

FELLIPE FERNANDES CARDOSO MARCELLINO

ROTEIRIZAÇÃO DE ENTREGAS PARA UMA *STARTUP* DO SETOR DE
ALIMENTAÇÃO SAUDÁVEL

Trabalho de Formatura apresentado à
Escola Politécnica da Universidade de São
Paulo para obtenção do Diploma de
Engenheiro de Produção.

São Paulo

2017

FELLIPE FERNANDES CARDOSO MARCELLINO

ROTEIRIZAÇÃO DE ENTREGAS PARA UMA *STARTUP* DO SETOR DE
ALIMENTAÇÃO SAUDÁVEL

Trabalho de Formatura apresentado à
Escola Politécnica da Universidade de São
Paulo para obtenção do Diploma de
Engenheiro de Produção.

Orientadora: Prof^a Dr^a Débora Pretti
Ronconi

São Paulo

2017

FICHA CATALOGRÁFICA

Marcellino, Fellipe Fernandes Cardoso

ROTEIRIZAÇÃO DE ENTREGAS PARA UMA *STARTUP* DO
SETOR DE ALIMENTAÇÃO SAUDÁVEL / F. F. C. Marcellino -- São
Paulo, 2017.

130 p.

Trabalho de Formatura - Escola Politécnica da Universidade de
São Paulo. Departamento de Engenharia de Produção.

1.Pesquisa Operacional 2.Logística 3.Roteirização 4.Heurística
I.Universidade de São Paulo. Escola Politécnica. Departamento de
Engenharia de Produção II.t.

Aos meus pais, à minha irmã e ao
meu avô, que se estivesse aqui com
certeza estaria muito orgulhoso.

AGRADECIMENTOS

Primeiramente à minha família, pelo suporte constante em todos os momentos, pela motivação nos períodos difíceis, pelos conselhos nos momentos de dúvida e, por fim, pelos valores e educação proporcionados.

À professora Débora Pretti Ronconi, por ter me introduzido à fascinante área de Pesquisa Operacional em seu curso de graduação e pela paciência, incentivo e dedicação demonstrados ao longo do desenvolvimento deste trabalho, sempre acreditando no potencial de seus alunos.

À Escola Politécnica e seus professores, pela oportunidade de poder cursar Engenharia de Produção em uma das melhores escolas de engenharia do Brasil e da América Latina e pela oportunidade oferecida de formação internacional através de seu programa de Duplo Diploma.

A todos os amigos e colegas feitos nesse período de graduação, tanto no Brasil quanto na França. Em especial, aos amigos do Centro Acadêmico de Engenharia de Produção e do *Bureau des Arts*, por todos os momentos, aprendizados, dificuldades e conquistas juntos, que foram de fundamental importância no processo de formação.

Por fim, a todos os mestres com os quais tive a oportunidade de aprender, tanto no Colégio Dante Alighieri quanto no Colégio Bandeirantes, pela formação de excelente qualidade e por me guiarem no caminho das Ciências Exatas.

RESUMO

Este trabalho apresenta uma proposta de melhoria do processo de roteirização de veículos da *Liv Up*, uma *startup* de alimentos saudáveis congelados. A *startup* encontra-se no início da fase de crescimento acelerado e, desta forma, precisa ganhar escala em todas as operações, sendo a operação logística o foco deste trabalho. Atualmente, o processo de roteirização é manual, o que dificulta sua escalabilidade e também impacta no custo, tempo do processo, nível de serviço e gestão da operação. Através de técnicas de Pesquisa Operacional, o problema foi modelado em Programação Linear Inteira Mista e testes foram realizados para um conjunto de clientes, de forma a se obter a solução exata. Devido à alta complexidade computacional dos problemas de roteirização, a obtenção da solução exata se mostrou viável para até 15 clientes roteirizados, o que representa menos de 1/3 do número atual. Assim, recorreu-se ao uso de métodos heurísticos de resolução. Utilizou-se o algoritmo de inserção I1 de Solomon (1987) como ponto de partida, adaptando-o para a realidade do problema. Em seguida, a eficácia do algoritmo foi verificada em escala reduzida. Por fim, o algoritmo foi testado num conjunto de instâncias reais e análises de sensibilidade das restrições do problema foram efetuadas. A ferramenta de roteirização desenvolvida neste trabalho proporcionou impacto positivo no tempo de processamento, no custo, no nível de serviço e na gestão da operação, havendo ainda espaço para melhorias incrementais em trabalhos futuros.

Palavras Chave: Pesquisa Operacional, Logística, Roteirização, Heurística

ABSTRACT

This paper proposes an improvement to the vehicle routing process of *Liv Up*, a startup in the healthy frozen food sector. The startup is at the beginning of an accelerated growth phase and, therefore, needs to scale all its operations, with logistics operations being the focus of this work. Currently, the routing process is manual, which hinders its scalability and also impacts the cost, the processing time, the service level and the operation management. Through Operations Research techniques, the problem was modeled in Mixed Integer Linear Programming and tests were performed for a set of clients, in order to obtain the exact solution. Due to the high computational complexity of routing problems, obtaining the exact solution proved feasible for up to 15 clients, which represents less than 1/3 of the current number of clients being routed. Thus, a heuristic approach to the problem proved necessary. Solomon's I1 insertion algorithm (1987) was used as the starting point and adapted to the reality of the problem. Then, the efficacy of the algorithm was verified on a reduced scale by a comparison with the exact solution. Finally, the algorithm was tested on a set of real instances and sensitivity analyzes of the constraints of the problem were performed. The routing tool developed in this work has had a positive impact on processing time, cost, service level and operation management, and there is still room for incremental improvements in future work.

Keywords: Operations Research, Logistics, Routing, Heuristic

LISTA DE ILUSTRAÇÕES

Figura 1 - Logotipo da startup	23
Figura 2 - Exemplos de produtos vendidos pela Liv Up	26
Figura 3 - Página inicial do website da startup.....	27
Figura 4 - Etapas de transformação do produto.....	28
Figura 5 - Exemplo de mapa de entregas	30
Figura 6 - Classificação de problemas de fluxo em redes	34
Figura 7 - Mapa ilustrativo das 4 zonas de entrega da empresa terceirizada	57
Figura 8 - Exemplo de pontos interno e externo a um polígono	60
Figura 9 - Descrição do algoritmo de verificação da zona	61
Figura 10 - Exemplo de rotas geradas pelo método exato para 14 clientes.....	69
Figura 11 - Descrição do algoritmo adaptado da heurística de inserção I1 de Solomon	73
Figura 12 - Roteiro efetivamente realizado para o problema R6	88
Figura 13 - Roteiro gerado pelo algoritmo para o problema R6	89
Figura 14 - Painel de controle da ferramenta de roteirização.....	111
Figura 15 – Aba de coordenadas dos vértices das fronteiras das zonas de entrega.....	111
Figura 16 - Aba de instâncias relacionadas aos clientes.....	112

