

Tabela 17: Exemplo de matriz de custos variáveis

	Mina 1	Mina 2	Mina 3
Cliente 1	413	122	289
Cliente 2	333	360	382
Cliente 3	600	130	354

Fonte: Barros & Correa Mineração LTDA

Para exemplificar, executou-se o algoritmo para uma amostra de 65 pedidos aos quais a empresa estudada atendeu no mês de Janeiro de 2014, tal como mostra o Quadro 5.

(continua)

Origem	**Destino**
Mina 3	Cliente 1
Mina 1	Cliente 1
Mina 3	Cliente 1
Mina 2	Cliente 1
Mina 2	Cliente 2
Mina 1	Cliente 1
Mina 2	Cliente 2
Mina 3	Cliente 1
Mina 2	Cliente 1
Mina 2	Cliente 2
Mina 2	Cliente 2
Mina 3	Cliente 1
Mina 2	Cliente 2
Mina 1	Cliente 1
Mina 2	Cliente 1
Mina 2	Cliente 1
Mina 2	Cliente 1
Mina 2	Cliente 1
Mina 2	Cliente 1
Mina 2	Cliente 2
Mina 3	Cliente 1
Mina 2	Cliente 2
Mina 1	Cliente 1
Mina 2	Cliente 2
Mina 2	Cliente 2
Mina 2	Cliente 2
Mina 1	Cliente 1
Mina 2	Cliente 1
Mina 2	Cliente 2
Mina 1	Cliente 1
Mina 2	Cliente 2
Mina 3	Cliente 1
Mina 2	Cliente 2
Mina 2	Cliente 2
Mina 3	Cliente 1
Mina 2	Cliente 2
Mina 1	Cliente 1

(conclusão)

Origem	**Destino**
Mina 2	Cliente 1
Mina 2	Cliente 1
Mina 2	Cliente 2
Mina 1	Cliente 1
Mina 2	Cliente 2
Mina 1	Cliente 1
Mina 2	Cliente 2
Mina 2	Cliente 1
Mina 3	Cliente 1
Mina 3	Cliente 1
Mina 2	Cliente 1
Mina 2	Cliente 2
Mina 2	Cliente 1
Mina 2	Cliente 1
Mina 2	Cliente 1
Mina 2	Cliente 2
Mina 2	Cliente 1
Mina 2	Cliente 1
Mina 2	Cliente 1
Mina 2	Cliente 1
Mina 2	Cliente 2
Mina 2	Cliente 1
Mina 2	Cliente 2
Mina 3	Cliente 1
Mina 2	Cliente 1
Mina 2	Cliente 1
Mina 2	Cliente 1
Mina 2	Cliente 1

Quadro 5: Pedidos da mineradora em Janeiro de 2014
Fonte: Barros & Correa Mineração LTDA

Executando-se o algoritmo para o atendimento dos pedidos do Quadro 5 utilizando-se 4 carretas, obteve-se a resposta em 9,18 segundos. No que se refere aos custos, verificou-se que o resultado dado pelo algoritmo utilizando o Método 1 foi 2,2% menor do que o custo da rota executada no cenário real. Já para o Método 2, a redução de custo decorrente do uso do algoritmo proporcionaria uma redução de custos em 7,6%, correspondente a R$11.508,00 por ano.

Efetuaram-se também testes caso a empresa desejasse atender aos pedidos do Quadro 5 utilizando menor número de veículos. Assim, verificou-se que o uso de 1, 2 ou 3 veículos acarretaria exatamente no mesmo custo total. Tal observação pode ser utilizada pela empresa como estratégia para otimização de seu serviço, uma vez que, dependendo dos prazos de entrega, condição dos veículos, disponibilidade dos motoristas, condições climáticas, dentre outras variáveis de decisão, a empresa poderia optar por utilizar 1, 2, 3 ou 4 carretas para o atendimento dos mesmos pedidos, sem que o custo total fosse alterado.

5 CONCLUSÃO

O gerenciamento de empresas transportadoras de carga envolve grande número de variáveis no processo decisório. Sendo assim, o aperfeiçoamento dos sistemas de gestão de frotas representa um fator de grande relevância à competitividade destas empresas. A otimização do processo decisório contribui para a redução de custos e abertura de novas possibilidades de investimento e planejamento estratégico.

O método proposto atingiu seu objetivo ao demonstrar que o problema de escalonamento de veículos para atendimento de pedidos pré-estabelecidos pode ser resolvido de maneira eficiente por meio de método heurístico baseado nos conhecidos problemas da Designação e Caixeiro Viajante.

O algoritmo é capaz de distribuir os veículos para atendimento de elevado número de pedidos, como comprovou ao escalonar o atendimento de 10 veículos a 2000 pedidos em 1,713 horas. Além da viabilidade do tempo de execução, em um estudo de caso, o método encontrou rotas mais baratas do que aquelas determinadas empiricamente, resultando em economia de R$ 11.508,00 por ano, para um exemplo real de pequeno porte.

Tais ganhos podem ser ainda maximizados quando se verifica a possibilidade de realizar de maneira rápida simulações para o mesmo conjunto de pedidos. Desta forma, auxiliando ao desenvolvimento de novas estratégias, conforme a necessidade de cada empresa.

Outra característica da implementação consiste na grande aplicabilidade. Este trabalho apresentou um estudo de caso em uma transportadora do modal rodoviário, que é o mais utilizado no Brasil. Entretanto, destaca-se que o mesmo algoritmo pode ser utilizado para otimização de atendimento a pedidos pré-estabelecidos no modal aéreo, aquaviário (marítimo e hidroviário) ou ferroviário, desde que atendam à restrição de carga completa.

Para continuidade do trabalho, indica-se o desenvolvimento de uma interface gráfica para interação com o usuário, desenvolvimento de banco de dados e implementação de solução do mesmo problema por meio de outras heurísticas, a fim de se comparar os custos e tempos de processamento. Salienta-se que nenhum algoritmo que objetivasse resolver o mesmo problema foi encontrado na literatura.

REFERÊNCIAS

ARENALES, Marcos; ARMENTANO, Vinicius; MORABITO, Reinaldo; YANASSE, Horacio. **Pesquisa operacional**. Rio de Janeiro: Campus/ Elsevier, 2007. 523 p.

BALLOU, Ronald H. **Gerenciamento da cadeia de suprimentos:** planejamento, organização e logística empresarial. 4. ed. Porto Alegre: Bookman, 2001. 532 p.

BONDY, J. A.; MURTY, U. S. R.. ***Graph Theory***. Springer, 2008. 655 p.

CAMPONOGARA, E. **Métodos de Otimização: Teoria e Prática**. Florianópolis: UFSC, 2006, 369 p. Disponível em: <http://www.das.ufsc.br/~camponog/Disciplinas/DAS-9011/LN.pdf> Acesso em: 07 out. 2013.

CHOPRA, S; MEINDL, P. 2011. **Gestão da Cadeia de Suprimentos: estratégia, planejamento e operações**. Pearson, São Paulo.

CORMEN, T. H; LEISERSSON, C. E; RIVEST, R. L.; STEIN, C**. *Introduction to Algorithms***. 3. ed. Cambridge: MIT Press, 2009, 1292 p.

DIESTEL, Reinhard. ***Graph Theory***. 2005. ed. Nova Iorque: Springer, 2005. 422 p. Disponível em: <http://www.math.uni-hamburg.de/home/diestel/books/graph.theory>. Acesso em: 29 nov. 2013.

FERREIRA, Carlos E.; WAKABAYASHI, Yoshiko. **Combinatória Poliédrica e Planos-de-Corte Faciais**. 1996. Disponível em: <http://www.ime.usp.br/~yw/2010/progint/livro-update2010.pdf>. Acesso em: 30 ago. 2013.

FIGUEIREDO, Daniel Ratton; FIGUEIREDO, Celina Miraglia Herrera de**. Teoria dos Grafos**: Aula 6. 2011. Disponível em: <http://www.land.ufrj.br/~classes/grafos/slides/aula_6.pdf>. Acesso em: 20 maio 2014.

GANHOTO, Marco. **O Problema do Caixeiro Viajante**. 2003. Disponível em: <http://www.siscorp.com.br/siscorpnews/sexta_edicao/assunto_marco.htm>. Acesso em: 08 dez. 2013.

GOLDBARG, Marco Cesar; LUNA, Henrique Pacca L**. Otimização combinatória e programação linear: modelos e algoritmos**. 2. ed. rev. atual. Rio de Janeiro: Campus; Elsevier, c2005. xiv, 518 p.

HILLIER, F. S; LIEBERMAN, G. J.. **Introdução à pesquisa operacional**. Rio de Janeiro: Campus, 1988. 805 p.

KUHN, Harold W. ***The Hungarian Method for the assignment problem***, Naval *Research Logistics Quarterly*, v. 2, p. 83–97, 1955.

LACHTERMACHER, Gerson. **Pesquisa operacional na tomada de decisões**. 4. ed. São Paulo: *Pearson Prentice Hall*, 2009. 223 p.

MACHADO, Fabricio C.. **Problema do Caixeiro Viajante com Coleta e Entrega**. Campinas: 2013. 19 p.

NASCIMENTO, Mariá Cristina Vasconcelos. **Metaheurística para o problema de agrupamento de dados em grafo.** 2010. 119 f. Tese (Doutorado) - Curso de Ciências de Computação e Matemática Computacional, Usp, São Carlos, 2010. Disponível em: <http://www.ppgia.pucpr.br/~fabricio/ftp/Mestrado/Bibliografia Agrupamento em Grafos/Tese_Nascimento.pdf>. Acesso em: 23 nov. 2013.

NOVAES, Antonio Galvao *et al*. **Gerenciamento de transporte e frotas.** 2. ed. Cengage Learning, 2008. 340 p.

NOVAES, A. G.. **Logística e gerenciamento da cadeia de distribuição: estratégia, operação e avaliação**. 2 ed. Rio de Janeiro: Elsevier, 2004.

Portal ANTT. RNTRC em números. Disponível em: <http://www.antt.gov.br/>.Acesso em: 27 Jun. 2013.

RENDER, B; STAIR, R. M.; HANNA, M. E. **Análise quantitativa para administração**. 10. ed. Porto Alegre: Bookman, 2010. 776 p.

SEDGEWICK, Robert. ***Algorithms in C:*** *Part 5: Graph Algorithms*. 3. ed. Pearson, 2002. 512 p.

SILVA, Ermes Medeiros da; SILVA, Elio Medeiros da; GONÇALVES, Walter; MUROLO, Afrânio Carlos. **Pesquisa operacional: programação linear, simulação**. 3. ed. São Paulo: Atlas, 1998. 184 p.

SOUSA, Diego Augusto de. **Programação Inteira:** *Método branch and bound – Programando com estruturas de dados*. 2008. 72 f. Monografia - Curso de Matemática Aplicada A Negócios, USP, Ribeirão Preto, 2008.

TACLA, D.; BOTTER, R. C.; HINO, C. M. **Estudo e aplicação de transporte colaborativo para cargas de grande volume -** Tese de Doutorado. Escola Politécnica da Universidade de São Paulo. Brasil, 2003.

TAHA, Hamdy A. **Pesquisa operacional.** 8 ed. São Paulo, SP: Pearson Prentice Hall, 2008. 359 p.

TENENBAUM, Aaron M.; LANGSAM, Yedidyah; AUGENSTEIN, Moshe. **Estruturas de dados usando C.** São Paulo: Makron, 1995. xx, 884 p.

APÊNDICE A: ARQUIVOS DE ENTRADA PARA TESTES: CUSTOS.

10	10								
105	282	80	147	120	103	400	320	40	94
205	326	29	479	30	29	496	499	418	226
360	150	348	114	409	448	391	90	62	246
230	469	248	241	488	354	103	152	67	321
152	179	366	415	279	22	87	396	177	270
193	381	502	370	363	98	273	266	49	225
104	397	100	292	34	15	453	155	318	117
456	410	177	279	495	301	199	178	396	461
290	160	290	202	408	273	205	329	158	140
339	225	252	485	391	208	43	323	168	239
73	143	251	160	106	135	72	319	202	237
197	228	104	328	125	266	277	286	111	157
55	20	243	173	256	351	69	123	202	176
102	335	79	168	290	258	204	195	141	339
83	20	286	341	195	254	23	346	285	273
72	20	313	247	15	68	10	210	191	145
279	347	273	72	60	191	317	139	143	30
223	349	62	106	277	186	108	124	230	226
27	292	43	46	123	34	222	277	110	117
65	158	172	224	188	157	81	322	97	167

Figura 38: Arquivo de custos utilizado nos testes 0 a 9

Fonte: Autoria Própria (2014)

20	20																		
13	406	464	33	396	272	405	427	447	275	140	253	95	243	498	100	146	371	21	339
64	201	240	369	356	363	395	60	146	464	274	146	310	219	187	371	83	132	86	118
169	249	312	225	199	151	37	194	30	93	282	272	110	159	275	505	22	275	22	97
326	160	488	398	42	192	164	182	266	216	477	206	312	125	287	496	141	45	401	427
232	150	481	454	500	286	168	422	132	76	164	196	68	106	215	18	121	316	378	436
149	460	354	127	260	210	29	437	497	498	369	348	301	163	448	347	301	338	467	241
323	123	481	337	88	44	158	353	195	108	473	323	359	13	233	149	364	141	395	136
159	417	69	335	485	311	177	365	435	50	229	412	431	296	293	448	489	119	334	83
379	339	326	14	109	163	479	408	214	41	436	397	175	423	370	138	36	228	355	403
404	495	357	185	64	230	385	441	229	389	34	220	62	74	204	129	499	301	35	336
168	59	365	240	444	508	255	300	210	371	99	37	137	189	31	211	66	318	396	444
418	14	362	145	368	132	35	467	213	31	354	465	55	367	89	133	219	303	216	232
498	458	452	362	378	18	244	437	248	279	235	33	217	134	502	32	157	126	211	82
91	331	199	428	207	428	444	260	140	116	93	412	130	461	386	120	258	49	67	377
463	322	201	268	210	456	415	158	25	156	321	241	289	278	146	151	191	195	194	22
199	128	503	415	104	412	302	302	263	216	80	178	263	411	254	353	64	364	372	399
339	417	76	392	154	435	351	199	81	447	300	33	275	159	374	149	84	329	26	80
31	96	212	284	204	337	50	49	373	419	457	196	400	22	421	432	351	196	479	201
183	185	345	90	102	205	408	86	80	27	316	147	55	208	165	239	346	112	210	490
507	17	67	319	85	214	191	275	164	133	273	115	274	200	181	474	262	168	195	334
9	44	118	228	162	104	226	111	265	282	117	292	348	63	324	139	237	21	128	354
284	140	174	111	104	321	86	291	237	346	41	9	320	231	225	89	291	67	129	11
324	167	218	341	336	247	336	48	228	249	255	253	316	139	140	352	53	148	241	46
23	258	157	278	317	88	235	234	9	129	167	101	253	299	187	290	274	198	62	223
277	249	139	29	349	181	61	339	76	44	310	257	264	144	146	72	107	142	71	59
190	254	105	134	200	146	30	217	114	160	355	92	12	299	319	288	304	235	143	149
283	276	25	114	117	20	110	123	335	269	178	24	170	310	290	211	245	34	285	133
298	41	135	127	295	305	247	255	285	308	209	326	305	181	110	211	139	34	60	192
312	102	20	186	92	347	136	304	149	160	146	149	173	97	17	184	56	261	55	114
192	324	65	151	53	348	75	34	28	272	259	21	195	81	109	151	312	293	18	93
97	191	197	333	114	258	331	160	305	23	69	247	164	65	224	55	209	319	221	191
177	102	190	144	137	243	226	288	277	153	25	325	23	288	168	124	23	137	102	80
66	216	76	218	47	210	251	301	122	43	95	38	151	90	202	184	192	279	38	191
170	153	111	87	74	114	9	207	296	51	132	256	93	322	194	287	111	15	145	139
348	130	192	200	150	313	163	205	258	142	21	62	351	270	102	177	261	294	115	126
69	259	353	347	12	242	104	313	96	90	147	93	22	83	105	247	104	302	167	331
102	58	15	98	84	210	254	342	25	349	46	153	109	180	133	44	58	245	242	183
259	92	192	31	221	236	98	83	159	210	222	212	88	34	136	254	230	137	78	117
14	60	15	280	264	326	276	233	248	24	277	151	147	46	135	260	18	335	146	136
237	82	67	298	305	168	95	58	282	235	310	162	57	263	15	279	55	140	342	233