LISTA DE GRÁFICOS

Gráfico 1 – Gasto médio anual por pessoa com alimentos saudáveis em 2016	25
Gráfico 2 - Comparação de algoritmos heurísticos quanto à eficiência (ordenada) e complexidade (abscissa). Quanto menor o valor da ordenada, maior é a eficiência do algoritmo.....	49
Gráfico 3 - Regressão linear da distância real vs distância linear	64
Gráfico 4 - Evolução do tempo de processamento em função do número de clientes	70
Gráfico 5 - Comparação do desvio em relação ao ótimo para os conjuntos A, B e C de problemas	84
Gráfico 7 - Custo total em função de α para o problema R1	90
Gráfico 8 - Custo incremental da adição de restrição de janela de tempo para 4 cenários	92

LISTA DE TABELAS

Tabela 1 - Variações do Problema de Roteirização de Veículos.....	36
Tabela 2 - Principais aplicações de heurísticas para o PRVJT.....	47
Tabela 3 - Custos de entrega por zona de chegada.....	57
Tabela 4 - Taxas de retorno por zona de saída	58
Tabela 5 - Exemplo de matriz-custo 4x4.....	59
Tabela 6 - Exemplo análogo à Tabela 5	62
Tabela 7 - Instâncias do modelo	66
Tabela 8 - Matriz-custo (em R\$)	67
Tabela 9 - Matriz-tempo (em horas).....	68
Tabela 10 - Soluções geradas pelo software CPLEX (D = depósito central).....	68
Tabela 11 - Características dos conjuntos de problemas A, B e C.....	77
Tabela 12 - Análise da eficácia da heurística para o conjunto A e $\alpha = 0,5$	77
Tabela 13 - Impacto do parâmetro α na eficácia do algoritmo.....	79
Tabela 14 - Custo Total Obtido em função de α para o problema A14.....	80
Tabela 15 - Impacto do retorno do veículo na eficácia do algoritmo.....	81
Tabela 16 - Impacto do aumento de restrições de janela de tempo na eficácia do algoritmo ..	82
Tabela 17 - Comparação do custo real e calculado pelo algoritmo, para $0 < \alpha < 1$	86
Tabela 18 - Comparação do custo real e calculado pelo algoritmo, para $0 < \alpha_1 < 10$ e $\alpha_2 = 0,87$	
Tabela 19 - Comparativo entre resultados em função de α	89
Tabela 20 - Custo calculado pelo algoritmo para diferentes cenários de janela de tempo	91
Tabela 21 - Economia incremental na eliminação das taxas de retorno no conjunto R de problemas	93
Tabela 22 - Instâncias do conjunto A de problemas.....	123

SUMÁRIO

1	INTRODUÇÃO	19
2	DESCRIÇÃO DA EMPRESA E DEFINIÇÃO DO PROBLEMA.....	23
2.1	Descrição da empresa	23
2.2	Definição do Problema	29
2.3	Escopo e Objetivo do Estudo	31
3	REVISÃO DA LITERATURA.....	33
3.1	Introdução à roteirização de veículos	33
3.2	Problema Clássico de Roteirização de Veículos (PRV).....	35
3.3	Problema de Roteirização de Veículos com Janelas de Tempo (PRVJT).....	40
3.4	Métodos Heurísticos	41
4	MODELAGEM EM PROGRAMAÇÃO LINEAR INTEIRA MISTA.....	51
4.1	Formulação do Problema.....	51
4.2	Coleta e Processamento de Dados	55
4.3	Resultados do Modelo de Programação Linear Inteira Mista	65
4.4	Análise dos Resultados.....	69
5	MÉTODO HEURÍSTICO DE RESOLUÇÃO	71
5.1	Adaptação do algoritmo de inserção I1 de Solomon	71
5.2	Eficácia do algoritmo desenvolvido	75
6	ANÁLISE DOS RESULTADOS OBTIDOS PELA HEURÍSTICA ADAPTADA	85

6.1	Comparação com solução atual.....	85
6.2	Análises de sensibilidade	90
6.3	Impacto dos resultados para a <i>startup</i>	94
7	CONCLUSÃO.....	99
7.1	Considerações Finais.....	99
7.2	Trabalhos Futuros	100
	REFERÊNCIAS BIBLIOGRÁFICAS	101
	ANEXO A – CÓDIGO VBA DO ALGORITMO DE CÁLCULO DA ZONA DE UMA COORDENADA.....	107
	ANEXO B – <i>GOOGLE MAPS DISTANCE MATRIX API</i>.....	109
	ANEXO C – FERRAMENTA DE ROTEIRIZAÇÃO	111
	ANEXO D – CÓDIGO INSERIDO NO SOFTWARE CPLEX	113
	ANEXO E – ALGORITMO ADAPTADO DA HEURÍSTICA I1 DE SOLOMON	115
	ANEXO F – INSTÂNCIAS DO CONJUNTO A DE PROBLEMAS.....	123

1 INTRODUÇÃO

O empreendedorismo no Brasil vem assumindo papel cada vez mais relevante na economia nos últimos anos. Segundo pesquisa de 2015 realizada pela *Global Entrepreneurship Monitor* e patrocinada pelo Sebrae, a taxa total de empreendedorismo no Brasil foi de 39,3% naquele ano, a maior desde 2002 (Bedê, 2016).

Por outro lado, segundo estudo de 2014 realizado pelo IBGE, cerca de 50% das empresas abertas encerram seus negócios nos primeiros 3 anos (Guimarães *et al*, 2014), sendo a má gestão dos recursos uma das causas mais comuns, de acordo com relatório publicado em 2016 pelo Sebrae (Greco *et al*, 2016). Assim, é de suma importância uma gestão eficiente de todas as áreas da empresa, desde o início da operação.

Neste contexto, o presente trabalho foi realizado na *Liv Up*, uma *startup* do setor de alimentação congelada saudável fundada em 2015. O foco do trabalho se deu na operação logística, sendo relevante para a *startup* devido à sua estratégia de entrega baseada no *delivery* de seus produtos.

Por se tratar de uma *startup*, a *Liv Up* ainda não possui as características de uma empresa madura. Desta forma, deve passar por estágios de desenvolvimento e amadurecimento.

Inicialmente, uma *startup* começa com uma ideia e um modelo de negócio, que deve ser validado e testado no mercado. Após seguidas iterações e aperfeiçoamentos da ideia, o produto/serviço é lançado.

Nesta etapa, contratações iniciais são feitas, um processo de atração de clientes é colocado em prática e a operação começa a funcionar, embora em pequena escala.

Caso o empreendimento tenha sucesso, ele segue para a etapa de crescimento acelerado. É no início desta fase que se encontra a *Liv Up*.

A etapa de crescimento acelerado é uma fase de transição entre uma *startup* e uma empresa madura. Em linhas gerais, esse é o momento em que a operação ganha escala. No entanto, este processo exige um aumento na produtividade de todas as operações para que seja viável.

Desta forma, a *Liv Up* deve adaptar todas as suas operações para um cenário de grande escala. A motivação deste trabalho é, especificamente, a escalabilidade da operação logística, que representa mais de 10% dos custos da *Liv Up*. Esta operação é de extrema importância para a

startup, já que afeta diretamente o nível de serviço e a satisfação do cliente, sendo uma oportunidade de diferenciação em relação à concorrência.

Com essa finalidade, serão empregadas técnicas de Pesquisa Operacional no desenvolvimento de uma solução de grande escala para o processo de roteirização das entregas, de forma a reduzir o custo e tempo do processo, repercutindo também no nível de serviço e gestão da operação.

O trabalho é estruturado em 8 capítulos e 5 anexos ao final.

O Capítulo 1 (atual) fornece uma visão introdutória do contexto e motivação do trabalho.

O Capítulo 2 apresenta uma descrição da *startup* com breve histórico, principais linhas de produto, cadeia de valor, entre outras informações. Esse capítulo também ilustra como é feito o processo de logística atualmente, os problemas encontrados e é concluído com uma descrição do escopo e objetivo do trabalho.

O Capítulo 3 reúne os principais conceitos, autores e trabalhos relacionados ao Problema de Roteirização de Veículos, após extensa revisão da literatura. Assim, ele apresenta as principais variações do problema, assim como as formulações do Problema de Roteirização de Veículos tradicional (PRV) e com janelas de tempo (PRVJT). Em seguida, uma breve introdução à complexidade computacional é dada e os principais métodos heurísticos relacionados ao PRVJT encontrados na literatura são abordados.

No Capítulo 4, o problema é formulado como um modelo de Programação Linear Inteira Mista e busca-se verificar a viabilidade da obtenção da solução ótima. Em seguida, a coleta e processamento dos dados de entrada, por ter uma dimensão relevante no problema, é explicada em detalhes. Por fim, testes computacionais são realizados e os resultados são analisados.

O Capítulo 5 contém o método heurístico desenvolvido neste trabalho. Ele inicia-se com a descrição das adaptações feitas à heurística de inserção I1 de Solomon (1987). Em seguida, o algoritmo é testado em 3 conjuntos de instâncias do problema e os resultados são comparados com a solução ótima, de forma a se ter uma visão inicial da eficácia do algoritmo em menor escala para diferentes cenários.

O Capítulo 6 apresenta e analisa os resultados do trabalho. O algoritmo é testado em um conjunto de instâncias reais do problema e, como não se tem o conhecimento da solução ótima, compara-se com a solução atual obtida pelo processo da *Liv Up*. Em seguida, duas análises de

sensibilidade são realizadas. Por fim, o impacto dos resultados para a *startup* é estimado tanto de forma quantitativa quanto qualitativa.

O Capítulo 7 conclui o trabalho com considerações finais e propostas de trabalhos futuros no tema. Ao final do trabalho, a revisão bibliográfica é apresentada e 5 anexos são inseridos, contendo códigos de programação, instâncias de problemas, entre outros elementos de consulta.

2 DESCRIÇÃO DA EMPRESA E DEFINIÇÃO DO PROBLEMA

Este capítulo fornece uma breve descrição da *startup* na qual o trabalho se baseia, abordando elementos como o mercado, a linha de produtos e a cadeia de valor, entre outros. Em seguida, a operação logística atual é detalhada, assim como os problemas encontrados. Por fim, o objetivo e escopo deste trabalho são definidos.

2.1 Descrição da empresa

Fundada em Dezembro de 2015 por engenheiros de produção da Escola Politécnica da USP, a *Liv Up* (logotipo na Figura 1) é uma startup que atua no mercado de alimentação orgânica e saudável, com escritório situado no bairro Vila Madalena, na capital paulista.

Figura 1 - Logotipo da startup



Fonte: Disponível em: <<http://www.livup.com.br>>. Acesso em: 10 de jul. 2017.

2.1.1 História

A ideia do negócio surgiu no fim de 2014, quando os fundadores Victor Santos e Henrique Castellani ainda trabalhavam no mercado financeiro e consultoria. Devido à acelerada rotina, sentiam a necessidade de uma alimentação prática e ao mesmo tempo saudável, o que não encontravam facilmente em São Paulo. Assim, ambos decidiram se dedicar exclusivamente a essa oportunidade de negócio.

Inicialmente, o modelo de negócio se baseava na produção e entrega a domicílio de refeições prontas preparadas no mesmo dia. Porém, esse modelo só se tornaria viável em escala nacional com a produção em diversas cozinhas. Como isso demandaria um alto investimento inicial, o risco do negócio seria muito alto. Para contornar este problema e permitir uma produção centralizada, optou-se pela venda de comidas congeladas.

Uma refeição colocada num *freezer* doméstico pode levar de seis a dez horas para ser totalmente congelada, o que é um processo longo e que compromete a qualidade do produto. Desta forma, a *Liv Up* utiliza uma técnica italiana de ultracongelamento que leva apenas 30 minutos. A utilização deste método garante que a água, ao congelar, tome a forma de microcristais, o que ajuda a manter as propriedades do alimento, como sabor, textura e consistência. Além disso, a validade do produto chega a ser de até 6 meses.

No início das operações, os próprios fundadores preparavam os pratos. Após a ajuda de investidores-anjos, dos próprios fundadores e de investidores profissionais num total de R\$1,6 milhões, a *Liv Up* cresceu num ritmo acelerado - em média, 40% ao mês. No final de 2016, a *startup* já possuía um faturamento anual de R\$2,8 milhões, 38 funcionários, 1 cozinha e uma base de 2000 clientes.

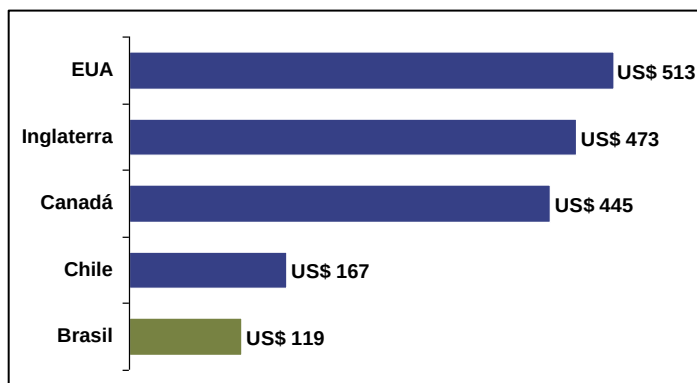
No início de 2017, a *Liv Up* começou a expandir geograficamente o negócio, entrando no mercado do Rio de Janeiro. À procura de novos investimentos, os fundadores pretendem adquirir uma nova cozinha e acreditam quintuplicar o faturamento anual até o final do ano.