Figura 39: Arquivo de custos utilizado no teste 10

Fonte: Autoria Própria (2014)

```
40 40
222 282 365 160 167 405 447 301 482 364 312 311 103 84 301 146 447 448 71 384 129 460 158 208 206 54 294 304 430 343 387 299 189 437 258 386 428 334 192 404
45 127 343 223 426 390 155 425 156 93 419 83 107 208 384 437 331 468 209 22 379 249 139 82 121 445 334 369 437 46 89 53 399 390 247 98 158 81 251 221
98 499 285 475 84 122 169 164 293 266 172 83 388 331 156 324 52 403 229 500 441 185 129 267 18 393 496 443 235 272 164 459 147 207 242 107 145 483 402 451
19 15 244 35 311 458 27 149 461 295 146 483 56 398 61 422 356 495 469 43 425 85 31 497 426 376 502 303 168 39 226 328 456 369 392 468 330 198 132 181
312 456 234 298 194 71 156 127 51 178 186 29 364 374 207 386 203 56 112 504 135 78 79 214 234 98 189 252 231 195 70 133 481 380 316 506 135 90 247 416
258 504 344 309 484 232 109 107 266 145 100 157 356 320 382 458 428 95 418 335 396 256 306 484 132 141 321 478 491 80 385 220 509 194 243 335 49 487 177 299
83 128 99 254 477 256 85 371 402 372 432 207 507 231 255 71 37 168 84 131 439 365 183 196 124 319 358 140 227 471 338 66 71 181 381 404 473 383 464 425
360 329 331 406 111 202 126 228 292 45 350 423 21 344 277 398 335 305 74 299 179 252 313 172 396 260 429 186 188 433 247 288 330 217 14 315 320 454 187 149
370 208 503 128 114 83 290 305 126 230 146 200 467 401 381 42 407 13 28 296 262 338 488 253 484 373 93 95 114 79 157 258 225 176 334 330 123 184 407 94
393 259 239 276 220 227 43 63 437 403 441 207 434 106 502 111 478 289 54 181 269 368 84 138 66 162 275 423 122 220 335 135 16 176 193 502 131 507 90 15
496 51 305 469 314 408 87 279 416 309 286 89 161 119 274 69 478 426 399 81 485 35 271 195 359 385 183 202 26 54 339 411 81 161 395 412 270 190 139 172
118 433 92 184 34 362 320 491 99 116 454 56 146 173 470 231 466 178 98 477 31 387 398 154 137 325 396 444 340 130 508 294 389 209 142 300 428 366 309 273
215 500 59 142 143 181 445 401 98 430 197 202 369 24 130 76 323 255 132 248 93 449 367 149 211 340 16 159 483 171 382 245 93 234 243 449 147 149 232 252
497 175 380 369 327 259 155 474 14 246 334 356 197 262 383 402 53 151 471 221 403 17 174 366 216 57 442 128 321 327 158 64 321 259 457 479 331 314 155 267
72 19 375 40 420 263 241 376 64 86 367 25 29 183 375 321 18 98 463 376 503 276 39 474 244 455 142 343 332 454 295 339 500 342 266 457 363 110 192 91
375 209 507 191 263 151 165 192 146 337 290 236 324 256 252 385 124 350 23 353 150 447 306 38 322 407 464 234 188 433 202 32 333 399 31 347 306 403 285 254
232 142 48 260 248 30 161 57 264 422 365 177 205 492 170 301 10 258 492 115 429 130 67 270 504 462 61 470 495 428 463 480 320 465 269 349 339 136 421 483
451 301 72 296 445 330 359 99 216 291 123 389 180 337 304 413 363 320 283 311 229 217 218 501 474 413 88 409 493 308 481 45 191 400 187 501 490 438 442 493
112 21 197 119 371 428 257 425 405 214 97 434 239 464 95 377 402 179 354 430 438 287 45 326 167 91 162 397 220 258 207 42 14 353 229 217 29 241 489 235
308 308 276 447 395 131 161 328 332 343 18 212 235 317 72 331 47 294 91 363 384 274 233 430 26 43 376 104 281 426 393 430 463 350 232 307 56 420 241 75
404 337 209 98 42 377 314 498 215 84 471 127 328 404 356 305 360 199 469 279 53 61 312 63 246 207 16 478 253 265 451 193 279 249 159 476 120 185 316 190
503 391 445 157 197 334 474 244 53 84 480 54 136 499 455 41 445 368 392 234 48 106 198 455 27 214 311 46 258 459 342 468 191 261 59 38 236 26 468 482
15 316 487 17 386 59 482 308 101 441 425 165 136 121 110 363 205 62 213 89 378 156 483 50 311 363 233 397 323 370 249 451 411 93 276 105 259 67 344 396
432 154 159 170 509 488 145 11 50 435 55 428 238 468 88 86 75 485 332 138 332 298 15 12 354 504 239 29 311 117 285 302 411 355 10 462 259 357 476 259
316 366 234 395 156 91 260 188 144 266 324 246 73 480 490 452 170 293 177 46 162 506 306 217 190 244 265 27 79 378 110 89 405 470 453 171 54 188 396 91
428 236 120 248 352 283 423 255 405 19 66 223 264 34 431 493 406 408 424 435 455 208 37 451 468 57 157 247 451 465 148 495 20 505 469 216 376 114 50 30
282 132 253 476 170 237 439 143 105 252 56 30 12 470 273 184 104 213 133 159 162 324 227 296 497 110 330 302 149 350 434 384 276 430 459 461 262 422 207 410
357 140 173 503 170 294 395 361 131 219 339 487 141 174 334 283 345 135 36 471 274 139 128 49 313 102 115 183 257 367 27 198 140 216 194 54 218 78 198 409
348 171 107 124 135 158 192 439 342 356 291 427 457 61 207 490 496 456 26 421 472 314 54 36 456 279 179 463 238 155 40 256 199 285 447 505 224 194 317 176
139 328 242 24 465 271 40 293 449 384 99 267 285 84 432 219 448 251 62 323 452 129 30 491 148 433 504 147 124 128 483 102 175 301 409 324 398 323 251 13
208 433 455 162 410 125 62 167 11 402 217 269 374 376 74 360 107 70 68 341 347 118 252 486 72 373 489 270 224 111 487 395 293 505 354 285 193 228 136 497
491 125 356 481 122 472 332 42 436 273 67 459 237 286 212 24 361 461 14 314 476 428 188 313 172 350 391 396 12 119 137 484 447 200 404 451 143 496 488 174
479 238 373 417 507 225 208 106 130 172 227 257 248 373 319 138 29 109 142 336 110 24 38 368 155 223 156 252 484 33 16 58 260 49 461 444 395 30 340 286
111 432 13 138 505 284 64 330 497 273 288 451 295 176 102 49 107 481 50 464 27 135 383 153 56 374 447 43 314 339 303 79 252 51 127 114 312 428 470 386
226 224 417 440 243 43 118 124 207 79 427 245 447 126 379 383 285 269 160 30 169 450 477 126 253 194 466 115 27 416 502 389 477 424 257 504 426 329 398 80
379 242 273 337 324 298 17 333 424 141 496 338 34 472 72 297 458 413 370 119 111 489 255 424 335 495 210 465 184 241 277 370 227 311 248 487 34 446 100 14
58 55 62 370 247 213 317 126 500 344 50 450 125 231 278 197 490 407 19 150 189 456 347 104 340 21 116 292 17 371 456 134 57 123 71 378 46 253 203 305
153 47 126 438 192 157 411 150 424 484 402 200 177 191 225 93 252 187 83 86 117 73 475 62 243 381 410 159 201 371 472 257 446 129 92 497 35 57 17 346
295 172 415 65 484 359 355 70 180 280 232 92 439 92 227 361 347 484 376 320 115 388 428 262 88 124 345 82 480 417 253 314 213 70 60 11 46 383 132 201
361 357 281 267 221 204 416 23 120 405 368 13 219 93 400 100 305 414 232 462 146 266 494 277 269 325 438 78 364 418 245 309 209 94 359 189 36 117 431 433
155 31 68 13 218 180 58 251 258 275 347 82 150 347 50 262 162 315 78 215 282 352 10 302 221 299 197 249 243 97 145 343 335 77 158 265 40 107 206 252
197 88 349 10 319 352 89 230 145 181 35 303 349 122 13 146 99 210 14 215 235 273 221 107 256 165 92 97 119 229 303 87 166 302 156 169 38 32 120 249
255 240 199 170 163 240 69 231 352 167 213 64 41 265 262 354 33 50 137 193 146 311 340 111 163 83 177 121 74 169 318 249 261 9 291 191 43 88 290 196
111 156 332 24 208 216 177 284 89 193 328 128 99 258 27 133 181 207 83 312 68 109 11 118 276 173 333 352 86 16 113 336 291 96 307 235 258 306 45 186
116 298 58 217 135 338 333 77 79 153 219 23 100 228 293 184 173 311 259 276 29 137 270 356 109 246 118 118 94 325 286 85 354 353 170 226 172 134 338 154
283 272 85 320 49 162 179 141 58 158 285 253 126 181 184 105 20 230 299 91 263 233 41 341 63 198 165 205 110 189 87 330 157 198 30 208 149 109 251 142
312 108 118 18 109 76 59 88 202 30 60 223 311 108 168 115 112 251 179 112 219 331 337 101 181 296 307 276 134 27 43 232 145 44 82 11 221 287 248 291
210 297 114 104 88 74 259 159 213 44 195 343 280 331 263 134 39 69 297 229 348 170 215 7 131 178 100 252 307 205 116 29 74 230 86 233 88 104 48 16
337 109 205 322 35 186 281 204 88 305 291 69 68 9 44 102 184 151 283 232 150 37 70 34 100 283 73 91 239 314 7 305 90 347 144 296 349 296 125 83
254 65 186 206 124 101 260 31 160 282 216 81 300 172 60 235 295 203 149 240 58 58 308 304 186 13 176 153 249 268 281 191 120 191 55 98 240 338 195 283
218 293 120 102 130 69 302 244 102 308 200 317 137 233 256 202 255 86 67 12 329 335 297 38 226 46 39 237 203 69 151 46 158 201 298 347 34 281 162 257
217 58 58 338 20 109 144 296 139 144 62 39 141 249 17 165 123 272 303 148 88 37 115 299 172 156 20 340 298 186 188 321 179 315 171 236 314 139 64 9
72 74 271 39 254 249 354 14 326 303 112 102 258 137 20 226 143 125 167 164 229 95 95 166 51 184 8 98 319 199 261 165 173 206 312 23 87 123 307 153
58 145 231 278 261 223 161 240 280 74 83 121 16 183 128 179 344 235 324 221 282 349 84 327 335 23 328 121 42 58 263 200 261 123 88 330 161 133 64 65
210 268 109 42 144 267 178 193 266 351 191 328 90 268 262 176 118 212 66 50 249 318 76 61 342 301 191 233 144 302 51 148 223 71 265 50 194 157 158 279
102 305 226 295 270 320 49 278 29 77 48 161 53 281 224 269 210 289 263 231 213 28 254 60 316 345 128 198 342 153 251 16 96 34 268 207 137 65 252 69
312 231 36 249 142 299 25 234 284 334 334 326 226 37 12 86 6 254 281 32 251 311 143 52 118 284 72 241 347 313 74 252 20 74 199 320 342 176 242 213
313 327 282 346 39 66 117 213 9 202 298 124 178 105 68 244 180 223 125 205 139 257 43 339 205 285 149 94 319 175 48 322 76 336 188 289 284 130 338 289
49 146 160 328 78 292 58 51 19 37 279 68 92 329 324 16 344 198 247 63 328 274 149 232 123 296 93 25 18 43 47 9 99 34 111 258 13 58 263 162
268 15 349 30 352 234 91 209 207 126 56 333 173 154 263 247 80 217 300 254 195 163 62 96 32 304 111 329 294 226 238 219 235 324 20 83 104 60 223 323
90 265 308 297 94 277 307 125 183 188 339 21 65 282 352 104 300 160 306 268 37 33 264 232 113 318 113 191 330 316 242 333 76 18 118 77 132 81 80 102
321 174 129 59 54 179 255 176 236 257 24 270 314 11 193 312 90 151 200 191 42 74 109 208 354 145 226 97 219 90 82 299 16 94 314 342 319 51 271 186
110 97 90 21 55 214 128 219 341 58 189 278 256 121 27 214 46 152 31 163 218 138 338 10 214 25 158 89 37 20 176 131 26 268 333 178 242 332 299 345
145 57 186 347 149 338 137 120 177 96 136 107 104 256 331 26 188 350 228 300 44 318 34 8 151 315 207 34 25 343 340 219 257 107 88 296 72 43 183 193
144 84 12 298 163 92 86 277 338 46 251 95 147 151 170 225 352 331 116 18 172 18 217 247 132 327 347 219 319 103 50 120 108 39 177 234 237 170 61 188
37 311 275 263 68 98 223 181 261 113 269 227 237 39 318 284 323 289 63 30 144 149 254 352 170 39 76 71 195 303 261 244 156 261 135 346 14 266 86 227
205 233 347 351 132 224 250 300 65 192 128 277 11 309 99 324 42 61 113 263 11 217 163 167 185 109 230 80 125 352 342 273 109 312 326 146 81 286 241 306
212 258 310 212 176 334 97 130 66 296 141 310 111 89 240 163 328 286 277 72 334 32 277 20 18 172 211 128 324 102 188 277 176 30 80 325 204 111 57 54
300 305 164 117 161 343 158 131 79 85 18 237 338 224 232 131 346 345 153 196 177 180 226 217 55 315 104 179 166 86 156 8 338 219 18 128 11 140 335 254
240 32 190 27 136 55 329 303 55 153 37 90 119 228 317 303 299 215 180 298 185 321 258 81 264 325 244 256 108 89 77 83 23 237 291 168 259 259 291 292
270 62 114 158 48 269 236 172 109 234 237 355 267 110 206 141 324 336 144 275 315 239 174 199 76 103 303 18 27 338 340 95 11 212 351 193 319 330 177 171
209 37 321 229 93 153 46 201 180 94 287 205 171 44 237 22 335 31 29 300 135 327 315 211 62 346 268 138 179 71 276 338 40 55 272 258 93 179 219 216
132 279 102 319 336 356 49 230 157 11 56 272 65 224 349 233 223 133 9 324 195 133 287 287 283 13 193 97 139 122 205 312 181 176 333 158 39 312 149 146
305 272 144 258 265 135 126 151 123 123 112 146 163 181 239 279 325 279 247 244 174 182 65 248 328 353 300 151 199 210 353 139 34 35 296 217 86 90 48 65
180 172 169 274 221 170 266 9 233 135 276 99 170 319 186 21 188 130 160 162 111 41 193 6 317 328 321 135 312 286 247 282 322 88 179 173 49 64 41 251
270 68 74 327 354 234 282 220 230 351 288 209 314 335 319 242 244 350 151 214 333 26 73 323 119 151 322 37 353 226 199 315 310 79 352 340 264 347 7 132
299 110 101 230 94 34 331 223 86 91 188 299 102 231 254 214 237 342 20 39 83 165 181 181 37 263 183 152 156 278 135 100 276 218 298 23 32 24 32 25
233 56 338 138 62 340 268 317 128 354 132 256 104 219 76 282 95 306 168 293 129 18 46 249 131 79 295 54 135 226 159 347 20 299 230 312 177 39 268 81
134 175 281 92 172 123 324 130 284 62 97 216 162 108 134 199 294 309 342 168 221 327 240 333 277 34 144 138 221 175 95 341 237 328 278 69 142 11 92 301
282 154 315 126 291 209 297 104 65 10 120 191 176 186 63 177 338 345 164 52 132 337 277 181 63 20 286 286 123 9 347 121 200 270 55 9 213 242 140 303
```

Figura 40: Arquivo de custos utilizado no teste 11

Fonte: Autoria Própria (2014)

```
 50   5
 84 207 440 212 204
 20 481  53 261  57
162 432 418 218  31
362 356  32 173 182
298 408 404 397 236
395 247 365  37  93
 44 182 437 441 217
217 240 352 175  97
293 112 243  71 498
227 154 193 252 500
497 339 394 209 176
178 125 345 395 153
 42 192 503 389  19
 41 330  79  54 151
249 339 192 355 236
419 162 145 231 245
 38 217 216 110  15
292  58  35 124 275
452 258 430 199 432
401 217 373 170 192
 38  92  27 125 474
393 171 423 329 439
 87 435 316  63 151
191 267  73 295 147
387 224 295  47 147
402 160  50 389  28
 21 323 472  28 182
237 391 396 199 479
272 218 125 314 462
176  49 262 336 306
181  63 300 257 191
139 291 462 130  72
357 456 131 492  23
219 161 185 467  89
205 430 115 229  25
229 286 213 390 415
 68 488  87 125 220
425 336 140 315 447
295 463 218  61 105
167 444  28 224 147
157 154 464 302 174
 93  48 398 402 129
 10 494 157 154 161
 10 507 341 354  65
 59 100 185 355  36
 41  82 301 375 263
444 369 458  99 484
406 277 374 229 100
359 130 210 292 188
 82 400 262  24 141
 58  13 113 253 208 276  30 151 205 158 347 124  29  28 174 293  26 204 316 280  26 275  60 133 270 281  14 165 190 123 126  97 249 153 143 160  47 297 206 116 109  65   6   6  41  28 310 284 251  57
144 336 302 249 285 172 127 167  78 107 237  87 134 230 237 113 151  40 180 151  64 119 304 186 156 111 226 273 152  34  44 203 319 112 300 200 341 235 324 310 107  33 345 354  69  57 258 193  90 279
307  37 292  22 282 255 305 246 170 135 275 241 352  55 134 101 151  24 300 261  18 296 221  51 206  34 330 277  87 183 209 323  91 129  80 149  60  97 152  19 324 278 109 238 129 210 320 261 146 183
148 182 152 121 277  25 308 122  49 176 146 276 272  37 248 161  76  86 139 118  87 230  44 206  32 272  19 139 219 235 179  90 344 326 160 272  87 220  42 156 211 281 107 247 248 262  69 160 204  16
142  39  21 127 165  65 151  67 348 349 123 107  13 105 165 171  10 192 302 134 331 307 105 102 102  19 127 335 323 214 133  50  16  62  17 290 153 312  73 102 121  90 112  45  25 184 338  69 131  98
```

Figura 41: Arquivo de custos utilizado no teste 12

Fonte: Autoria Própria (2014)

```
  5  50
339 115  89 374 309 393 376 102 213  27 126 368  47 111 316 245 276 408 397  99 471 505 412 479 156  12 414 248  56 264 241 448 446 183 140 216  17 410 390  77 141 142 384 388 298 373 282 498 223 491
 97 137 427 501 248 257  54 359 251  10 390  29 151 320 299 133  98 356 172 314 382  15 102  24 482 509  99 334 140 296 232 300 117 332 499  11 389 501 452 469 382  99 104 361  24  81 358 167 170 230
302  88 323 119  15 140 474 325 194 420  42 446 303 490 414 366 410 358  56 352 390 110 465 472  24 446 341 127 200 394 403 225 292 163 110 198  77 292 349 385  21  37 492 165  53 281  20 404 276 352
 18  89 221 115 426 340 170 378 130 225 140 414 276 325 488 457 455 468 331 343 170 332 182  43 310  40 301 493 167 177 142 248 232 133 144 182 394 208 488  43 306 320 342 232 274 199 169 333 226 320
 61 293 203 263 205 508 239  68  20 278 358 398 282 482 209 311 367 309 501 104 124 468 175 227 247 414 153 147 304 335 143 251 168 206 446 369  80 222 145 421 153 352 299  78  99 278 112 324 179 358
237  67 211  12  42
 80  95  61  62 205
 62 298 226 154 142
261 350  83  80 184
216 173  10 298 143
275 179  97 237 355
263  37 331 118 167
 71 251 227 264  47
149 175 135  90  13
 18   6 293 157 194
 88 272  29  97 250
257  20 312 289 278
 32 105 212 193 197
 77 223 342 227 337
221 209 289 341 146
171  93 256 319 217
193  68 286 318 256
285 249 250 327 216
277 120  39 231 350
 69 219 246 240  72
329 267 272 118  86
353  10  76 232 327
288  71 325 127 122
335  16 330  30 158
109 337  16 216 172
  8 356 312  27 289
289  69 238 210 107
173 233  88 345 102
 39  97 139 116 212
184 207 275 123 234
168 162 282  99 100
313 209 157 173 175
312  81 204 162 117
128 232 114  93 144
 97 349  76 100 312
151   7 138 127 258
 11 272  53 275  55
286 350 204 145 155
272 316 244 341 101
 53 328 269  30 294
 98 267  14 214 107
 99  69  25 223 246
268  72 344 239 209
271 252 115 162  54
208  16  37 191  69
261  56 196 139 194
197 250  13 118  78
348 116 282 233 226
156 118 193 158 125
343 160 246 223 250
```

Figura 42: Arquivo de custos utilizado no teste 13
Fonte: Autoria Própria (2014)

```
15 15
14 19 13 19 19 14 11 11 11 12 14 11 10 18 13
12 19 19 15 10 10 11 11 13 13 12 18 16 14 14
18 12 11 13 17 19 16 14 19 16 11 17 14 19 12
14 18 15 17 16 14 13 16 14 17 15 18 18 11 18
14 11 13 12 10 19 19 17 17 17 10 19 17 12 15
17 13 16 18 15 14 15 13 19 19 14 13 18 15 13
13 19 13 16 11 18 10 19 13 12 15 14 14 19 16
17 19 11 12 17 16 13 11 14 17 12 10 17 15 11
15 15 12 11 13 12 10 11 17 17 14 11 18 17 11
18 19 16 13 10 16 17 12 17 14 17 14 10 18 17
16 12 17 16 14 19 17 10 14 14 15 19 10 11 17
14 12 12 19 10 10 13 12 19 11 11 12 12 18 14
10 12 16 17 14 11 16 12 17 10 14 19 10 17 15
14 16 16 19 16 10 12 14 11 17 14 15 18 19 13
17 10 15 14 10 13 10 16 12 19 13 15 18 14 12
 9  8 12  9  9 11  9 11 10 12 11  9  6  9 11
13 13  8 12  7  9 13 13 10 13  8  8  8 11  6
 9 13  7 10  9 11  9  7  8 11 11  8 11 11 10
13 10  9 11  8 12 11  8  7  9 11 13 11 13  9
13  6 11 11  6 10  7 11  9  6  9  6  9 11  6
 9  6 13  9 13  9 12 11  8 11 13  6  7  6  9
 7  7 11  9 13 10  6  9  6 11 11  9 11  8  6
 7  7  9 11 11  9 13  7  7  8  6  8  8  9 11
 7  9 13  9 11 13  9  9 11 11  9 13 11  7  8
 8  9 11 11 11 13  8 11 11  9  9  7  6 11 13
 9  8  7 10  6  9 10  8  9 11 10  7  9  9  9
 7 12 11 12 13  9  9  6  7  9 13  8 13 10 10
 6 11  9 12 11 12  9 11 12  6  6  8  6 12 12
12  9 13  7  8 10 13 10 11 12  7 12 11 13  9
 9  9  8 12 10  9 11  7  7 11 11  9 10  9  8
```

Figura 43: Arquivo de custos utilizado no teste 14
Fonte: Autoria Própria (2014)

15	15													
68	11	17	82	103	105	45	28	27	49	102	34	13	78	54
80	40	13	24	78	94	11	39	94	21	77	80	23	87	69
102	76	40	61	80	42	88	75	35	43	69	97	76	60	45
62	24	75	107	69	10	16	14	92	29	45	90	40	63	40
15	70	49	90	107	53	60	12	84	87	22	96	34	45	81
22	14	19	89	76	40	78	57	24	68	64	48	63	11	73
104	97	12	76	75	51	25	74	12	44	16	16	104	57	53
59	101	47	53	47	75	37	67	77	73	101	19	109	74	51
72	46	60	17	91	77	65	38	92	12	29	49	107	56	31
27	99	108	60	60	17	32	23	87	97	104	69	72	48	29
97	18	70	58	55	20	22	47	23	29	90	22	43	32	41
78	40	89	94	93	49	40	94	36	64	93	13	96	27	17
41	71	34	80	80	41	17	80	75	76	105	80	42	58	18
52	30	106	23	60	33	76	101	24	65	77	49	99	79	90
65	43	105	63	17	28	54	27	10	62	45	102	86	56	17
47	55	71	43	10	15	72	41	50	18	67	54	28	36	45
7	27	53	16	48	9	67	70	32	69	12	27	49	20	30
11	9	27	52	34	13	8	32	41	75	48	62	23	74	73
57	16	42	74	62	62	53	37	11	41	40	65	55	16	44
72	54	55	48	74	53	52	32	63	41	38	65	55	41	11
73	65	29	6	37	27	35	52	53	11	13	34	28	23	19
31	7	61	11	41	54	17	25	45	22	15	27	11	53	37
19	27	52	9	8	39	51	46	26	16	32	65	55	70	18
18	65	24	64	58	16	8	53	64	60	16	25	52	16	6
34	14	30	20	60	47	30	51	8	67	20	44	53	45	43
71	53	48	31	15	44	11	70	20	72	62	65	73	53	31
23	55	67	62	67	33	11	13	34	48	15	9	55	34	71
9	16	53	27	23	44	72	76	74	50	30	67	29	69	60
54	60	41	44	31	7	39	51	39	33	22	18	40	55	39
37	48	31	27	56	51	37	35	21	20	28	11	12	62	11

Figura 44: Arquivo de custos utilizado no teste 15
Fonte: Autoria Própria (2014)

15	15													
994	175	463	181	684	862	210	912	618	535	812	670	177	442	347
353	163	127	256	117	64	808	994	825	327	43	815	992	544	260
285	531	388	968	816	73	55	540	884	642	340	831	646	64	505
613	966	199	61	782	896	231	760	534	915	176	106	262	216	981
663	956	495	27	832	102	279	300	645	956	517	879	922	855	56
987	408	668	658	876	630	318	577	444	328	139	86	22	30	221
101	871	488	309	711	886	992	619	229	317	512	832	301	472	15
319	330	534	94	785	87	814	231	313	388	916	480	143	300	339
364	393	318	77	963	143	347	585	692	701	695	221	24	254	932
524	496	919	107	441	1009	971	652	237	151	93	537	214	240	666
389	128	995	392	66	656	345	418	788	863	914	286	600	283	627
912	842	932	103	264	917	86	451	833	27	858	606	972	583	778
177	197	921	413	812	123	667	839	201	854	458	173	340	524	679
648	11	789	198	616	267	712	638	18	491	45	82	717	741	172
976	225	170	518	664	241	325	635	95	195	875	843	213	838	346
695	247	199	429	464	690	70	223	254	366	272	638	123	453	683
122	114	371	676	669	285	609	230	275	347	89	589	137	7	157
324	88	271	139	346	467	341	373	222	643	696	652	644	552	118
126	179	677	42	18	460	216	65	53	74	274	72	289	138	362
478	81	571	547	582	613	497	549	674	308	46	184	568	431	464
603	44	51	627	71	440	620	60	100	706	459	641	86	186	168
146	565	38	161	195	222	694	569	242	679	241	60	466	498	227
638	695	377	531	209	403	433	161	409	456	292	315	587	446	444
432	577	618	373	451	310	160	219	484	165	551	583	140	12	66
374	228	449	640	669	229	221	271	490	105	604	18	597	343	136
568	30	237	123	361	97	358	641	486	65	639	600	320	31	612
468	570	581	74	615	60	582	335	154	375	200	424	121	57	590
123	694	452	183	645	15	210	100	16	149	419	680	237	501	149
309	380	44	151	598	20	330	209	177	167	198	408	366	518	586
242	181	353	686	39	154	10	237	652	466	438	544	475	120	242

Figura 45: Arquivo de custos utilizado no teste 16
Fonte: Autoria Própria (2014)

10	10								
279	42	189	307	254	30	216	53	11	208
297	207	168	81	279	83	111	127	261	143
74	138	180	237	205	185	164	286	193	120
41	89	178	61	40	206	206	163	14	268
167	170	128	66	219	133	139	81	192	216
278	242	11	190	46	214	204	30	202	283
262	268	235	89	238	286	225	213	194	91
70	116	296	164	279	87	256	14	181	81
282	119	133	191	155	228	188	155	179	274
184	111	104	128	91	232	249	81	277	160
195	207	51	28	116	194	183	48	197	128
29	144	96	62	118	169	187	81	83	77
132	117	125	124	89	7	164	207	93	72
214	56	165	42	46	132	62	114	133	89
177	195	143	27	153	32	166	195	108	63
20	58	129	144	93	149	200	60	159	162
151	77	114	144	97	142	157	179	131	174
37	88	200	114	56	20	149	9	108	56
7	182	135	9	134	141	135	126	125	193
145	100	83	187	151	198	63	56	191	111

Figura 46: Arquivo de custos utilizado no teste 17
Fonte: Autoria Própria (2014)

10	10								
270	78	68	22	147	161	80	169	83	186
101	227	248	244	233	77	249	72	255	286
246	186	212	144	249	176	280	230	30	151
181	283	256	101	285	205	128	275	230	141
293	152	67	95	132	65	224	161	104	202
238	249	117	41	24	302	42	198	122	298
225	95	32	89	291	256	69	45	167	158
300	63	211	187	38	33	304	17	270	156
25	149	178	196	111	309	238	134	219	230
229	152	44	26	122	255	287	300	69	49
188	70	172	126	205	166	157	209	17	160
54	158	130	198	106	174	66	44	104	106
47	173	148	179	46	81	22	147	124	30
15	170	100	70	66	28	62	130	137	18
102	163	174	199	92	16	203	26	77	85
112	53	123	143	45	211	179	23	216	178
55	174	195	89	156	29	48	212	166	200
118	50	160	192	112	138	31	11	93	209
58	178	20	160	72	85	116	188	153	48
130	200	105	98	141	208	110	109	160	34

Figura 47: Arquivo de custos utilizado no teste 18
Fonte: Autoria Própria (2014)

10	10								
292	244	204	271	212	21	124	50	42	304
206	137	205	86	156	72	213	295	220	52
138	97	145	129	277	24	166	254	143	22
240	206	188	177	201	80	56	45	17	253
179	10	260	140	126	59	107	271	275	208
115	48	205	197	308	270	89	247	75	233
180	239	206	79	159	309	56	93	247	278
111	161	197	289	17	61	237	151	107	80
220	17	56	220	116	154	173	148	299	156
98	290	307	143	29	239	165	104	24	42
204	144	96	167	125	80	125	77	153	68
170	95	67	144	6	33	167	112	11	202
142	143	101	131	181	143	144	137	39	214
189	60	90	123	97	137	55	202	153	100
148	109	193	140	88	215	111	11	81	20
14	50	16	55	41	188	216	42	107	167
86	149	116	39	74	62	39	165	121	115
34	206	177	31	189	172	65	105	103	72
29	153	100	11	192	52	172	74	209	16
212	36	15	177	145	163	194	55	109	29

Figura 48: Arquivo de custos utilizado no teste 19
Fonte: Autoria Própria (2014)

10	10								
251	82	195	284	233	191	41	102	226	74
242	267	108	38	278	66	223	194	243	274
290	281	43	259	107	91	282	282	306	145
93	303	111	249	11	136	135	198	295	121
244	115	178	16	92	297	94	158	229	106
23	261	165	276	281	111	155	234	31	27
288	49	227	91	96	198	289	78	114	290
296	122	177	234	158	156	235	24	147	74
42	72	198	138	254	36	39	158	66	72
157	235	62	53	118	70	64	274	186	128
175	169	202	65	170	16	201	207	29	109
57	186	196	212	80	182	34	85	50	164
136	75	30	77	124	115	158	123	138	43
198	26	181	174	11	193	63	163	96	37
163	194	74	7	64	196	67	110	177	82
133	46	63	95	207	77	138	109	25	48
28	156	197	94	65	108	202	164	27	44
71	135	197	138	110	163	54	16	110	191
158	170	214	206	160	21	79	102	46	130
51	191	101	84	74	18	202	51	50	89

Figura 49: Arquivo de custos utilizado no teste 20
Fonte: Autoria Própria (2014)

10	10								
265	34	123	103	61	52	158	231	213	185
104	80	80	187	69	227	234	96	170	132
217	45	282	115	154	58	125	119	39	181
15	83	106	177	230	91	68	93	182	167
70	165	19	234	288	300	92	262	221	235
151	75	171	85	296	21	283	60	43	101
298	262	209	221	37	151	282	11	182	183
240	292	273	42	50	59	18	20	83	305
287	70	265	114	120	270	165	156	117	71
257	105	72	267	63	151	243	221	236	47
185	72	151	10	48	105	208	167	200	179
23	55	31	58	115	52	183	204	48	73
86	55	197	74	13	119	146	191	185	50
72	130	80	123	163	59	154	29	79	186
42	48	107	160	201	207	25	34	83	44
36	158	40	63	209	14	105	41	188	105
110	163	87	47	64	198	197	12	115	170
161	67	83	65	183	41	7	13	109	154
149	118	27	127	154	30	127	58	81	165
129	92	126	116	164	70	128	213	49	32

Figura 50: Arquivo de custos utilizado no teste 21
Fonte: Autoria Própria (2014)