2.1.2 Mercado¹

Em 2016, o mercado brasileiro de alimentos e bebidas saudáveis representou R\$93,6 bilhões em faturamento, colocando o país na quinta posição no ranking mundial, como pode-se observar pelo gasto médio anual por pessoa, no Gráfico 1. Segundo a agência de pesquisas *Euromonitor Internacional*, nos últimos cinco anos as vendas deste setor no Brasil cresceram em média 12,3% ao ano, comparado com 8% ao ano no resto do mundo. Desta forma, existe uma clara tendência no Brasil e no mundo de busca por uma alimentação saudável, orgânica e livre de aditivos químicos.

¹ Disponível em: < <http://www.fiorde.com.br/wordpress/blog/venda-de-produtos-organicos-cresce-em-2016>>. Acesso em: 10 de set. 2017.

Gráfico 1 – Gasto médio anual por pessoa com alimentos saudáveis em 2016



Fonte: Elaborado pelo autor

O relatório Tendências Mundiais de Alimentação e Bebidas 2017, elaborado pela agência de pesquisas Mintel, mostra que quatro em cada cinco brasileiros estão dispostos a gastar mais se o alimento tiver maior valor nutricional e 79% dos entrevistados já substituem produtos convencionais por outros mais saudáveis. Os dados comprovam que os hábitos de alimentação no Brasil estão mudando tendo em vista um estilo de vida mais saudável. A previsão é que este mercado cresça 4,4% ao ano até 2021, segundo a *Euromonitor*.

2.1.3 Produtos

A *Liv Up* possui uma linha de produtos variada, contando com 6 categorias de alimentos, vendidos em porções individuais congeladas e embaladas. As receitas foram preparadas por nutricionistas da Universidade de São Paulo e implementadas por cozinheiros com experiência internacional. Segue abaixo exemplos de produtos para cada linha de produto da *startup* e ilustrações na Figura 2.

- Vegetais: abobrinha recheada com ricota, creme de abóbora, purê de cenoura com gengibre, entre outros.
- Proteínas: hambúrguer de salmão, filé mignon grelhado, kafta de cordeiro, entre outros.
- Carboidratos: arroz integral, risoto de abobrinha, batata doce, entre outros.
- Snacks & Doces: barra de açaí, bolo de banana com nozes, tapioca sticks, entre outros.

- Sopas: caldo verde, sopa de frango com legumes e sopa de legumes.
- Massas e Molhos: molho cassé, panqueca de ricota, nhoque integral de mandioquinha, entre outros.

Figura 2 - Exemplos de produtos vendidos pela Liv Up



Fonte: Adaptado de: <<http://www.livup.com.br/cardapio-comida-congelada>>. Acesso em: 10 de jul. 2017.

Além disso, *kits* temáticos - *low carb*, vegetariano etc – e pratos contendo porções das 6 categorias são sugeridos e podem ser comprados em conjunto.

2.1.4 Concorrentes

Apesar de ser um mercado relativamente novo, já existem alguns concorrentes da *Liv Up* com modelo de negócio e linha de produtos semelhante, voltados para a alimentação congelada saudável.

Dentre eles, destacam-se a *Pronto Light*, *Light Chef*, *Lucco Fit* e *Congelados da Sônia*.

2.1.5 Fornecedores

A busca de fornecedores foi baseada em ingredientes orgânicos, sem agrotóxicos e sem conservantes. Desta forma, formou-se parcerias com agricultores familiares.

Hoje, a *Liv Up* tem parceria com agricultores familiares de Sorocaba, Porto Feliz e Tietê, além de cooperativas de grãos no Paraná, Rio Grande do Sul e Sergipe.

2.1.6 Canal de Vendas

Atualmente, o único canal de vendas utilizado é o *website* da *startup*, cuja página inicial pode ser observada na Figura 3. Nele, é possível encontrar a descrição dos produtos, informação nutricional e modo de preparo. Além disso, no próprio *site* é possível selecionar os produtos e realizar a compra *online* através de diversos meios de pagamento.

Figura 3 - Página inicial do website da startup



Fonte: Disponível em: <<http://www.livup.com.br>>. Acesso em: 10 de jul. 2017.

No futuro, pretende-se diversificar o canal de vendas, incluindo vendas indiretas através de redes varejistas como supermercados, lojas de produtos naturais etc.

2.1.7 Funções e cadeia de valor

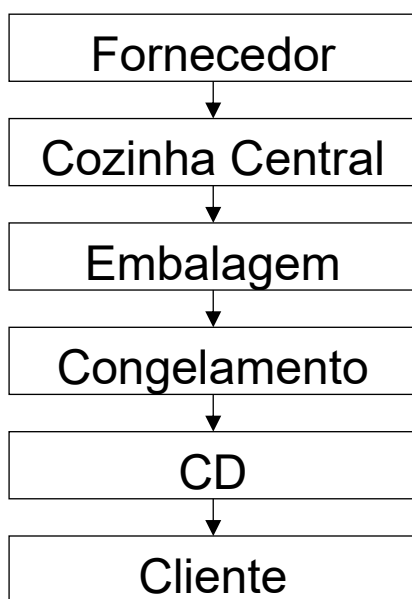
Por se tratar de uma *startup*, não existe um organograma bem definido da estrutura organizacional. Porém, é possível identificar algumas funções dentro do empreendimento:

- Produto: esta área é responsável pela seleção dos ingredientes, elaboração da receita das refeições e controle de qualidade, com auxílio de *chefs* e nutricionistas.
- Produção: área operacional localizada na cozinha central, onde os ingredientes são transformados em refeições e congelados na sequência.
- Compras: área responsável pela escolha e negociação de preços com fornecedores.
- Marketing: área responsável pela divulgação da marca, atração e retenção de clientes, através de mídias como rádio, redes sociais e jornais.

- CRM (Customer Relationship Management): área responsável pela interação e relacionamento com o cliente com o auxílio de ferramentas de gestão.
- Tecnologia: área responsável pela programação e suporte de sistemas de informação, como o *website* e o banco de dados.
- Logística e Expedição: área responsável pelo planejamento das entregas (através da definição de rotas e itens a serem entregues para cada cliente), pelo *picking*, conferência e gestão do estoque.
- Administrativo: área responsável pela contabilidade, finanças, recrutamento, entre outras funções.

Até chegar ao consumidor final, a matéria-prima passa por diversas etapas de agregação de valor, como ilustrado na Figura 4:

Figura 4 - Etapas de transformação do produto



Fonte: Elaborado pelo autor

Inicialmente, a matéria-prima é transportada do fornecedor para a Cozinha Central. De acordo com receitas pré-definidas, os ingredientes são preparados por cozinheiros e transformados em refeições. Em seguida, a refeição pronta passa pela etapa de embalagem.

Após embalado, o produto segue para o processo de ultracongelamento, explicado anteriormente. Uma vez congelado, o produto é transportado para o armazém (centro de distribuição), onde é estocado em *freezers*.

Finalmente, os itens são separados por cliente e transportados para o consumidor final.

2.2 Definição do Problema

2.2.1 Logística atual

Atualmente, existe apenas um analista dedicado à área de logística, responsável pelo planejamento diário e controle desta operação.