10	10								
187	246	65	262	64	302	17	205	242	268
255	88	33	249	126	49	160	169	262	289
128	94	277	245	71	300	56	178	234	277
277	221	70	247	43	65	272	210	195	242
303	144	179	302	233	142	12	258	93	77
91	292	128	88	246	18	191	192	189	118
232	126	287	179	54	27	238	54	102	246
107	46	264	241	103	81	186	113	263	136
283	72	175	221	155	307	225	215	142	221
306	281	134	101	277	179	85	79	167	41
130	178	89	193	212	63	162	74	198	214
172	61	65	154	100	204	88	32	50	196
45	23	193	48	125	89	200	184	122	93
183	174	171	172	211	61	125	168	154	70
44	88	49	30	163	172	37	72	108	193
211	34	209	45	99	12	18	56	214	125
11	111	39	190	8	133	166	130	157	59
143	118	124	146	180	134	37	79	150	55
169	183	163	136	65	132	71	184	99	116
187	202	193	169	53	82	172	95	154	28

Figura 51: Arquivo de custos utilizado no teste 22
Fonte: Autoria Própria (2014)

10	10								
130	93	153	108	117	50	291	171	146	28
35	39	101	69	93	139	48	108	233	194
159	79	268	228	203	299	298	220	20	296
29	39	200	19	40	16	121	148	141	52
62	181	52	168	296	42	166	212	86	188
161	47	287	291	108	305	267	119	245	175
281	202	17	169	39	126	175	280	147	66
89	103	188	241	163	47	232	167	34	62
85	135	152	308	21	133	129	31	287	113
230	13	266	272	307	139	232	292	110	209
90	24	111	20	43	112	196	62	59	160
65	27	55	27	126	32	141	72	94	9
107	70	187	139	36	200	11	131	106	186
75	48	159	13	117	203	118	168	215	190
81	65	142	27	207	75	27	114	14	214
34	97	209	11	29	213	88	32	93	97
203	33	208	84	116	186	122	162	90	162
119	75	153	103	148	83	195	116	21	204
102	163	13	98	60	171	102	23	200	76
19	135	207	36	131	122	46	43	79	146

Figura 52: Arquivo de custos utilizado no teste 23
Fonte: Autoria Própria (2014)

10	10								
169	238	299	55	116	162	181	193	161	177
72	174	94	139	123	242	237	89	252	243
122	116	136	106	41	214	301	308	151	300
64	154	196	119	265	22	162	266	65	44
36	104	52	266	33	26	159	107	201	242
88	146	132	181	47	30	198	207	183	52
79	253	117	29	252	104	256	272	189	55
18	39	128	156	258	73	262	255	195	66
209	300	212	152	258	68	252	173	132	21
268	197	201	277	78	189	271	12	304	36
118	50	85	44	25	61	55	12	146	187
166	121	81	107	72	102	177	27	209	137
209	65	95	137	36	92	81	89	148	140
38	97	74	83	186	126	20	109	106	193
81	86	28	185	23	32	176	180	180	54
113	169	149	15	18	20	72	51	47	132
126	165	210	113	111	138	179	183	176	189
135	62	215	186	74	144	190	178	121	8
112	176	105	45	140	128	132	136	92	212
123	170	209	30	169	36	38	46	14	25

Figura 53: Arquivo de custos utilizado no teste 24
Fonte: Autoria Própria (2014)

10	10								
46	242	228	222	215	208	47	295	121	195
146	305	38	71	47	299	72	196	293	204
186	240	144	29	55	164	189	194	232	32
218	226	111	288	185	266	289	74	273	71
126	277	173	203	297	114	88	274	105	69
113	241	274	58	64	186	209	246	291	21
59	100	90	126	75	307	21	57	202	222
36	247	50	101	145	173	151	45	292	154
105	159	67	99	102	197	291	199	283	54
187	61	156	248	179	184	249	94	140	138
32	102	130	152	88	79	41	25	73	130
169	213	167	158	193	168	69	172	111	42
159	26	100	77	121	191	62	34	46	109
155	49	20	201	142	40	88	70	69	173
150	32	38	129	207	44	52	101	71	125
145	209	114	186	79	130	214	121	137	128
32	50	132	202	61	146	14	105	203	174
206	137	135	51	191	172	39	31	139	65
84	205	162	191	73	203	141	204	198	97
136	142	22	49	48	14	155	107	37	96

Figura 54: Arquivo de custos utilizado no teste 25
Fonte: Autoria Própria (2014)

10	10								
177	237	258	232	51	68	173	104	282	119
166	94	233	215	112	154	305	153	156	142
37	306	80	127	74	71	249	84	182	166
288	159	210	195	157	223	94	191	297	93
236	47	81	133	42	155	306	96	172	256
104	104	53	190	38	171	187	294	307	131
253	37	234	197	192	305	146	261	266	136
84	297	187	301	182	298	272	302	85	140
169	215	190	48	149	52	164	266	17	127
125	47	45	262	158	277	14	164	99	193
123	116	25	201	165	72	177	58	118	87
165	65	214	111	32	72	25	207	150	32
180	163	55	146	56	37	163	130	132	31
162	150	88	136	93	132	137	210	33	183
35	78	51	109	29	26	134	127	104	110
47	107	49	156	108	119	213	208	36	193
121	213	174	65	214	130	102	190	114	9
72	107	58	133	67	205	182	211	186	114
197	109	127	207	120	214	186	59	11	69
83	99	116	65	179	91	95	97	88	135

Figura 55: Arquivo de custos utilizado no teste 26
Fonte: Autoria Própria (2014)

10	10								
201	85	239	36	286	99	126	116	129	252
223	186	126	201	181	89	179	42	90	157
160	23	219	144	76	279	44	277	146	172
70	45	58	82	62	179	106	43	267	33
236	115	51	91	258	226	218	120	252	166
39	211	252	149	74	60	108	251	262	190
231	123	119	27	13	220	63	62	61	262
98	289	68	86	246	23	307	226	232	295
303	252	212	18	60	206	171	218	29	170
111	175	114	278	139	34	155	191	56	224
140	156	111	48	165	27	161	68	212	77
59	130	16	31	80	147	86	202	176	122
167	88	153	40	35	176	83	47	148	79
25	140	100	57	63	104	18	60	12	194
200	126	53	43	180	51	9	172	41	97
69	62	195	125	158	41	153	16	144	23
88	125	30	74	152	75	44	214	119	108
81	29	193	30	83	175	43	158	152	133
90	62	102	186	176	183	42	162	20	39
176	109	120	23	116	132	183	206	118	156

Figura 56: Arquivo de custos utilizado no teste 27
Fonte: Autoria Própria (2014)

10	10								
36	58	17	131	106	288	133	174	124	226
273	49	97	107	244	240	102	34	204	173
227	80	160	47	231	36	145	57	219	42
197	121	83	106	70	24	18	142	123	286
83	127	252	93	177	115	210	175	132	279
93	269	164	240	69	23	201	135	170	88
32	97	84	48	117	290	53	305	261	62
105	118	243	229	281	259	249	111	283	188
60	240	237	92	72	17	242	250	309	108
116	105	210	304	284	273	254	123	24	228
25	191	158	137	58	65	22	73	41	81
40	34	55	84	88	188	67	82	167	73
11	67	111	58	176	114	58	170	165	146
91	74	32	74	65	167	33	160	64	212
74	170	161	48	123	48	81	196	50	198
201	167	25	16	80	16	202	181	11	191
93	71	101	12	146	140	37	174	169	177
121	23	39	99	122	94	213	77	174	86
86	142	153	86	92	118	182	198	216	16
158	121	29	200	195	61	43	131	75	159

Figura 57: Arquivo de custos utilizado no teste 28
Fonte: Autoria Própria (2014)

10	10								
274	150	91	271	119	257	81	94	154	224
63	293	163	234	176	279	120	78	274	19
186	147	193	175	57	36	52	206	120	305
49	83	285	177	12	62	178	191	272	210
64	186	169	239	38	100	309	294	201	106
302	187	79	208	67	55	110	185	83	75
159	268	254	247	110	30	305	266	72	198
204	65	174	128	70	297	287	38	113	294
289	140	38	122	264	304	20	102	263	73
148	35	83	96	167	88	46	167	11	52
191	44	130	34	44	211	111	142	202	103
104	205	102	58	130	130	187	45	97	24
63	114	135	199	118	55	177	121	26	58
189	163	122	123	167	145	172	89	85	67
83	123	39	8	26	46	76	48	184	116
179	195	25	43	69	38	20	207	212	61
56	83	36	124	216	76	213	200	13	32
65	54	144	133	205	129	186	26	71	116
107	191	83	190	140	58	50	79	184	7
156	13	213	146	74	52	138	205	51	36

Figura 58: Arquivo de custos utilizado no teste 29
Fonte: Autoria Própria (2014)

APÊNDICE B: ARQUIVOS DE ENTRADA PARA TESTES: PEDIDOS.

```
50
9 8
1 8
2 2
5 2
3 6
6 9
4 2
7 2
2 9
7 6
8 1
3 9
5 8
9 6
8 3
8 5
4 3
2 5
2 5
4 1
2 4
1 2
2 6
6 7
4 2
1 8
2 3
0 7
1 2
3 9
6 7
1 0
8 4
7 6
3 5
4 1
3 8
3 4
3 0
2 6
1 0
9 0
7 5
5 8
5 1
2 6
0 9
7 0
6 9
8 7
```

Figura 59: Listagem de pedidos utilizado nos testes 0 a 9
Fonte: Autoria Própria (2014)

```
1
2
```

Figura 60: Quantidade de veículos e posição inicial para teste 0
Fonte: Autoria Própria (2014)

```
2
2 0
```

Figura 61: Quantidade de veículos e posição inicial para teste 1
Fonte: Autoria Própria (2014)

```
3
2 1 0
```

Figura 62: Quantidade de veículos e posição inicial para teste 2
Fonte: Autoria Própria (2014)

```
4
2 1 1 0
```

Figura 63: Quantidade de veículos e posição inicial para teste 3
Fonte: Autoria Própria (2014)

```
5
2 2 0 1 5
```

Figura 64: Quantidade de veículos e posição inicial para teste 4
Fonte: Autoria Própria (2014)

6
2 2 5 1 4 5

Figura 65: Quantidade de veículos e posição inicial para teste 5
Fonte: Autoria Própria (2014)

7
2 0 2 5 1 4 5

Figura 66: Quantidade de veículos e posição inicial para teste 6
Fonte: Autoria Própria (2014)

16
2 2 5 1 4 5 0 2 1 5 4 7 2 1 0 1

Figura 67: Quantidade de veículos e posição inicial para teste 7
Fonte: Autoria Própria (2014)

17
2 2 5 1 4 5 0 2 1 5 4 7 2 1 0 5 2

Figura 68: Quantidade de veículos e posição inicial para teste 8
Fonte: Autoria Própria (2014)

20
2 2 5 1 4 5 0 2 1 5 4 7 2 1 0 5 2 1 5 0

Figura 69: Quantidade de veículos e posição inicial para teste 9
Fonte: Autoria Própria (2014)

```
60
38 31
1 13
33 38
34 34
13 30
21 18
11 22
3 33
1 8
10 23
26 33
0 32
20 25
39 23
3 4
10 4
4 0
23 27
36 39
37 23
34 31
8 14
27 1
11 23
29 15
5 10
37 25
16 31
21 34
37 33
24 30
12 4
24 22
3 13
2 32
12 1
0 17
28 25
24 12
33 28
10 32
36 37
0 4
39 38
17 30
33 38
39 13
15 15
10 19
29 3
27 7
22 16
22 2
17 1
26 17
36 32
10 10
5 33
9 18
22 0
5
2 6 25 37 6
```

Figura 70: Arquivo de pedidos utilizado no teste 10

Fonte: Autoria Própria (2014)

```
60
 1  10
 2  19
13  19
 8   5
14  12
19   4
 1   3
15   8
 1   0
 8   8
19   1
19   5
12   7
19  17
 6   9
16  19
10  10
 7  13
17  16
 6  14
 1   2
13  13
15   2
 1   9
 3  15
12   7
16  16
10   1
 8   6
10  17
18  10
 8   1
10  11
17  19
 8   2
 3  12
 6   3
11  13
15   7
12   6
 4  14
10  11
 0   3
15  19
 6   2
12  10
19   5
 3  10
 1  10
12   6
12  16
10  19
 6  12
16  10
 0   3
15   6
 7   1
17   2
16   7
 2  14
 5
 8  11  18  6  15
```

Figura 71: Arquivo de custos utilizado no teste 11
Fonte: Autoria Própria (2014)

```
60
24  4
18  0
18  0
 4  1
31  4
48  4
44  0
10  2
20  1
33  0
 6  4
17  3
25  0
18  3
47  0
13  1
45  1
44  4
34  1
30  3
12  3
44  0
 4  1
47  4
36  4
36  0
 9  1
20  0
32  2
25  3
 7  0
10  3
34  4
39  2
22  0
10  2
23  3
23  1
 4  2
17  2
49  1
48  2
19  2
 0  4
35  4
37  3
46  0
14  4
26  0
 6  4
26  4
 7  3
44  2
31  2
47  0
 9  1
22  2
14  3
15  3
 2  3
 5
 4  2  2  3  4
```

Figura 72: Arquivo de custos utilizado no teste 12

Fonte: Autoria Própria (2014)

60				
3	33			
4	4			
3	11			
2	20			
1	42			
2	49			
3	40			
1	32			
0	3			
1	42			
2	14			
3	7			
1	37			
1	26			
1	37			
3	31			
3	9			
1	33			
2	23			
1	33			
3	49			
4	39			
1	20			
0	5			
1	38			
4	8			
1	6			
3	11			
3	2			
2	48			
4	38			
3	11			
4	38			
0	38			
3	26			
0	24			
4	43			
0	5			
2	32			
2	26			
1	43			
3	28			
2	42			
2	34			
3	24			
0	20			
1	38			
3	29			
3	9			
3	29			
3	32			
4	27			
0	31			
1	24			
4	5			
0	35			
4	20			
3	18			
3	23			
4	42			
5				
16	25	4	9	16

Figura 73: Arquivo de custos utilizado no teste 13

Fonte: Autoria Própria (2014)

```
 7 5
13 13
 4 14
 9 12
 1  5
 4  4
13 12
 0 10
 8  4
 6  0
 1 12
14  4
 7  9
 4 14
12  5
14 12
 2  8
14 14
 1 14
14  7
13 11
 5  5
 1  3
 8  8
10  9
 8  6
 0 13
 8 14
 2 14
11  5
 1  8
10  3
14  8
12 13
14 10
10 14
 9 11
 2  1
 7  2
11  0
 0 10
 5 13
 2  3
 2  5
13 14
 8  9
 2  2
14  2
 7 12
12  7
12  1
 1  6
12 11
10 10
 7 10
 4 12
 6  3
 1  8
13 11
 3 11
10  3
13  5
12 13
 2  3
 2 14
11  6
 0  2
13  2
 4  6
 0  4
 0  8
13  9
 6 14
 8  1
12 14
 5 13
 2
 5  2
```

Figura 74: Arquivo de custos utilizado no teste 14

Fonte: Autoria Própria (2014)

```
75
 3   5
 1   2
 2   1
 8   0
 3   7
 7   1
 8  10
 3   1
12   1
11   5
 1   8
 6  14
 2   1
 5   1
 5   0
 1   1
 7  12
12   7
 3   9
 2  11
 9   4
 8   8
14   6
 5   5
 7   9
 8   2
 0  14
 3   0
12  10
 5   5
14   5
 0  14
13   1
 9   4
13   1
 5  10
 8  11
 7   2
 1   0
 1   6
12  14
 1  11
 7  10
 5   9
 8  10
 5   1
14  13
13  14
13  10
 2   7
14   1
 2  10
12   5
12  14
 8   9
 0   8
14   5
10   3
 6   0
11   6
 7   1
 8  14
 8   5
12  13
13   1
 0   3
 1  10
 6   2
 0   4
 5   6
 5   9
 7   5
 6   3
 0  13
 3  13
 2
13   6
```

Figura 75: Arquivo de custos utilizado no teste 15

Fonte: Autoria Própria (2014)

```
75
12 10
12 14
9 5
11 7
1 14
1 10
5 7
11 0
4 0
5 7
4 13
5 5
12 0
2 2
8 8
6 1
5 4
13 12
14 7
9 4
10 4
11 2
10 12
1 13
12 1
14 4
10 4
14 10
11 5
3 4
13 9
11 9
0 14
14 4
1 0
2 14
4 4
7 5
6 5
14 4
14 9
14 12
11 9
2 0
0 11
6 0
13 5
5 8
14 12
11 6
0 6
7 14
6 12
0 0
4 1
2 3
8 9
14 12
13 0
13 12
3 0
6 4
13 0
7 10
11 11
11 3
3 6
8 8
12 4
0 8
9 0
9 0
13 13
14 7
11 1
2
13 8
```

Figura 76: Arquivo de custos utilizado no teste 16
Fonte: Autoria Própria (2014)

```
80
 7  0
 3   1
 2  0
 0  7
 6  9
  1 3
 5  2
 9   1
 4  5
 7  0
  1 7
 7  5
 3   1
 4  0
 5  8
 9  4
 4  5
 3  0
 2  5
 2   1
 6  4
 4   1
 3  5
 2  2
 9  3
 5  9
 6  7
 0   1
 5  7
 9  0
 2  2
 8  0
 9  2
 6  8
 5  8
 3  3
 8   1
 4  9
 6  3
 3  3
 4  4
 2  8
 9  7
 2  5
 4  6
 7  4
  1 4
 3  4
 3   1
 3  7
  1  1
 2  4
 5   1
 0  7
 5  9
 5  6
 4  2
 6  7
  1 3
 5  0
 9  5
  1 2
 3  9
 5  7
 2  3
 4  4
 2  4
 4   1
  1 0
 2  8
 4  9
 3  8
 2  0
 5  2
 0  9
 8  3
 2  7
  1 0
 6   1
 4  9
  1
  1
```

Figura 77: Arquivo de custos utilizado no teste 17
Fonte: Autoria Própria (2014)

```
85
 3  7
 7  6
 8  7
 0  2
 2  9
 4   1
 7  7
 7  5
 0  5
 5  2
 8  3
  1   1
 4  5
 7  3
 7  7
 9  4
 3  2
 3  7
 4  9
 5   1
  1  9
 6  6
 6  0
 3   1
 6  9
 4  7
 3  7
 4  9
 0  4
  1  0
 6  7
 3  8
 5  2
 4  7
 7   1
 0   1
 0  0
 4  8
  1  4
 0  4
 5   1
 8  9
 5  7
 2  9
 4  6
 0  7
 6   1
 8  2
 3   1
 6  7
 0  5
 4   1
 9  9
 7   1
 8  9
 2  5
 4  9
 5  7
 3   1
 3  6
 6  3
 3  3
 7  6
 9  6
 0  3
 2  7
 8  9
 9  0
 3  9
 3  4
 4  4
  1  7
 4  9
 9  8
 2  7
 5   1
 5  5
 3  7
 6  2
 3  9
 7  2
 9  0
 0  9
 5   1
 5  5
  1
  1
```

Figura 78: Arquivo de custos utilizado no teste 18
Fonte: Autoria Própria (2014)

```
90
 5 4
 2 9
 5 2
 3 9
 9 3
 7 5
 8 5
 3 0
 6 9
 8 3
 5 5
 8 0
 4 9
 0 5
 1 0
 7 3
 4 9
 2 3
 5 0
 9 0
 2 1
 7 5
 1 3
 6 0
 6 9
 3 1
 3 5
 8 9
 6 8
 2 3
 8 2
 1 8
 5 6
 7 3
 7 4
 4 7
 8 5
 9 2
 9 0
 6 4
 0 5
 3 2
 5 8
 0 8
 1 2
 3 5
 4 0
 9 1
 4 7
 7 5
 2 1
 4 3
 1 1
 4 2
 2 7
 9 8
 2 2
 3 2
 6 0
 7 0
 3 2
 5 3
 7 8
 4 3
 3 5
 4 5
 0 2
 8 0
 1 1
 9 3
 3 2
 4 1
 4 9
 0 9
 7 6
 7 5
 3 9
 1 1
 7 4
 4 0
 8 2
 2 1
 7 5
 7 8
 2 2
 7 7
 3 8
 2 7
 2 7
 5 1
 1
 7
```

Figura 79: Arquivo de custos utilizado no teste 19
Fonte: Autoria Própria (2014)

100

1	8	4	2
1	9	0	9
1	2	8	9
1	1	1	4
7	5	6	0
2	9	0	4
9	0	3	5
5	4	5	7
2	3	2	4
3	5	9	7
8	8	8	4
8	7	8	4
4	4	0	2
0	8	0	6
5	5	2	6
7	5	8	7
0	4	5	3
0	5	2	4
7	1	4	3
2	9	3	0
2	5	4	0
4	0	5	9
1	2	4	5
2	5	8	7
8	3	2	9
3	4	0	1
8	9	7	3
2	3	2	9
1	1	9	2
6	5	3	3
3	9	9	3
4	0	8	9
9	5	3	6
0	2	5	5
1	6	0	5
1	8	2	1
6	1	7	2
3	4	8	6
0	9	7	8
4	9	3	8
5	6	6	1
1	3	5	9
1	7	7	7
9	7	9	6
5	8	3	7
6	7	8	3
6	5	4	4
0	9	5	0
9	4	1	
2	5	0	
8	0		
0	6		

Figura 80: Arquivo de custos utilizado no teste 20
Fonte: Autoria Própria (2014)

```
200
 0 2    0 3
 5 9    4 0
 2 9    8 9
 2  1   5 4
 5 0    4 7
 0 9    0 6
 4 2    2 6
 6  1   6 2
  1 5   7 3
 6  1   7  1
 7  1   6 8
 5 0    2 2
 9 7     1 2
 8 7    4 6
 9 8    8 6
 8 7    5 3
 0 7    3 9
 0 2    8  1
 2 2    4 3
  1 8   7 3
 6  1   2 6
 0 0    9 9
 9 0    3 3
 0 0     1 7
 2 7    7 6
 9 5    5 7
 7 5    7 2
 3 8    5 0
 4 2    9 8
  1 9   3 2
 2  1   5 5
 6 2    3 0
 2 4    8 8
 3 7    3 5
  1  1  4 9
 4 2    2 2
 3 5    8 2
 6 6    9 7
 5 2    0 5
 6 7    2 4
 3 5    8 9
 7  1   5 2
 5 7    8 5
 5 3    6 0
 7 6    2  1
 0 3    0  1
 5 3    7 5
 6 8    8 5
 4 0    7  1
 3 5    3 7
  1 5   8 0
 8 9    2 7
 0 7    5 0
 3 7    8 5
 9 5    6 9
 3 5    4  1
 3 4     1 2
 8 2    2 2
 0 4    0 3
  1 5    1  1
 2 2    8 4
 4  1   0 3
 7 3    7 4
 9 0    5 7
 5 0    7 8
 5 9    7 7
 5 8    3 5
 3 3    5 3
 5 9     1 0
 5 2    8 6
 5 4    6 5
 7 0    8 3
  1 6   2 3
 0 4    8 8
 2 7    2 0
 5 2    6 6
 4 3    6 8
 5 4    0 9
 3 9     1 2
 5  1   9 7
 7 7    3  1
  1 7   2  1
 4 5    4 5
 0 0    9 5
 7 9     1 5
 7 5    9 6
 9 6    2  1
 5 7    6 3
 6 6    8 4
 7  1   3  1
  1 0   6 9
 2 3    8 9
 3 3    2  1
 4 4    6 2
 0 6     1 9
 2 7    6 5
 6 6    7 8
 3 8    3 8
 4 3    9 3
        5 8
        2 7
         1
        3
```

Figura 81: Arquivo de custos utilizado no teste 21
Fonte: Autoria Própria (2014)

1000

5 8	6 1	4 5	9 2	8 3	2 8	5 2	8 3	6 3	6 4	1 1
1 1	3 5	8 0	7 4	0 0	7 9	9 6	6 4	5 1	7 2	7 6
9 4	6 8	6 3	1 9	2 4	9 3	9 0	0 3	0 2	9 4	8 0
9 3	6 3	9 4	0 8	6 4	9 4	5 2	7 7	9 7	9 2	4 2
8 3	8 4	8 8	0 2	0 5	6 6	1 3	9 7	6 3	4 5	9 3
1 5	5 0	7 5	0 8	2 6	5 0	2 2	5 8	5 1	6 3	9 0
5 0	6 4	7 1	0 5	5 7	8 0	2 1	1 2	1 9	8 9	3 6
0 7	6 5	7 6	4 4	6 2	4 5	0 4	1 5	6 9	6 5	9 2
0 7	6 8	8 5	6 5	1 5	7 4	1 7	7 1	8 2	8 4	4 1
3 0	1 9	3 5	2 4	3 7	2 1	6 3	5 3	9 0	2 3	9 5
9 2	5 3	8 5	0 1	5 3	8 3	3 9	3 9	0 8	7 5	2 4
9 4	3 5	4 5	1 8	9 3	6 2	9 5	4 0	8 5	1 7	6 6
9 9	4 4	0 9	8 9	2 1	5 9	6 1	8 3	2 6	3 9	7 3
1 1	0 0	0 0	6 7	7 0	6 1	5 0	6 2	4 7	4 8	3 2
8 4	8 6	2 0	8 9	8 8	5 6	4 5	7 1	0 1	6 6	9 2
6 5	2 2	2 6	9 7	2 2	3 1	6 0	9 3	0 3	6 0	8 2
6 1	3 9	3 1	5 7	7 2	3 1	1 5	3 9	6 7	1 2	5 0
4 5	1 4	6 2	7 1	1 8	0 4	8 7	4 9	6 8	5 6	5 5
9 0	1 6	0 1	9 2	0 4	9 8	4 9	4 7	7 6	9 1	7 6
3 7	8 2	5 0	3 7	3 3	6 3	5 6	4 8	2 6	2 4	9 1
2 4	7 8	7 9	9 2	1 5	2 4	9 4	4 3	6 0	3 1	1
8 7	7 3	9 7	8 9	7 7	2 0	0 2	6 4	0 4	5 1	5
9 4	1 7	6 7	2 4	8 9	9 0	7 5	0 6	9 3	8 8	
7 2	6 0	7 1	2 6	1 6	0 2	0 3	0 6	9 6	6 0	
4 4	5 0	2 3	0 3	2 5	7 7	6 7	6 3	9 1	2 2	
3 6	3 1	8 2	4 9	5 5	0 5	2 6	9 9	2 0	7 9	
1 1	7 9	4 9	6 0	3 6	6 1	3 0	7 6	1 2	9 6	
0 9	2 1	8 9	4 7	9 3	4 2	7 7	1 5	0 9	4 2	
5 9	5 0	1 6	6 8	7 8	4 8	0 4	3 9	7 2	8 1	
5 8	2 3	5 9	4 5	7 4	2 4	6 3	6 3	8 8	1 4	
6 1	1 9	8 3	7 3	9 9	0 7	8 2	7 9	0 0	8 6	
4 1	8 9	3 9	2 8	3 6	7 4	9 0	3 1	7 2	1 4	
0 5	9 4	8 1	0 2	3 6	5 9	8 5	9 3	0 3	4 2	
2 1	8 3	5 0	2 6	3 8	6 9	4 8	7 7	6 7	5 9	
2 2	6 2	0 1	3 4	7 6	0 2	1 6	2 3	2 0	0 9	
5 1	0 1	5 2	5 4	2 9	9 5	3 5	7 3	2 5	5 3	
9 3	0 9	4 1	4 2	3 6	0 7	6 4	6 1	0 9	0 0	
2 7	9 8	3 3	4 5	3 0	7 3	5 3	7 4	0 0	6 3	
0 5	9 9	2 6	8 0	6 6	4 2	4 3	1 1	5 0	9 5	
7 2	5 0	5 9	0 0	0 8	3 0	7 2	6 9	3 5	9 2	
8 8	0 7	3 3	5 7	2 8	7 7	1 2	2 0	9 1	4 2	
5 9	4 7	0 3	5 4	2 0	9 1	1 9	5 7	3 6	9 0	
3 0	3 6	8 1	7 7	9 2	2 7	4 3	9 0	5 6	6 8	
4 0	4 6	0 9	0 7	9 0	3 4	1 0	7 4	2 6	8 3	
1 5	5 8	4 9	2 6	7 8	5 9	5 8	0 5	1 9	5 9	
1 6	8 0	3 0	5 0	7 1	5 0	6 0	8 8	6 8	0 8	
5 8	7 0	3 2	7 7	9 1	4 7	8 7	7 0	7 2	2 9	
7 1	0 2	2 9	2 8	8 5	1 4	1 8	6 4	5 4	4 8	
7 9	1 4	0 7	3 4	5 6	0 5	0 0	5 2	6 4	7 1	
5 9	9 7	5 5	5 4	6 2	7 4	5 8	1 9	5 6	8 7	
5 8	2 7	5 8	2 2	2 8	2 6	0 5	4 1	5 9	9 2	
0 3	2 1	6 4	3 9	5 9	6 2	7 6	4 3	9 5	0 1	
2 7	6 7	5 7	6 8	5 5	3 3	9 1	2 6	7 5	5 2	
1 1	5 9	5 3	1 4	9 4	3 9	0 6	8 5	9 6	5 5	
5 3	9 7	2 4	1 2	2 0	8 3	3 7	8 1	0 5	9 5	
2 4	3 0	7 8	0 5	7 1	3 4	1 3	2 7	0 8	3 4	
0 3	1 9	5 8	0 9	9 5	6 9	2 2	2 1	5 2	2 6	
5 9	6 2	1 6	6 6	9 1	9 5	2 4	9 7	1 8	2 2	
0 9	5 8	5 7	0 2	5 0	1 5	0 2	6 9	3 4	8 0	
3 1	7 8	0 1	1 2	1 9	5 1	9 5	3 7	1 6	7 4	
9 8	1 5	0 0	9 4	1 1	8 6	5 0	3 6	1 3	9 6	
1 5	8 4	4 6	5 2	3 4	5 8	5 1	8 4	7 8	1 7	
5 7	6 3	0 2	7 3	7 8	8 3	9 8	8 6	2 3	8 5	
1 5	0 9	1 0	3 9	7 9	9 9	1 8	5 0	2 9	7 2	
8 8	8 8	1 8	9 1	8 4	0 2	0 9	7 1	2 8	1 9	
7 3	9 2	0 3	5 6	4 8	6 3	1 6	1 8	3 6	2 6	
7 8	8 3	4 1	1 2	0 5	6 0	6 8	1 2	2 3	9 4	
8 4	8 2	2 1	3 2	0 9	7 2	4 3	0 0	4 8	3 8	
8 8	6 4	9 9	1 4	6 3	4 7	1 0	7 5	1 7	3 4	
9 3	5 4	2 5	5 5	6 9	2 7	7 4	7 7	1 1	2 0	
5 5	9 6	6 9	2 5	2 6	8 6	9 1	0 3	7 7	3 0	
6 5	8 0	9 6	3 0	4 0	6 1	2 3	6 6	6 1	8 5	
4 5	5 6	3 6	4 3	6 0	2 3	0 9	8 5	6 6	4 2	
4 3	8 4	6 7	2 6	6 0	1 9	7 6	7 3	9 4	7 0	
5 5	7 5	0 1	4 7	8 9	0 7	6 6	0 3	5 6	4 6	
6 3	1 2	1 2	4 9	4 0	8 4	7 0	9 3	5 7	9 0	
8 3	9 9	2 9	4 9	5 9	9 1	5 8	3 4	0 8	7 3	
1 8	6 5	6 4	4 8	2 4	4 4	3 0	6 3	2 4	0 2	
9 3	3 9	9 1	1 4	6 2	3 9	4 5	0 9	7 8	6 2	
1 1	6 0	4 3	8 8	6 0	3 9	8 5	4 0	6 4	6 6	
7 0	1 6	4 7	7 9	7 8	1 1	8 5	2 8	6 3	7 3	
7 2	4 4	9 4	2 7	9 3	7 2	4 0	1 2	2 5	0 1	
0 2	3 4	3 0	9 0	8 2	4 5	6 5	7 8	8 3	1 7	
6 0	0 0	6 1	0 2	6 7	6 1	8 2	5 2	0 0	9 5	
1 9	7 3	2 8	4 8	3 4	4 2	9 7	8 0	5 0	3 9	
0 4	9 2	3 6	0 8	6 5	0 5	8 3	2 4	7 4	5 3	
5 4	5 8	9 2	4 2	3 3	7 2	0 3	7 4	2 0	8 5	
6 5	4 2	8 8	5 2	1 7	2 0	7 0	3 6	7 5	0 1	
4 2	1 3	5 3	8 6	2 4	0 7	8 3	3 4	8 3	7 5	
1 7	3 5	9 1	3 0	6 3	1 6	8 5	6 2	9 7	8 6	
4 4	5 8	0 9	2 9	7 3	4 4	8 0	4 4	4 2	1 0	
2 5	6 1	3 1	1 8	6 6	3 0	3 6	2 7	4 8	4 5	
6 0	2 8	3 1	6 3	5 4	7 6	6 6	5 1	1 2	9 7	
6 6	4 3	3 8	2 2	2 6	6 8	8 3	2 5	7 6	9 5	
5 7	6 9	1 7	3 1	7 2	6 4	6 8	2 5	1 6	9 2	
6 6	3 5	7 6	1 9	5 5	0 3	0 9	8 5	9 6	1 1	
5 6	0 0	2 6	1 1	9 5	8 6	0 8	5 8	5 5	8 7	
7 1	1 0	6 5	3 1	2 6	3 2	8 0	0 7	9 2	4 4	