A operação se inicia com o pedido realizado pelo cliente na plataforma *online*. Se o pedido for realizado antes das 22h, será entregue no dia seguinte. Caso contrário, ele se junta aos pedidos do dia seguinte, sendo entregue após 2 dias. Além disso, o cliente pode escolher entre dois períodos de entrega: tarde (das 14h às 17h) e noite (das 19h às 22h). Caso o cliente tenha restrição de horário, pode solicitar por telefone o recebimento dentro de um intervalo de tempo menor de 1 a 2 horas, sem custo extra.

Na manhã do dia seguinte, o analista extrai uma base no sistema contendo a lista de clientes com endereço, seus respectivos pedidos, a janela de entregas e uma coluna que indica se o modo de pagamento é através de vale-refeição. Como atualmente não é possível efetuar pagamento *online* com vale-refeição, é necessário o transporte da máquina de cartão até o cliente.

Com o auxílio de uma base de endereços dos clientes, o analista plota numa plataforma *online* de mapas a localização de todas as entregas de cada período, como ilustra a Figura 5.

Atualmente, o número de entregas realizadas é, em média, 90 por dia. Com o crescimento acelerado da *startup*, acredita-se que esse número venha a dobrar nos próximos meses.

2.2.2 Problemas encontrados

Alguns problemas foram identificados na atual operação logística. Estes problemas podem ser classificados como problema de nível de serviço, problema de tempo de processamento, problema de custo e problema de gestão.

O problema de nível de serviço está relacionado ao atendimento das rotas às restrições de entrega. Como mencionado anteriormente, restrições de janela de tempo são solicitadas por telefone por alguns clientes. Devido à complexidade da malha de entrega, não é possível disponibilizar a opção de janela de entrega reduzida a todos os clientes no *website*, o que compromete o nível de serviço. Com o aumento de escala, a complexidade aumenta ainda mais.

O problema de tempo de processamento diz respeito ao tempo total de roteirização. Como o processo atual é realizado manualmente, o tempo total é muito elevado (cerca de uma hora para 90 entregas por dia). Com o aumento acelerado no número de entregas por dia, esse processo se tornará inviável.

O problema de custo, por sua vez, refere-se ao custo total de transporte. Com o processo manual, não é possível atingir um custo de transporte próximo ao ótimo. Em grande escala, isso implica num custo adicional de ordem de grandeza considerável.

Por fim, o problema de gestão pode ser visto como um problema de tomada de decisão. Dado que o processo é manual e lento, não é possível simular de forma rápida o custo total de transporte dado um cenário de entregas e de custos. Assim, decisões estratégicas, como por exemplo a de utilizar frota própria ou de terceiros, são difíceis de serem tomadas.

2.3 Escopo e Objetivo do Estudo

O presente trabalho tem como objetivo auxiliar a *startup* de comidas saudáveis congeladas *Liv Up* a operar em grande escala, através do aumento na eficiência da operação logística.

Para atingir este objetivo, uma solução para os problemas identificados na área de logística será desenvolvida e proposta à *startup*.

De modo a solucionar os problemas de nível de serviço, tempo de processamento, custo e gestão, uma ferramenta automatizada de roteirização será desenvolvida. Em seguida, a mesma será utilizada em análises de sensibilidade relacionadas ao *tradeoff* entre custo e nível de serviço.

3 REVISÃO DA LITERATURA

O presente capítulo tem como objetivo introduzir o problema de roteirização de veículos, apresentando sua relevância e as diferentes categorias e tipos encontrados na literatura. Um maior detalhamento será dado para o problema clássico (PRV) e para o Problema de Roteirização de Veículos com Janelas de Tempo (PRVJT), devido à maior relevância para o trabalho.

Devido à alta complexidade computacional desses problemas, geralmente é necessário o uso de métodos heurísticos. Assim, este capítulo também aborda os principais métodos heurísticos de solução do PRVJT encontrados na literatura até hoje.

O conteúdo deste capítulo será, em seguida, usado como base para o desenvolvimento da ferramenta proposta por este trabalho.

3.1 Introdução à roteirização de veículos

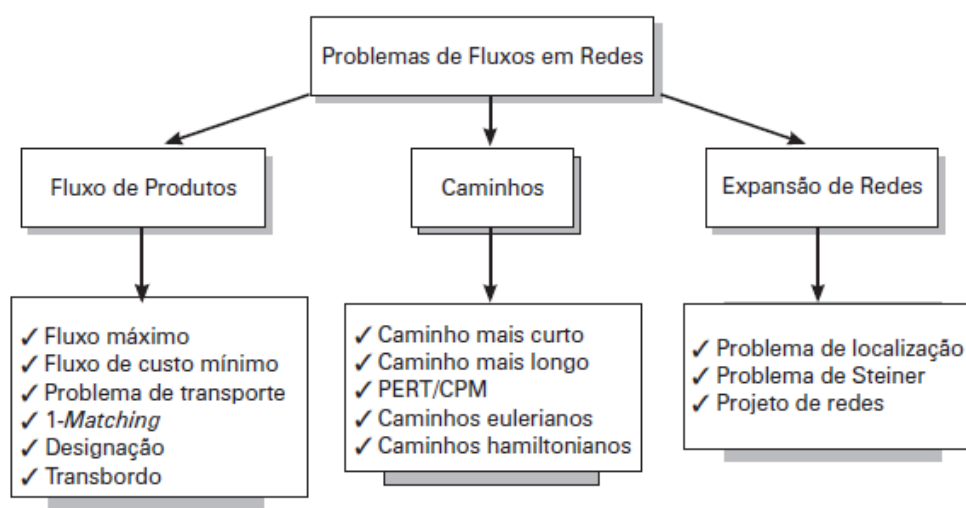
Diversos autores definiram o problema de roteirização de veículos. Christofides (1976) define a roteirização como um problema de fornecimento de bens a clientes geograficamente dispersos usando-se um certo número de veículos que partem de um mesmo depósito, minimizando distância, tempo ou custo. Para Ballou (2001), a roteirização visa buscar os melhores trajetos de um veículo através de uma malha viária, de forma a minimizar o tempo ou a distância. Segundo Goldbarg e Luna (2005), trata-se de determinar sequências de visitas atendendo a uma função objetivo. Desta forma, pode-se definir o problema como a determinação de sequências de visitas a nós (clientes) passando por arcos (malha viária) com o uso de veículos, de forma a minimizar o custo, a distância ou o tempo total, sujeito a restrições.

A preocupação com a roteirização de veículos vem crescendo nas últimas décadas, principalmente pela sua aplicabilidade na logística empresarial. Estima-se que os custos de transporte representam de 1/3 a 2/3 dos custos logísticos de uma empresa (Ballou, 2001). Através de investimentos e estudos nesta área, é possível obter economia de tempo e custo, aumentando a qualidade do serviço e as margens operacionais das empresas e da cadeia de suprimento como um todo. Segundo Golden e Assad (1986), empresas como Du Pont, Kraft e

Chevron obtiveram redução de custo na faixa de 10-15% através de sistemas de roteirização. Desta forma, o problema clássico e suas variantes são bastante estudados na literatura.

O problema de roteirização de veículos pode ser considerado uma extensão do conhecido Problema¹ do Caixeiro Viajante, com restrições adicionais de capacidade dos veículos e demanda dos clientes (Goldbarg; Luna, 2005). De fato, ambos pertencem a uma categoria ampla de problemas de Pesquisa Operacional conhecida como problemas de fluxo em redes. Também são encontrados nessa categoria os problemas clássicos de fluxo máximo, de caminho mínimo, de transporte e de designação (Golden; Ball; Bodin, 1981), além de outros, como observa-se na Figura 6. O denominador comum dessa classe de problemas é a existência de pontos de oferta ligados a pontos de demanda, às vezes com a presença de intermediários, e restrições de equilíbrio de fluxo em cada nó.

Figura 6 - Classificação de problemas de fluxo em redes



Fonte: Goldbarg; Luna, 2005

A primeira formulação do problema de roteirização de veículos encontrada na literatura remete ao fim da década de 50, através da publicação do artigo *The Truck Dispatching Problem* por Dantzig e Ramser (1959). Desde então, muitas abordagens e variações do problema foram

¹ O Problema do Caixeiro Viajante (*Travelling Salesman Problem*, em inglês) consiste em determinar um roteiro que saia da origem, passe por todos os nós de uma rede uma única vez e retorne à origem, de forma a minimizar a distância ou custo total (Goldbarg; Luna, 2005).

desenvolvidas, sendo que o método de Dantzig e Ramser continuou a ser mencionado em estudos posteriores, como por exemplo em *Vehicle Scheduling Revisited* (Waters, 1984).

O problema clássico de roteirização de veículos foi o ponto de partida para diversas variações do mesmo. Esses problemas podem ser classificados em categorias que diferem quanto à localização dos clientes, tipo de frota, tipo de demanda, função objetivo, entre outras dimensões (Bodin; Golden, 1981). Exemplos de variações do problema podem ser observados na Tabela 1.

Para os fins do presente trabalho, merecem destaque o problema clássico de roteirização de veículos e sua variante com janelas de tempo, detalhados nas próximas seções.

3.2 Problema Clássico de Roteirização de Veículos (PRV)

Existem várias maneiras de se formular um problema de roteirização de veículos. Uma das mais conhecidas é a formulação de Fisher e Jaikumar (1981), detalhada a seguir.

Índices

i = nó de origem

j = nó de destino

k = veículo utilizado

Parâmetros

N = número de nós (clientes) do problema (nó 0 corresponde ao depósito central)

K = número de veículos utilizados

Q_k = capacidade do veículo $k = 1 \dots K$

d_i = demanda do cliente $i = 1 \dots N$

c_{ij} = custo do trajeto com origem no nó i e destino no nó j

Tabela 1 - Variações do Problema de Roteirização de Veículos

	Número Roteiros	Localização dos clientes	N. depósitos ou bases	Tipo de demanda	Restrições	N. Entregas por cliente	Variáveis de decisão	Período	Tipo de frota	Número de veículos
TSP	1	nós	1	determinística	—	1	roteiros de entrega	1 dia	1 veículo	limitado
CPP	1	arcos	1	determinística	—	1	roteiros de entrega	1 dia	1 veículo	limitado
MTSP	múltiplos	nós	1	determinística	—	1	roteiros de entrega	1 dia	homogênea	limitado
VRP	múltiplos	nós	1	determinística	capacidade do veículo	1	roteiros de entrega	1 dia	homogênea	limitado
MDVRP	múltiplos	nós	múltiplos	determinística	capacidade do veículo	1	roteiros de entrega	1 dia	homogênea	limitado
CARP	múltiplos	arcos	1	determinística	capacidade do veículo	1	roteiros de entrega	1 dia	homogênea	limitado
SVRP	múltiplos	nós	1	estocástica	capacidade do veículo	1	roteiros de entrega	1 dia	homogênea	limitado
VRPSD	múltiplos	nós	1	determinística	capacidade do veículo	≥ 1	roteiros de entrega quantidade entregue a cada cliente	1 dia	homogênea	limitado
FSVRP	múltiplos	nós	1	determinística	capacidade do veículo	1	roteiros de entrega dimensionamento da frota	1 dia	homogênea	ilimitado
HFFVRP	múltiplos	nós	1	determinística	capacidade do veículo	1	roteiros de entrega	1 dia	heterogênea	limitado
FSMVRP	múltiplos	nós	1	determinística	capacidade do veículo	1	roteiros de entrega dimension. e compos. frota	1 dia	heterogênea	ilimitado
PVRP	múltiplos	nós	1	determinística	capacidade do veículo	1	roteiros de entrega	M dias	homogênea	limitado
TDVRP	múltiplos	nós	1	determinística	capacidade do veículo tempo dependente	1	roteiros de entrega	1 dia	homogênea	limitado
VRPTW	múltiplos	nós	1	determinística	capacidade do veículo janela de tempo	1	roteiros de entrega	1 dia	homogênea	limitado
VRPSTW	múltiplos	nós	1	determinística	capacidade do veículo janela de tempo flexível	1	roteiros de entrega	1 dia	homogênea	limitado
PDP Dial-a-ride	múltiplos	nós	1	determinística	capacidade do veículo precedência entre tarefas	1	roteiros de entrega	1 dia	homogênea	limitado

Legenda:

- TSP: Problema do Caixeiro Viajante
- CPP: Problema do Carteiro Chinês
- MTSP: Problema de Múltiplos Caixeiros Viajantes
- VRP: Problema Clássico de Roteirização de Veículos
- MDVRP: VRP com múltiplos depósitos
- CARP: VRP com demanda em arcos
- SVRP: VRP com demanda estocástica
- VRPSD: VRP com entregas fracionadas
- FSVRP: VRP com dimensionamento de frota homogênea
- HFFVRP: VRP com frota heterogênea fixa
- FSMVRP: VRP com dimensionamento de frota heterogênea
- PVRP: VRP periódico
- TDVRP: VRP com tempo dependente
- VRPTW: VRP com janelas de tempo
- VRPSTW: VRP com janelas de tempo flexíveis
- PDP / dial-a-ride: Problema de coleta e entrega (VRP + precedência)

Fonte: Tese de doutorado de Belfiore (2006)

Variáveis de decisão

$$x_{ijk} \begin{cases} 1, \text{ se o arco } (i,j) \text{ é percorrido pelo veículo } k \\ 0, \text{ caso contrário} \end{cases}$$

$$y_{ik} \begin{cases} 1, \text{ se o veículo } k \text{ é utilizado para a entrega do cliente } i \\ 0, \text{ caso contrário} \end{cases}$$

Função objetivo

$$\min \sum_{i=0}^N \sum_{j=0}^N \sum_{k=1}^K c_{ij} x_{ijk} \quad (3.1)$$

Restrições

$$\sum_{i=1}^N d_i y_{ik} \leq Q_k \quad k = 1 \dots K \quad (3.2)$$

$$\sum_{k=1}^K y_{0k} = K \quad (3.3)$$

$$\sum_{k=1}^K y_{ik} = 1 \quad i = 1 \dots N \quad (3.4)$$

$$\sum_{i=0}^N x_{ijk} = y_{jk} \quad j = 0 \dots N, k = 1 \dots K \quad (3.5)$$

$$\sum_{j=0}^N x_{ijk} = y_{ik} \quad i = 0 \dots N, k = 1 \dots K \quad (3.6)$$

$$\sum_{i,j \in S \times S} x_{ijk} \leq |S| - 1 \quad S \subseteq \{1, \dots, N\}, 2 \leq |S| \leq N-1, k = 1 \dots K \quad (3.7)$$

$$x_{ijk}, y_{ik} \in \{0,1\} \quad (3.8)$$

A função objetivo (3.1) minimiza o custo total de todos os trajetos dos K veículos.