Figura 82: Arquivo de custos utilizado no teste 22

Fonte: Autoria Própria (2014)

```
400    1 2   5 8   3 5
7 6    6 6   8 7   7 9
5 7    9 0   6 1   7 5
1 1    8 2   4 7   5 6
8 8    7 0   6 8   7 5
8 7    3 5   6 5   7 7
2 4    5 5   7 7   2 1
1 3    0 0   9 8   6 1
6 0    9 4   4 3   0 6
4 4    0 7   6 5   5 9
8 5    5 5   7 2   0 5
4 7    9 9   9 4   6 8
8 8    9 2   5 2   3 7
1 8    6 4   7 0   3 5
6 7    5 3   8 4   6 5
6 5    1 6   8 8   3 7
2 6    9 6   6 5   4 4
2 0    0 7   3 0   0 7
2 8    1 7   6 9   9 2
6 3    4 8   6 2   8 6
7 4    3 4   2 8   4 3
5 4    6 7   7 9   8 8
0 1    4 8   7 3   6 5
7 1    8 6   2 6   6 2
7 9    2 6   6 2   9 2
7 8    2 1   1 6   9 0
7 2    1 1   4 6   9 3
6 7    8 4   5 0   6 1
5 9    6 9   0 0   3 6
7 8    1 8   6 3   9 7
4 4    3 8   0 5   0 3
3 4    4 9   7 1   0 5
8 4    6 1   9 1   5 7
1 0    5 3   0 0   0 5
0 0    2 5   7 9   2 9
1 5    1 6   1 1   2 8
7 7    9 9   8 8   1 8
0 7    3 0   3 7   5 0
0 9    2 6   1 0   1 2
6 2    2 7   7 5   4 4
0 6    7 5   0 5   0 8
2 3    5 8   2 7   0 8
2 5    1 5   8 0   8 5
5 1    3 3   9 3   0 8
5 4    9 9   6 9   6 0
3 6    6 7   8 0   0 4
0 2    5 8   3 7   8 9
4 9    0 7   9 7   7 0
6 6    5 6   0 4   8 8
8 8    4 9   5 7   2 2
2 2    6 7   5 7   8 1
9 9    5 3   2 5   2 8
1 9    9 0   4 9   1 3
1 0    8 5   9 7   8 6
3 2    1 0   4 7   5 8
0 2    4 5   4 1   7 1
3 6    0 6   0 7   2 4
5 2    0 6   5 1   9 2
4 2    1 9   2 5   9 2
0 9    0 2   6 6   7 5
1 8    7 7   7 2   8 3
5 2    6 3   0 3   2 5
4 9    1 4   4 9   0 4
9 4    1 2   0 2   6 3
0 7    8 2   4 7   4 4
2 6    5 6   2 1   4 4
0 7    7 8   1 7   7 5
7 9    4 1   2 2   2 4
9 2    7 3   8 8   0 4
0 6    8 5   9 7   7 8
1 9    0 6   8 7   6 6
3 4    0 8   5 1   4 2
4 0    9 4   3 5   9 9
3 5    4 7   7 1   1 8
1 2    6 4   6 7   8 1
8 3    5 4   1 9   0 4
5 7    0 0   9 3   7 0
2 7    6 2   9 9   6 4
6 8    6 3   1 0   8 1
7 3    1 6   7 9   7 7
8 5    9 6   5 4   5 0
8 7    9 1   6 8   4 1
6 5    6 2   1 8   9 5
7 0    9 9   1 7   9 4
3 3    8 5   2 4   6 7
8 3    8 1   6 5   5 9
0 4    5 3   7 0   4 6
6 8    0 2   5 7   7 6
1 9    2 7   2 0   9 7
9 3    3 8   1 2   0 9
4 0    6 4   2 8   6 1
9 6    8 5   5 6   6 8
7 4    6 1   0 4   0 3
3 0    6 8   7 1   8 3
5 4    8 8   6 7   3 3
9 4    1 9   7 7   8 8
9 9    9 9   5 0   1 4
2 7    1 8   2 4   1 3
3 4    3 3   7 4   5 6
5 6    1 4   1 0   6 1
                   8 7
                   1
                   5
```

Figura 83: Arquivo de custos utilizado no teste 23

Fonte: Autoria Própria (2014)

```
200
5 5
9 3
1 1
3 1
5 8
9 0
4 3
7 4
6 3
5 9
3 2
1 6
8 6
6 5
2 2
2 2
8 7
4 0
8 1
9 5
5 5
8 6
2 2
4 1
3 5
1 9
2 2
0 8
4 1
4 2
4 6
4 7
3 3
2 7
0 9
7 4
6 5
0 0
9 5
8 4
7 4
3 5
1 2
3 5
8 6
4 6
2 4
2 0
0 3
5 4
5 9
7 1
4 0
9 8
1 2
8 7
0 4
4 1
1 2
1 7
2 3
8 2
3 6
5 8
1 6
2 8
8 5
6 0
8 9
0 1
0 4
7 7
2 8
5 3
0 0
1 6
0 7
1 9
9 3
6 7
0 6
3 4
7 3
6 9
7 8
6 8
9 5
3 4
1 2
9 4
3 9
3 4
8 2
9 5
3 6
8 3
1 1
3 0
4 8
2 9
4 1
0 6
6 3
```

```
3 9
7 6
3 3
2 7
9 7
2 1
7 6
0 7
6 1
8 9
3 7
9 6
3 3
8 9
1 1
0 1
4 9
2 7
9 0
2 4
2 9
2 2
8 7
0 6
2 8
3 5
4 4
6 9
0 8
7 6
5 8
5 0
5 8
8 1
0 4
7 1
2 8
4 5
3 1
2 9
4 5
1 6
5 5
0 7
2 8
5 9
6 4
8 6
1 3
2 1
7 2
0 0
6 2
5 9
5 7
8 5
7 0
4 2
8 0
7 4
1 2
9 0
0 9
9 5
7 4
0 5
5 2
3 2
3 8
5 8
7 0
6 7
5 4
3 9
6 5
2 9
8 9
9 5
6 9
0 7
4 8
7 7
1 1
5 8
0 9
0 8
6 8
7 2
9 0
6 9
3 5
8 7
7 9
4 1
6 9
0 0
8 4
6
8 2 2 1 5 3
```

Figura 84: Arquivo de custos utilizado no teste 24

Fonte: Autoria Própria (2014)

150

7	4	3	4
6	6	4	0
4	3	1	0
8	8	6	7
3	4	9	3
9	6	1	0
6	3	0	2
7	5	9	2
3	8	6	1
4	5	8	1
0	7	8	6
2	7	8	6
7	9	3	4
5	6	1	9
5	4	3	4
4	0	1	5
9	8	5	7
0	4	5	5
8	6	1	1
2	6	0	0
8	5	3	1
7	9	3	2
1	2	3	5
9	2	2	7
2	9	3	1
5	8	4	9
7	8	5	3
9	9	8	2
3	5	9	0
2	0	2	8
8	4	3	3
9	7	5	9
2	2	6	2
1	5	1	6
4	5	3	5
6	9	8	9
1	4	1	8
8	1	0	4
1	1	1	5
1	3	5	9
3	6	5	8
9	2	0	7
7	5	9	2
4	5	9	6
2	3	4	4
8	3	4	7
6	0	5	2
4	9	0	8
6	9	6	5
5	4	0	0
7	5	5	0
2	5	9	3
3	7	1	0
0	0	1	8
4	4	2	1
0	5	7	2
7	1	1	2
0	4	3	8
3	3	3	6
6	0	4	1
8	9	8	6
8	9	8	8
9	0	5	7
7	9	9	3
1	2	5	8
2	3	9	4
7	6	3	8
3	8	7	1
6	8	6	6
6	5	0	6
5	4	1	9
8	4	9	6
6	6	2	6
6	4	0	8
6	5	4	7
		2	
		2	4

Figura 85: Arquivo de custos utilizado no teste 25
Fonte: Autoria Própria (2014)

```
500      6 2   2 3   9 3   1 7
  1 8    8 5   3 9   8 9   6 9
  3 4    2 1   2 5   8 6   9 8
  6 6    8 4   7 5   3 5   6 6
  4 4    0 7   7 8   5 0   5 5
  9 6    0 2   9 7   9 9   4 8
  7 7    8 0   0 4   3 5   6 9
  6 9    4 9   6 0   9 2   6 1
  9 9    7 6   4 4   7 0   2 4
  3 6    1 1   1 9   7 5   6 7
  3 2    8 0   9 3   6 7   8 0
  3 6    4 7   5 5   1 2   6 5
  8 6    3 2   6 3   9 5   7 0
  5 1    1 6   8 3   0 5   5 6
  7 2    1 4   2 9   0 7   9 2
  1 9    4 8   2 4   1 6   9 5
  3 3    1 9   3 3   3 9   1 0
  1 5    7 1   1 1   5 3   7 3
  1 6    4 9   7 2   8 2   8 4
  2 6    9 8   2 2   8 6   0 5
  0 0    9 2   6 6   1 7   8 3
  6 6    7 4   5 1   5 9   3 4
  9 3    1 4   0 7   4 9   7 2
  8 7    2 0   3 7   2 7   3 9
  1 7    2 5   6 0   8 8   7 2
  8 9    6 1   4 7   8 9   6 8
  9 7    6 0   1 2   1 7   6 3
  4 7    0 8   6 2   6 4   9 2
  2 4    0 5   9 0   4 6   4 0
  1 4    8 8   2 6   1 1   8 9
  7 8    8 9   2 8   5 1   6 1
  2 4    4 3   9 7   1 5   5 2
  7 5    7 7   5 1   5 6   4 7
  7 7    7 2   3 3   0 6   8 8
  8 0    6 8   0 9   5 6   7 1
  4 7    7 0   9 8   9 7   5 2
  2 5    7 3   0 6   4 0   9 3
  9 8    5 0   7 2   5 3   5 6
  5 0    2 0   9 6   0 4   0 9
  7 6    9 8   1 8   5 8   9 5
  2 5    2 4   8 9   0 6   9 9
  1 5    0 4   6 7   3 5   4 7
  3 0    3 6   3 4   0 4   1 9
  0 9    1 5   1 6   7 8   0 0
  0 4    9 3   6 9   1 0   2 1
  8 1    3 8   7 6   0 6   2 5
  9 7    8 7   9 5   5 4   1 7
  3 9    8 9   7 7   7 7   8 0
  3 4    5 3   8 4   9 8   5 6
  5 9    1 8   2 8   5 4   3 6
  8 8    6 2   4 7   4 1   5 4
  7 5    6 3   0 0   8 4   5 4
  5 5    1 0   0 7   8 5   6 9
  2 7    3 3   2 5   9 1   1 1
  2 1    5 7   1 1   7 2   4 1
  4 4    5 8   1 2   6 2   3 9
  7 8    1 5   0 8   3 4   6 1
  4 6    0 3   8 7   5 4   9 9
  6 4    3 8   3 5   9 2   7 9
  4 9    8 5   5 8   9 1   3 1
  4 4    2 7   5 0   8 5   7 1
  4 2    0 3   0 1   4 0   7 9
  9 2    6 6   0 6   7 2   1 7
  6 7    9 5   6 4   2 5   0 7
  0 8    9 1   4 0   7 5   8 4
  0 5    1 1   9 8   5 8   5 4
  3 8    9 7   3 4   7 3   4 7
  8 9    1 2   1 4   3 8   3 0
  3 8    7 5   2 9   9 2   8 6
  6 4    0 0   1 9   1 9   0 5
  8 2    6 1   4 9   6 4   3 1
  6 7    1 8   6 4   6 4   3 0
  6 9    4 0   5 3   5 0   0 3
  0 8    8 1   2 5   3 9   2 1
  7 7    1 7   0 9   8 1   5 1
  3 7    1 3   2 4   6 6   7 2
  3 8    8 3   5 9   7 0   1 7
  2 0    2 5   7 4   8 8   8 2
  4 9    4 2   4 2   5 3   8 0
  1 3    4 7   3 3   3 1   0 3
  0 0    6 9   6 5   9 1   9 5
  5 8    6 8   1 4   9 9   3 3
  7 5    5 1   3 0   4 8   1 2
  4 0    5 7   7 4   5 4   3 4
  6 3    6 7   1 1   9 2   6 4
  6 1    7 2   0 2   4 4   9 9
  4 7    1 2   3 4   3 2   1 1
  5 1    5 4   1 8   1 5   1 1
  1 0    1 3   4 4   3 3   3 2
  7 0    7 0   9 3   9 5   3 9
  9 3    7 5   2 2   5 8   9 5
  2 4    3 5   7 9   5 5   7 8
  4 3    4 9   2 3   4 7   4 0
  8 4    2 8   1 4   5 7   3 4
  4 1    0 5   1 9   5 8   6 9
  5 0    4 0   3 0   5 6   1 4
  9 0    7 1   4 6   0 3   0 3
  5 7    9 0   5 1   8 2   9 4
  7 5    3 8   6 4   2 3   8 3
  9 7    7 5   0 7   7 7   2 7
                           5 4
                           5
                           8 2 2 3 2
```

Figura 86: Arquivo de custos utilizado no teste 26
Fonte: Autoria Própria (2014)

600		4	6	3	3	6	5	7	9	1	6						
2	4	4	7	9	9	7	1	4	3	9	9						
1	7	8	1	8	1	3	9	8	5	0	1						
1	4	1	5	2	5	5	2	6	3	2	9						
5	9	7	1	9	7	3	1	7	8	3	2						
6	2	0	5	5	3	9	7	4	8	1	6						
1	0	8	0	2	6	7	7	1	5	7	3						
9	7	8	1	3	9	5	0	6	4	5	0						
3	1	5	1	9	4	2	5	5	2	7	8						
6	6	0	4	1	6	0	4	3	3	5	5						
6	4	0	9	7	9	8	8	9	1	6	4						
5	0	2	0	1	6	6	4	3	4	3	6						
1	2	5	4	9	3	3	4	3	7	4	0						
1	0	3	3	1	2	8	0	1	1	2	1						
4	9	9	3	1	7	9	4	2	0	3	3						
3	3	3	9	8	0	8	0	3	8	5	8						
6	4	2	6	9	7	8	9	6	2	4	3						
1	9	6	3	2	1	0	9	9	3	1	0						
9	2	4	1	4	6	6	9	9	4	9	9						
3	0	7	5	3	6	0	7	8	3	4	4						
8	1	3	6	1	2	7	4	8	6	2	7						
6	9	5	6	9	1	0	4	0	6	2	5						
3	8	3	3	8	0	2	8	5	3	1	5						
4	0	7	9	0	7	1	7	5	4	1	3						
0	0	0	7	5	0	2	5	6	9	1	5						
6	5	1	7	1	2	3	5	5	4	6	8						
6	9	7	9	6	6	9	8	1	1	2	6						
6	1	6	8	0	2	7	0	3	6	5	4						
0	2	4	9	8	7	2	1	0	7	8	3						
6	7	2	4	5	6	8	0	5	4	4	3						
0	1	4	5	7	0	6	9	4	2	7	1						
8	1	3	1	6	9	1	9	3	6	8	5						
6	9	1	7	1	9	7	5	4	2	3	4						
5	3	6	5	7	8	2	8	7	1	2	1						
6	5	8	0	7	2	4	9	9	1	0	6						
2	0	6	6	0	4	9	1	6	4	5	0						
9	0	9	3	1	1	9	8	7	1	7	7						
3	1	4	3	3	4	7	1	5	9	6	5						
8	2	1	2	9	0	8	8	4	4	0	2						
0	7	8	4	1	6	6	9	6	3	0	3						
8	2	1	9	3	5	0	0	6	6	0	5						
2	0	2	1	5	8	6	3	5	5	0	2						
5	7	1	5	2	6	8	2	4	6	4	2						
8	0	6	7	6	7	6	8	0	5	3	7						
5	4	8	6	6	2	8	7	6	6	1	9						
9	7	3	1	8	8	9	0	2	2	9	5						
0	8	6	6	5	9	4	1	3	4	3	4						
4	1	4	9	8	5	7	6	9	1	4	5						
4	5	5	2	9	6	3	7	8	2	0	0						
9	4	0	9	4	8	9	3	3	7	0	2						
4	7	5	5	4	2	8	1	0	3	3	5						
0	7	4	3	8	4	4	6	1	9	4	2						
3	8	8	5	0	7	8	9	6	8	6	2						
4	6	9	1	6	8	6	5	9	7	8	0						
9	4	6	8	3	1	8	4	2	1	7	1						
1	1	5	9	2	2	8	4	6	1	3	9						
1	0	7	5	3	8	7	4	5	4	7	8						
1	2	2	4	8	5	1	8	2	4	4	2						
5	5	8	7	4	7	6	8	1	5	9	7						
1	9	7	6	8	6	1	5	7	3	2	3						
4	6	0	5	9	7	9	1	8	9	0	4						
8	0	2	9	5	0	9	1	3	2	8	6						
1	1	5	3	5	5	3	2	5	6	8	8						
9	6	5	1	9	7	0	2	8	8	3	9						
4	8	5	1	5	0	6	7	7	3	6	5						
5	5	5	8	2	6	7	2	4	4	4	9						
0	8	2	6	8	8	2	1	6	2	7	2						
7	9	6	6	5	1	0	7	0	4	1	5						
4	7	7	2	6	3	3	3	8	4	5	2						
4	2	8	3	1	7	5	5	0	5	5	9						
1	4	2	8	4	7	7	0	7	4	2	2						
8	6	3	7	9	0	1	1	4	6	3	7						
3	9	3	9	1	8	6	3	5	2	5	3						
2	5	7	1	9	2	4	3	6	1	2	2						
1	6	9	6	1	5	4	2	6	7	8	7						
0	9	1	7	1	4	5	7	8	6	7	7						
5	5	5	9	9	1	7	8	0	3	8	7						
9	8	5	9	0	9	8	0	3	7	8	9						
5	5	2	2	5	9	8	3	8	1	9	7						
5	7	4	6	6	8	6	2	9	5	1	0						
3	3	2	8	4	2	2	3	3	4	6	0						
6	2	7	1	0	8	5	5	0	4	2	4						
8	7	5	2	2	1	5	5	7	9	7	3						
2	2	8	8	8	1	2	1	3	5	4	6						
1	4	4	9	1	4	4	8	1	8	4	9						
5	7	7	4	9	1	0	8	3	6	1	5						
0	5	0	2	8	8	3	1	8	3	3	6						
2	0	8	7	7	5	2	5	0	6	1	0						
3	6	1	9	6	1	8	4	7	0	8	0						
0	6	9	3	0	7	6	2	0	0	8	1						
8	5	9	4	0	1	6	6	1	5	9	8						
0	5	2	3	5	1	1	9	7	8	5	5						
5	7	4	0	3	3	8	5	4	0	7	5						
0	4	9	8	2	0	1	8	6	7	3	6						
4	5	1	2	0	0	3	6	7	7	3	9						
2	6	7	2	3	6	4	7	6	0	0	1						
7	2	9	1	1	7	3	1	0	0	8							
1	5	6	8	3	3	7	6	1	4	8	3	8	1	5	8	8	3
9	3	7	0	7	8	2	3	3	4								
8	6	4	8	6	1	0	3	9	7								
7	0	1	1	0	4	3	2	8	7								

Figura 87: Arquivo de custos utilizado no teste 27

Fonte: Autoria Própria (2014)

```
1000   8 7  7 3  6 5  0 3  8 1  9 0  6 7  8 8  3 4    5 8
 1 6   5 2  0 9  3 1  5 4  7 8  1 7  5 4  7 2  0 7    0 7
 3 4   7 6  2 3  7 7  8 7  6 5  4 7  4 5  4 9  4 4    8 4
 1 6   9 7  8 5  5 7  2 1  2 5  6 6  4 8  5 0  6 7    3 2
 0 3   2 6  0 7  1 8  1 1  0 9  7 6  1 1  1 1  2 7    4 0
 4 3   8 3  0 9  0 6  7 6  9 3  0 0  2 4  9 6  2 3    9 7
 9 0   5 5  8 9  6 3  8 3  3 0  4 0  7 6  6 7  6 9    5 8
 9 8   7 2  6 6  4 1  9 5  7 6  3 0  0 9  0 5  4 7    9 4
 0 5   9 2  5 7  1 2  3 8  4 4  9 8  8 0  0 6  5 9    6 4
 8 7   2 4  3 4  8 9  5 5  4 7  4 7  4 5  7 1  1 8    7 6
 5 3   0 5  8 0  6 7  8 5  4 4  0 9  0 4  3 1  3 3    1 9
 2 1   3 3  9 0  8 2  2 8  7 6  9 3  2 2  6 3  3 8   10
 3 9   8 0  3 1  0 0  0 5  3 8  6 0  6 4  5 9  7 2    3 5 5 3 9 9 7 7 8 6
 7 0   1 8  1 6  3 3  8 3  2 2  0 8  3 3  3 8  9 8
 5 0   3 7  6 1  7 9  7 2  3 1  3 5  4 2  4 5  7 2
 9 0   0 1  1 8  8 8  2 5  6 8  4 2  3 5  1 8  8 0
 5 6   9 0  8 6  1 7  8 7  6 2  8 1  9 0  6 4  6 5
 5 2   6 6  0 3  0 2  6 5  2 3  7 0  0 5  3 4  4 6
 6 4   6 9  7 7  1 6  8 5  9 4  8 2  3 0  3 3  7 5
 2 8   8 1  7 4  1 8  5 0  0 7  2 3  9 7  5 7  6 9
 9 0   1 8  0 9  2 9  7 3  4 8  6 2  2 0  7 1  1 2
 4 3   4 0  5 4  7 2  9 3  3 3  6 6  0 5  1 2  3 9
 2 3   4 5  1 7  4 6  2 4  3 1  1 2  0 9  5 5  2 6
 1 2   7 8  7 3  6 2  6 6  2 9  1 2  2 6  3 6  0 4
 4 8   5 8  8 6  0 3  0 5  3 4  9 8  9 0  7 8  4 8
 2 9   6 3  2 3  6 1  0 9  3 2  7 6  1 0  2 6  9 4
 4 5   1 3  6 4  2 0  0 5  6 2  3 0  4 1  7 8  2 8
 7 0   2 3  6 5  3 0  6 8  8 5  4 3  7 3  3 2  7 6
 1 5   3 6  8 7  7 8  3 9  8 9  7 8  4 2  1 6  6 1
 0 7   6 2  7 9  3 8  3 3  7 6  5 1  0 8  3 2  7 7
 3 3   9 0  7 7  1 7  8 6  3 9  1 0  2 1  8 2  8 2
 5 9   1 6  1 4  1 8  8 3  8 9  0 6  9 5  5 8  1 4
 0 4   6 0  4 7  9 6  5 1  2 9  7 4  8 6  8 1  0 3
 8 6   2 8  9 9  6 5  9 3  1 9  4 9  1 2  9 8  6 4
 6 8   5 1  5 9  0 0  1 4  2 4  5 4  2 8  3 6  1 2
 6 9   9 1  0 3  3 4  9 3  0 3  7 7  2 1  5 0  8 8
 6 7   5 6  4 2  8 8  8 8  7 3  1 4  0 5  6 4  8 3
 9 1   8 8  8 6  9 1  5 7  9 6  9 2  4 7  9 7  3 7
 8 6   1 6  5 8  1 5  9 3  5 2  1 3  7 6  9 0  4 3
 9 1   9 3  3 8  1 0  3 4  2 0  4 2  1 7  5 3  4 9
 3 5   7 1  8 2  8 0  4 3  8 6  0 8  3 9  8 0  5 1
 9 6   0 1  3 3  7 3  3 8  7 2  5 0  2 0  2 1  2 9
 2 4   2 1  8 9  2 2  4 1  5 8  4 7  6 4  6 8  9 9
 0 2   5 5  3 1  9 7  2 2  8 5  6 4  0 0  6 3  1 3
 7 8   7 9  9 0  5 8  6 3  3 2  7 9  3 5  4 0  3 7
 0 0   6 0  4 5  1 0  3 3  3 8  0 4  5 8  1 6  4 2
 8 2   6 2  6 9  6 3  7 2  2 8  6 5  0 0  4 6  1 8
 9 1   2 4  3 3  6 2  1 6  3 2  2 0  7 5  1 2  1 8
 7 6   7 4  9 5  6 5  2 7  1 4  1 9  8 8  0 3  4 9
 4 9   3 4  6 9  9 6  6 2  4 4  9 1  7 0  8 6  4 1
 2 1   1 2  6 5  0 1  3 4  1 2  1 7  2 2  7 7  3 4
 4 7   6 2  4 7  6 3  1 3  4 1  2 5  6 8  8 8  6 4
 7 4   9 7  6 8  1 1  2 4  3 0  4 0  0 6  2 9  4 8
 2 1   8 9  3 8  9 0  8 7  5 3  4 2  2 2  4 9  8 8
 5 8   5 4  4 2  0 9  9 4  2 4  5 7  8 8  1 6  1 7
 7 8   1 4  0 1  7 9  8 7  5 3  6 7  9 1  5 6  0 0
 3 5   0 9  5 1  5 9  5 6  8 7  6 9  5 5  3 4  0 3
 6 8   8 6  4 0  7 3  3 6  2 7  1 7  5 3  4 2  9 5
 5 8   2 5  8 8  3 0  9 7  1 5  1 9  7 7  2 0  6 0
 1 4   2 2  9 5  4 5  1 9  7 6  2 2  8 3  9 3  7 9
 3 0   9 9  5 3  9 7  1 7  2 6  6 9  7 1  9 9  5 2
 6 6   1 3  6 3  6 1  3 6  2 1  7 2  0 5  0 9  7 7
 2 3   5 0  4 0  1 0  6 0  4 1  8 6  1 2  9 5  3 6
 4 9   8 8  4 9  6 4  3 2  8 1  7 4  9 9  1 1  0 2
 8 8   3 6  3 5  2 7  9 1  8 3  8 7  6 5  9 1  5 4
 5 1   1 0  4 8  1 9  3 9  7 2  0 7  2 9  9 0  4 2
 1 8   9 9  0 4  8 1  5 3  0 0  4 4  6 7  8 8  8 6
 3 4   7 7  3 7  3 8  2 2  3 0  3 6  7 2  8 9  3 6
 7 5   3 6  1 8  9 0  1 9  5 8  3 2  8 6  6 0  0 9
 3 5   8 9  1 6  3 9  2 8  0 6  4 7  2 4  1 5  8 4
 9 9   4 2  2 4  7 1  7 7  7 1  1 8  9 6  7 9  4 9
 2 3   6 5  8 0  6 5  5 7  8 6  6 9  0 7  8 1  6 3
 4 3   7 2  9 9  2 2  3 9  3 9  4 2  7 3  1 7  3 3
 8 6   6 2  8 6  2 3  6 1  4 9  4 9  1 5  6 7  4 6
 6 8   8 2  6 2  0 2  3 1  5 1  4 1  3 1  5 6  8 4
 9 1   3 1  9 4  0 6  0 9  2 8  0 8  6 7  4 5  5 7
 6 8   6 7  0 0  0 6  9 8  9 8  5 1  0 3  8 3  8 4
 3 3   9 7  2 3  4 8  0 9  2 1  2 7  7 0  3 3  6 6
 3 7   7 1  4 8  7 9  5 7  5 7  7 1  1 8  9 8  5 5
 0 1   5 0  5 1  2 6  3 4  6 5  4 6  3 2  9 3  3 0
 4 3   0 0  0 1  4 7  7 8  3 9  3 7  6 7  9 8  1 0
 4 0   5 7  2 3  0 2  5 9  6 6  8 5  3 1  0 2  9 8
 5 7   1 4  1 9  9 6  1 5  8 0  6 8  5 4  0 9  6 6
 2 2   4 9  0 7  2 6  4 8  9 9  3 5  7 0  4 6  9 8
 4 7   0 3  9 7  5 0  8 6  0 7  8 1  5 2  4 5  7 9
 1 7   3 1  7 0  6 8  4 4  5 5  1 3  6 6  6 5  4 8
 2 5   9 7  5 7  7 9  5 7  0 8  3 0  1 8  8 0  7 2
 8 0   2 8  0 9  4 1  6 4  8 6  6 3  8 5  8 5  8 8
 0 1   9 0  2 5  1 7  1 2  8 6  6 9  8 0  2 5  2 9
 0 7   5 8  5 6  6 2  2 2  3 4  9 2  6 3  3 6  8 8
 8 2   3 2  6 8  2 4  7 7  0 1  3 3  9 1  3 4  4 9
 6 3   4 8  8 1  9 5  0 7  0 8  6 3  0 8  7 1  2 3
 6 8   3 3  7 2  4 7  0 2  9 9  2 2  0 2  2 0  3 5
 9 9   8 2  7 4  0 4  1 5  0 2  0 4  0 7  3 1  4 5
 8 9   7 8  9 8  1 0  8 9  9 2  7 3  9 8  3 7  6 2
 8 3   6 0  2 2  8 1  9 7  5 0  8 1  1 4  0 6  6 1
 5 9   5 9  8 3  7 1  6 0  0 8  8 5  0 7  1 1  3 1
 3 5   7 3  2 3  0 6  4 0  2 1  1 5  1 9  6 6  5 1
 2 8   0 5  1 9  0 4  0 3  3 0  9 9  0 3  1 0  1 4
```