O conjunto de restrições (3.2) garante que a capacidade de cada veículo é igual ou superior à soma das demandas dos clientes atendidos pelo veículo.

Os conjuntos de restrições (3.3) e (3.4) garantem, respectivamente, que todos os veículos saiam do depósito central e que cada cliente seja atendido por um único veículo.

Os conjuntos de restrições (3.5) e (3.6) são restrições de fluxo em rede, garantindo que cada veículo deixe um nó somente se entrar no mesmo, e apenas uma vez.

O conjunto de restrições (3.7) garante que, para cada subconjunto de clientes S , não se forme um trajeto fechado (subrota) passando por esses clientes. É fácil perceber que à medida que N

aumenta, o número de subconjuntos de S aumenta consideravelmente, o que justifica a alta complexidade computacional do problema.

Por fim, o conjunto de restrições (3.8) indica o tipo das variáveis, que no caso são binárias.

Outra maneira de se formular o problema foi descrita em Arenales *et al.* (2007). Essa formulação difere da anterior em três aspectos: elimina-se a variável de decisão y_{ik} , adiciona-se um novo conjunto de restrições de tempo total máximo do trajeto e adiciona-se o nó $n+1$, correspondente ao depósito central. Neste caso, o nó 0 é sempre a origem (arco sai do depósito) e o nó $n+1$ é sempre o destino (arco chega no depósito).

Índices

i = nó de origem

j = nó de destino

k = veículo utilizado

$i, j \in \{0, \dots, N+1\}, i \neq j, i \neq N+1, j \neq 0$

Parâmetros

N = número de nós (clientes) do problema (nó 0 e $n+1$ correspondem ao depósito central)

K = número de veículos utilizados

T = tempo total máximo de uma rota

Q_k = capacidade do veículo $k = 1 \dots K$

d_i = demanda do cliente $i = 1 \dots N$

c_{ij} = custo do trajeto com origem no nó i e destino no nó j

t_{ij} = tempo de viagem do nó i ao nó j , incluindo o tempo de serviço em i

Variável de decisão

$$x_{ijk} \begin{cases} 1, & \text{se o arco } (i,j) \text{ é percorrido pelo veículo } k \\ 0, & \text{caso contrário} \end{cases}$$

Função objetivo

$$\min \sum_{i=0}^N \sum_{j=1}^{N+1} \sum_{k=1}^K c_{ij} x_{ijk} \quad (3.9)$$

Restrições

$$\sum_{k=1}^K \sum_{j=1}^{N+1} x_{ijk} = 1 \quad i = 1 \dots N \quad (3.10)$$

$$\sum_{i=1}^N d_i \sum_{j=1}^{N+1} x_{ijk} \leq Q_k \quad k = 1 \dots K \quad (3.11)$$

$$\sum_{i=0}^N \sum_{j=1}^{N+1} t_{ij} x_{ijk} \leq T \quad k = 1 \dots K \quad (3.12)$$

$$\sum_{j=1}^{N+1} x_{0jk} = 1 \quad k = 1 \dots K \quad (3.13)$$

$$\sum_{i=0}^N x_{ihk} - \sum_{j=1}^{N+1} x_{hjk} = 0 \quad h = 1 \dots N, k = 1 \dots K \quad (3.14)$$

$$\sum_{i=0}^N x_{i,N+1,k} = 1 \quad k = 1 \dots K \quad (3.15)$$

$$\sum_{i,j \in S \times S} x_{ijk} \leq |S| - 1 \quad S \subseteq \{1, \dots, N\}, 2 \leq |S| \leq N-1, k = 1 \dots K \quad (3.16)$$

$$x_{ijk} \in \{0,1\} \quad (3.17)$$

A função objetivo do problema (3.9) é equivalente a (3.1).

O conjunto de restrições (3.10) é equivalente ao conjunto (3.4) e o conjunto (3.11) equivale ao (3.2).

O conjunto de restrições (3.12) garante que, para cada veículo, o tempo total da rota não ultrapasse o limite superior T .

Os conjuntos de restrições (3.13) a (3.15) são restrições de fluxo em rede equivalentes a (3.5) e (3.6), exigindo que cada veículo parta do depósito ($i=0$) somente uma vez, deixe o nó h somente se entrar nesse nó e retorne ao depósito ($j = N+1$) somente uma vez. O conjunto de restrições (3.15) não é necessário, mas é inserido somente para enfatizar o retorno ao nó $N+1$.

Os conjuntos de restrições (3.16) e (3.17) são equivalentes aos conjuntos (3.7) e (3.8).

O conjunto de restrições (3.16), além de ter forte caráter combinatório, é difícil de ser implementado em linguagens de programação. Uma alternativa para este conjunto de restrições foi proposta por Miller *et al* (1960). Em seguida, diversos trabalhos contribuíram com melhorias a essa proposta, como em *Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints* (Desrochers; Laporte, 1991).

3.3 Problema de Roteirização de Veículos com Janelas de Tempo (PRVJT)

O problema de roteirização de veículos com janelas de tempo é uma extensão do problema clássico e é explicada abaixo, de acordo com descrição de Arenales *et al.* (2007).

O conjunto de restrições adicional do problema refere-se ao horário de chegada nos nós, que deve estar dentro de um intervalo de tempo $[a_i, b_i]$, $i \in \{1, \dots, N\}$. Os veículos saem do depósito no instante 0 e devem retornar durante o intervalo $[a_{N+1}, b_{N+1}]$. Além disso, um veículo pode chegar no cliente antes de sua janela de tempo e esperar sem custo.

Todas as restrições do problema clássico descrito em Arenales *et al.* (2007) se mantêm, exceto os conjuntos (3.12) e (3.16), que não são mais necessários.