Figura 88: Arquivo de custos utilizado no teste 28

Fonte: Autoria Própria (2014)

```
2000
1 5
5 2
8 5
2 5
5 7
9 4
7 3
4 3
3 7
6 0
5 3
2 6
7 5
0 4
2 0
2 3
3 9
9 0
1 3
0 2
5 9
1 5
7 1
5 2
6 7
2 6
3 5
1 4
0 4
9 0
8 7
6 2
5 0
7 2
4 0
9 6
5 9
3 8
1 0
0 1
7 6
8 3
1 5
7 3
5 7
6 0
5 2
9 0
8 8
3 5
4 2
2 2
5 9
0 4
8 9
0 7
8 0
0 0
6 8
3 8
6 3
6 5
2 9
0 3
2 3
0 1
5 8
0 2
5 0
6 3
3 8
7 1
7 1
7 2
5 7
6 2
3 4
4 5
7 4
3 6
7 3
2 3
8 0
4 5
5 0
5 2
5 0
6 9
6 1
9 0
0 4
3 6
6 9
2 4
9 6
8 1
0 1
1 2
0 1
9 0
1 7
2 9
0 5
1 6
5 5
0 7
6 0
4 2
7 3
8 0
6 1
2 6
7 7
0 9
9 2
6 7
1 1
5 2
4 0
0 0
2 3
5 9
8 3
7 1
6 3
8 2
7 3
4 0
6 4
7 3
0 3
8 9
1 2
3 4
1 1
8 4
2 1
8 1
8 7
0 2
9 8
5 2
3 0
2 8
9 8
4 3
9 8
1 8
1 7
0 4
9 8
1 6
2 7
2 8
8 8
9 4
0 2
7 6
8 0
6 8
3 6
6 8
5 3
9 5
1 5
6 1
8 7
5 1
1 1
9 7
3 0
8 4
1 8
0 9
3 5
5 1
5 4
9 4
1 6
4 8
3 8
0 2
4 0
8 8
9 5
5 3
2 1
7 7
3 8
9 9
3 5
7 5
7 7
7 6
9 3
3 8
7 9
0 7
1 0
7 2
4 1
1 7
4 1
0 2
5 0
4 1
4 4
2 9
4 1
9 0
9 8
7 2
0 0
9 6
1 2
9 0
9 7
1 8
5 3
9 1
2 0
0 6
2 7
4 9
9 3
6 9
0 8
8 5
5 1
1 0
5 2
2 4
1 3
5 4
3 5
9 9
6 8
5 2
7 2
8 2
2 3
0 9
8 9
4 2
6 4
8 1
8 5
4 7
6 4
4 2
6 4
7 9
1 4
9 8
5 2
7 1
8 2
5 0
5 5
7 2
4 3
1 2
9 1
5 1
7 4
9 0
4 5
4 2
4 2
7 0
6 4
3 6
7 2
7 2
1 0
0 7
6 4
6 4
8 0
2 9
7 7
6 3
5 4
8 1
4 4
1 7
6 2
8 2
4 9
6 0
7 6
1 0
6 6
5 2
7 6
2 1
2 1
3 5
7 9
6 3
6 6
6 5
3 4
2 7
2 7
8 8
0 9
4 1
3 3
5 3
5 1
2 4
3 8
6 3
8 3
4 9
0 5
4 7
9 3
6 0
1 4
1 6
4 3
0 6
3 8
6 9
9 9
1 9
1 4
9 8
6 5
4 7
8 9
5 0
4 4
6 5
2 0
6 4
3 8
3 0
9 6
8 5
4 9
8 2
2 5
1 2
3 1
1 4
0 0
6 3
2 3
2 5
6 3
8 5
5 9
2 6
8 3
8 1
0 7
0 2
3 4
9 6
1 4
3 6
0 6
9 1
7 3
7 5
5 8
6 8
8 0
6 9
2 9
2 3
7 4
6 9
5 5
5 8
2 0
2 0
4 3
5 5
8 0
6 8
3 7
4 6
0 3
4 8
8 1
5 0
7 3
9 4
2 6
7 3
9 7
8 8
2 5
4 6
0 0
1 3
9 4
6 2
5 0
0 1
3 4
1 1
0 4
1 7
0 5
7 3
1 7
4 6
2 0
6 0
6 3
6 5
2 7
3 0
1 7
0 3
9 8
0 3
6 3
2 1
8 2
2 1
8 8
9 4
4 3
4 0
2 1
6 1
5 4
8 0
0 4
6 5
9 8
3 7
1 5
3 9
9 7
3 8
3 4
9 5
3 4
8 9
8 2
5 1
5 5
1 7
2 3
5 6
9 0
4 1
5 7
8 2
6 6
2 3
3 9
2 9
2 7
2 6
1 1
9 0
8 2
0 7
5 1
2 5
0 6
4 7
6 1
1 9
0 9
1 1
8 2
4 6
0 0
8 4
2 4
2 1
0 7
9 1
7 3
8 5
3 6
5 2
1 6
1 0
7 0
6 4
5 5
4 5
7 9
7 2
3 4
9 5
0 5
8 6
7 9
3 5
4 0
9 2
7 4
4 2
3 0
6 3
4 9
0 4
1 0
4 9
3 4
7 9
1 0
8 1
6 8
0 8
8 4
3 6
1 9
9 3
3 1
5 2
0 7
2 0
3 3
9 3
9 2
1 4
7 4
4 9
5 0
9 1
1 4
5 1
5 7
8 8
8 3
3 3
2 7
5 8
8 0
0 1
7 6
8 9
5 3
4 0
4 9
2 2
2 6
3 4
9 4
7 5
4 0
7 3
4 5
2 6
3 0
9 5
6 2
5 2
0 9
6 5
9 0
0 0
8 9
8 8
4 0
5 4
4 4
2 3
3 6
1 3
8 7
7 4
9 6
1 8
8 2
0 8
3 7
7 8
0 5
8 3
5 3
4 3
7 1
0 8
6 5
2 7
7 0
3 2
0 9
7 1
2 8
9 0
3 9
1 3
8 2
5 6
9 8
2 1
9 1
9 9
3 4
4 7
6 0
1 2
2 1
8 9
7 4
7 7
6 6
1 2
8 8
4 3
4 5
0 2
0 9
4 0
2 0
9 7
3 7
5 3
2 1
8 3
7 3
8 6
2 7
5 2
8 3
1 5
4 9
0 4
7 7
9 5
8 5
7 4
8 7
1 1
6 9
7 6
5 7
7 6
5 4
9 0
4 7
9 4
7 9
2 0
8 9
2 2
0 3
0 1
9 3
2 9
0 8
0 0
9 5
9 9
1 1
4 6
6 6
0 7
5 1
5 6
3 2
1 0
3 6
7 3
8 0
0 5
3 2
0 5
0 7
8 1
6 4
8 3
3 1
4 7
2 5
4 9
8 9
5 3
0 2
1 1
9 6
5 2
0 4
6 4
1 9
8 9
8 9
0 6
0 4
0 6
4 9
9 0
0 1
9 3
1 9
4 8
3 6
8 0
2 4
6 1
6 5
6 9
9 6
2 6
4 4
6 7
5 1
2 2
9 9
0 2
6 5
0 5
7 2
5 0
1 7
8 7
8 2
6 6
3 5
0 2
3 7
7 2
7 2
7 8
6 9
1 7
8 3
9 0
6 4
3 7
6 5
8 8
1 4
8 5
6 5
7 9
0 2
3 8
7 8
8 8
8 5
1 6
6 1
4 2
4 2
0 7
0 1
0 2
8 6
6 7
7 2
5 2
9 3
8 7
6 7
3 9
2 2
5 8
0 8
9 0
3 4
2 9
3 5
0 0
0 5
7 5
3 8
0 8
0 1
3 6
1 7
0 5
0 0
2 9
9 7
8 1
2 6
0 1
0 3
4 1
1 5
6 0
5 3
9 0
3 2
8 2
8 8
9 0
6 6
5 6
0 5
4 0
0 6
0 0
3 9
9 5
0 1
5 5
1 1
8 2
9 5
7 2
5 4
0 9
4 1
6 4
3 1
2 0
8 7
3 2
1 5
2 4
6 0
8 5
7 4
3 5
8 6
1 3
7 5
3 7
8 8
9 6
5 5
6 2
2 9
7 2
9 7
9 1
5 5
5 4
0 2
3 0
6 0
2 5
3 1
3 9
6 2
7 5
1 7
2 1
0 5
2 3
7 9
8 8
0 9
1 3
1 5
3 6
6 3
8 7
4 4
5 4
8 5
3 2
9 5
1 4
4 2
9 0
1 0
6 0
4 0
9 5
2 4
5 8
5 1
6 5
6 7
5 8
6 0
2 2
6 0
3 1
4 5
6 4
4 6
3 2
3 1
5 9
8 7
1 5
5 0
2 8
0 4
9 3
6 1
6 1
7 0
2 5
0 3
4 4
7 1
3 3
8 6
1 6
0 6
4 4
3 4
1 2
9 1
2 2
7 7
1 8
2 1
4 9
2 6
5 5
3 5
1 8
0 9
5 5
4 7
0 3
8 4
7 6
1 8
0 1
0 4
1 5
3 5
4 8
8 5
7 3
3 4
9 7
5 8
5 1
9 2
3 5
6 5
0 3
0 3
3 5
2 5
2 6
9 2
1 7
6 1
5 7
7 8
3 9
3 1
4 8
2 3
5 8
7 5
1 9
7 8
3 7
3 9
0 8
1 4
1 2
3 0
9 4
3 1
6 2
3 4
1 1
7 4
1 2
4 1
6 9
1 3
9 9
9 0
7 0
0 8
2 0
8 9
3 1
6 3
2 7
8 5
8 6
3 8
7 2
6 9
2 9
0 7
0 7
1 1
2 2
0 6
0 3
6 6
5 3
3 5
0 0
7 2
7 9
3 1
7 7
8 5
3 8
8 0
2 4
4 2
2 8
7 7
4 8
5 7
2 5
0 7
2 4
6 4
5 7
3 4
9 5
3 5
3 8
7 3
3 4
4 7
1 5
3 0
1 7
7 6
3 2
0 0
6 8
9 1
8 1
8 5
7 6
9 1
5 2
7 4
5 0
8 1
7 9
4 7
4 1
3 1
4 5
6 5
5 3
2 6
8 5
8 5
8 7
8 1
1 1
7 4
1 1
3 6
9 5
4 8
4 3
0 3
9 6
7 2
9 6
9 6
4 3
2 0
5 0
1 8
7 8
6 4
6 9
1 4
1 0
3 7
6 8
0 8
0 6
1 1
1 5
6 1
2 2
7 1
1 8
7 8
4 4
2 0
4 4
9 1
5 5
7 0
9 3
0 1
2 5
2 1
3 5
3 5
5 6
7 4
8 0
3 0
0 7
9 1
9 0
0 7
6 7
6 2
0 8
5 7
0 3
3 5
7 5
6 1
5 4
7 8
8 9
1 7
5 6
9 9
7 1
8 9
4 3
1 2
0 8
2 6
8 7
7 1
6 4
5 4
4 9
3 1
2 2
7 3
8 7
7 0
4 8
4 5
7 0
5 2
3 1
7 0
2 4
3 8
2 1
3 2
9 2
8 4
6 0
9 2
3 9
7 9
5 0
5 4
6 2
7 1
5 1
0 1
6 0
3 6
9 3
3 4
4 4
5 9
2 6
3 1
5 8
5 8
3 1
5 2
7 4
3 3
1 5
0 7
7 0
5 7
2 7
2 8
3 0
0 2
3 1
9 7
2 7
8 7
0 2
2 6
1 9
2 4
9 2
2 1
1 0
2 7
3 0
0 8
9 3
0 3
5 9
8 2
6 1
7 8
6 9
5 8
2 8
2 6
1 9
5 2
4 3
1 0
6 7
3 6
0 4
0 6
9 3
1 5
8 2
6 9
4 4
5 8
4 6
7 9
0 3
1 2
1 9
1 2
6 0
2 6
0 5
2 4
5 9
7 0
2 1
3 3
8 5
9 1
2 8
7 4
0 2
9 6
4 3
7 2
9 2
2 6
4 7
7 8
1 9
4 9
5 2
7 6
6 5
0 9
6 9
5 1
1 6
4 4
4 2
3 0
8 6
0 5
1 4
0 2
8 3
6 0
9 2
0 9
5 0
4 9
6 4
7 9
6 5
0 2
3 0
9 5
3 1
2 9
5 2
9 7
8 3
6 9
6 3
6 6
7 3
1 1
5 4
3 5
0 4
5 4
9 9
6 6
4 2
3 1
4 0
8 9
9 8
3 8
7 2
3 2
3 4
8 2
4 3
6 5
5 3
4 1
7 4
1 1
0 2
6 6
6 3
6 2
6 0
8 3
0 6
4 7
1 6
2 4
7 6
1 5
3 6
0 5
9 9
4 2
7 4
3 4
5 7
4 6
6 4
4 8
3 0
0 5
1 0
7 9
4 0
4 3
4 9
9 5
0 7
4 3
2 3
9 5
2 5
7 9
5 9
7 0
5 3
9 8
0 3
6 1
1 0
0 9
8 7
4 3
4 7
6 4
1 7
0 9
2 1
2 5
3 9
7 9
8 5
9 3
2 4
8 6
0 5
0 8
0 0
4 2
7 2
0 6
4 9
0 4
7 9
9 3
4 2
3 5
0 3
2 4
1 2
1 3
2 1
8 7
5 3
2 0
0 3
2 2
6 5
4 5
3 0
0 7
8 7
4 4
8 7
3 0
4 4
2 6
2 7
8 8
1 8
8 9
3 3
7 6
7 9
8 8
9 1
2 6
0 5
4 8
8 2
0 2
6 2
9 0
6 1
9 2
0 2
8 3
6 5
3 3
3 4
8 1
5 7
2 5
1 2
6 9
5 6
3 0
4 5
9 3
8 6
6 1
1 6
5 6
0 8
2 9
5 9
2 1
3 3
2 2
9 2
1 7
5 8
5 8
0 1
7 2
2 0
8 1
8 1
0 3
8 0
2 6
9 8
3 6
8 1
9 7
6 6
7 1
4 9
3 8
6 0
3 9
1 0
6 6
1 0
6 4
8 2
4 0
7 8
8 8
3 3
4 6
6 2
0 1
7 7
5 0
6 9
8 2
4 1
0 6
3 4
1 6
4 9
4 8
6 9
7 9
9 8
1 7
2 1
5 0
6 9
2 1
8 1
7 9
5 8
8 0
4 4
8 3
8 0
1 0
0 8
9 6
5 8
1 9
1 8
3 6
0 7
8 6
9 2
8 1
8 8
9 8
9 0
7 6
0 8
4 9
4 7
5 9
4 2
8 8
7 0
5 8
7 2
4 9
7 8
9 6
5 3
6 9
0 1
9 1
5 8
0 7
4 5
7 4
7 9
3 2
8 6
8 0
5 4
5 3
0 0
8 1
5 6
9 4
1 4
5 9
6 5
1 7
4 1
6 6
3 4
5 6
8 8
6 9
7 8
9 0
8 0
7 5
2 9
3 3
5 6
5 2
5 0
1 1
1 9
6 5
9 4
8 0
2 1
3 3
9 7
7 6
6 3
9 0
0 1
1 6
6 4
8 7
6 8
8 1
5 2
8 9
5 5
9 0
0 5
0 5
5 0
0 9
8 4
9 8
6 6
9 9
6 3
4 4
3 7
0 8
6 0
1 6
3 2
7 4
6 8
8 1
6 8
8 5
0 4
8 1
7 0
2 6
2 4
2 4
9 6
1 5
7 8
5 2
5 5
4 5
5 5
9 6
0 3
8 6
3 4
9 9
5 9
9 4
0 6
7 5
1 8
6 2
1 4
3 0
8 9
6 1
0 2
2 4
9 0
5 5
4 7
6 7
4 4
2 0
2 4
1 4
3 6
4 4
8 9
8 5
4 8
1 4
5 3
6 5
8 7
7 2
9 4
7 2
7 9
4 6
3 8
7 8
9 3
8 1
6 8
6 3
5 7
8 3
0 0
3 9
3 3
9 4
4 9
4 6
0 8
4 1
3 9
4 6
5 1
8 1
5 6
7 5
6 1
1 0
3 8
0 6
9 4
5 6
4 5
8 0
5 8
0 1
3 4
6 3
7 5
3 0
9 2
4 9
4 5
6 9
2 5
2 3
4 1
6 8
5 8
8 0
8 1
4 0
5 6
2 7
0 1
1 8
9 5
4 7
1 0
4 4
3 4
3 9
2 1
9 6
3 8
3 8
2 0
3 2
9 8
9 5
1 2
7 3
6 0
0 5
5 0
4 3
2 5
6 4
2 6
0 5
6 0
3 0
8 6
4 5
6 2
1 7
8 2
4 9
8 3
9 3
9 2
2 7
1 5
8 5
8 9
6 4
1 5
2 8
5 8
2 0
2 5
7 3
7 8
2 8
5 1
9 1
1 3
6 9
8 3
5 0
7 8
4 1
7 9
1 9
4 7
5 7
6 1
7 7
3 5
0 6
7 3
0 5
5 2
5 9
7 4
3 3
9 6
0 9
8 4
1 3
6 3
3 7
6 7
2 1
8 0
2 8
5 2
1 6
9 8
4 4
3 5
7 2
0 1
6 4
6 9
0 5
6 1
6 7
8 3
9 3
8 4
7 6
8 7
5 4
8 0
8 2
1 2
4 3
9 0
1 8
7 8
9 0
1 8
4 9
3 2
8 7
0 3
0 4
0 1
2 5
1 9
4 9
9 8
3 4
4 4
6 1
0 0
3 2
6 1
3 3
9 3
6 7
4 4
7 8
1 3
7 7
4 6
6 1
5 5
0 0
3 6
4 3
6 8
9 9
1 4
5 5
2 7
7 7
9 2
1 6
3 3
5 0
7 2
8 8
0 4
0 7
7 1
2 0
3 9
5 9
4 3
4 7
7 5
6 6
3 4
9 8
7 6
2 0
6 3
7 2
8 1
2 8
4 7
8 4
9 4
5 6
2 8
8 2
0 5
2 0
6 3
3 4
6 6
3 7
0 6
2 6
1 1
8 8
2 5
6 6
8 8
2 9
5 3
2 9
3 2
3 1
9 2
6 9
5 1
9 2
6 2
8 2
0 7
5 3
6 7
7 8
7 0
5 3
1 7
5 6
3 2
8 1
0 2
4 3
1 8
8 5
4 2
7 7
1 5
5 3
0 6
2 5
0 2
5 2
7 4
0 0
6 3
6 2
1 6
6 2
3 8
9 7
6 1
1 8
4 4
4 3
9 2
1 8
3 1
4 5
6 9
3 7
9 8
3 8
3 3
3 6
4 7
6 0
8 6
0 3
6 8
3 2
3 4
7 9
1 2
3 8
4 2
9 7
6 2
6 5
5 5
1 3
5 7
4 7
2 6
6 2
9 6
8 8
2 3
3 0
4 4
4 5
8 1
8 2
5 5
4 4
1 5
7 3
1 7
8 6
5 7
9 9
8 0
7 4
9 7
3 1
5 2
6 2
9 1
4 4
3 7
1 1
1 6
2 6
0 9
4 7
2 6
7 8
7 5
2 2
6 1
4 8
6 7
8 5
4 9
7 8
6 5
2 2
9 1
6 6
2 8
10
1 3 1 5 4 0 2 2 8 4
```


Figura 89: Arquivo de custos utilizado no teste 29
Fonte: Autoria Própria (2014)