Índices

i = nó de origem

j = nó de destino

k = veículo utilizado

$i, j \in \{0, \dots, N+1\}$, $i \neq j$, $i \neq N+1$, $j \neq 0$

Parâmetro adicional

$[a_i, b_i]$ = intervalo de tempo de chegada do veículo ao nó i

Variável de decisão adicional

s_{ik} = instante em que o veículo k chega ao nó i

Função objetivo

$$\min \sum_{i=0}^N \sum_{j=1}^{N+1} \sum_{k=1}^K c_{ij} x_{ijk} \quad (3.18)$$

Restrições adicionais

$$s_{ik} + t_{ij} \leq s_{jk} + (1 - x_{ijk})M_{ij} \quad \forall(i,j), k = 1 \dots K, M_{ij} = \max \{b_i + s_{ik} + t_{ij}, 0\} \quad (3.19)$$

$$a_i \leq s_{ik} \leq b_i \quad k = 1 \dots N+1, k = 1 \dots K \quad (3.20)$$

O conjunto de restrições (3.19) garante que se o veículo k deixa o nó i e chega ao nó j , então não pode chegar em j antes de $s_{ik} + t_{ij}$. Isso também elimina as subrotas do problema, tornando desnecessário o conjunto (3.16).

O conjunto de restrições (3.20) garante que todas as janelas de tempo sejam respeitadas, inclusive o tempo total da viagem. Isso elimina a necessidade do conjunto (3.12).

3.4 Métodos Heurísticos

O problema de roteirização de veículos possui restrições de caráter combinatório, o que o torna complexo do ponto de vista computacional (Goldbarg; Luna, 2005).

A complexidade computacional pode ser entendida de forma simplificada como o consumo de tempo (medido pelo número de operações) de um determinado algoritmo de solução de um problema. É natural que quanto maior o tamanho das instâncias do problema, maior será o consumo de tempo. Porém, dependendo da relação entre o tamanho das instâncias e o consumo de tempo, pode-se dizer que um algoritmo é tratável ou não. Caso o tempo de processamento seja uma função polinomial do tamanho da instância, diz-se que o algoritmo é polinomial e que o problema pertence à classe P de problemas. Neste caso, ele é considerado um algoritmo “rápido” e tratável (Sipser, 2006).

Além da classe P de problemas, existem outras classes, como a dos problemas NP, NP-completos e NP-difíceis. Brevemente, a classe NP representa os problemas de decisão que podem ser resolvidos em tempo polinomial não determinístico. A classe de problemas NP-completos representa todos os problemas em NP tal que é possível reduzir qualquer outro

problema NP a ele em tempo polinomial. Por fim, a classe NP-difíceis são os problemas de busca tal que existe algum problema NP-completo redutível a eles (Garey; Johnson, 1979).

A maioria dos problemas de roteirização de veículos pertence à classe de problemas NP-difíceis, inclusive o Problema de Roteirização de Veículos com Janelas de Tempo. Isso significa que, até o presente momento, não se conhece nenhum algoritmo que resolva o problema em tempo polinomial. Desta forma, o consumo de tempo varia exponencialmente com o tamanho das instâncias do problema e sua solução é intratável através de métodos exatos (Lenstra; Rinnooy Kan, 1981).

Por este motivo, a Pesquisa Operacional desenvolve estratégias para contornar este problema. Para Cunha (1997), os métodos de solução de problemas de roteirização podem ser divididos em três categorias: métodos exatos, métodos heurísticos e métodos emergentes (meta-heurísticas). As meta-heurísticas são mais complexas e, geralmente, são aplicadas após métodos heurísticos visando modificar e aprimorar a solução.

A etimologia da palavra “heurística” vem do grego *Heuriskein*, significando “descobrir”. Assim, está relacionada à descoberta e resolução de problemas.

Segundo Nicholson (1974), heurísticas são procedimentos intuitivos de resolução de problemas de forma a se chegar numa solução razoável de maneira inteligente. Para Reeves (1993), heurísticas são técnicas que buscam soluções ótimas ou boas (mas não as garantem) e que são aceitáveis do ponto de vista do custo operacional. O que difere os métodos heurísticos de métodos aproximativos é que, no primeiro caso, não se fornecem limites de qualidade da solução comprovados matematicamente.

Desta forma, métodos heurísticos podem ser entendidos como métodos de solução alternativos que buscam reduzir a complexidade e custos operacionais em troca da perda de garantia de obtenção da solução ótima do problema.

Diferentes abordagens heurísticas podem ser encontradas na literatura. Uma possível classificação de métodos heurísticos foi proposta por Bott e Ballou (1986) *apud* Belfiore (2006):

- **Procedimentos de Economia ou Inserções:** esta abordagem constroi uma solução através da progressão de uma configuração para outra em diversos passos, baseando-se

no critério de minimização da função objetivo do problema. Não é necessário que as configurações intermediárias sejam factíveis. Alguns exemplos desta abordagem podem ser encontrados nos trabalhos de Mole e Jameson (1976) e na conhecida heurística de Clarke e Wright (1964).

- **Procedimentos de Primeiro Agrupar e Depois Roteirizar:** esta abordagem consiste em duas etapas. A primeira visa agrupar os nós ou arcos de demanda segundo algum critério de proximidade, formando *clusters*. A segunda visa construir rotas econômicas independentes para cada agrupamento. Um exemplo dessa abordagem pode ser encontrado no trabalho de Gillet e Miller (1974).
- **Procedimentos de Primeiro Roteirizar e Depois Agrupar:** esta abordagem realiza a operação inversa da anterior. Em um primeiro momento, cria-se uma rota (geralmente infactível) envolvendo todos os pontos de demanda. Numa segunda etapa, a rota é dividida em rotas menores e factíveis. Um exemplo dessa abordagem pode ser encontrado no trabalho de Newton e Thomas (1974).
- **Procedimentos de Melhoria ou Troca:** nesta abordagem, parte-se de uma solução inicial viável e, a cada passo, altera-se a solução de modo a obter uma nova solução viável de menor custo. Dentro desta categoria estão presentes as Heurísticas por Aproximação através de Programação Matemática, as Heurísticas de Busca Local e as Heurísticas por Otimização Iterativa.

Alguns autores indicam ainda uma outra categoria, o Procedimento de Relaxação. Nesta abordagem, algumas restrições do problema são relaxadas de forma a obter uma solução factível facilmente (Belfiore, 2006).

Os procedimentos acima descritos não se restringem a problemas de roteirização de veículos, mas são muito utilizados na solução dos mesmos. Um extenso trabalho de revisão da literatura sobre métodos heurísticos aplicados ao Problema de Roteirização de Veículos com Janelas de Tempo foi realizado por Bräysy e Gendreau (2005) e Belfiore (2006) e alguns pontos são mencionados abaixo.

Um dos trabalhos mais conhecidos nesta área foi publicado por Solomon (1987). Solomon desenvolveu sete heurísticas construtivas para resolver o PRVJT, considerando frota homogênea, número de veículos ilimitado e objetivando minimizar tanto a distância total quanto o tempo total das rotas:

- **Heurística de economia:** extensão da heurística de Clarke e Wright (1964), considerando restrições de tempo.
- **Heurística de economia com limite de tempo de espera:** semelhante à anterior, com um critério adicional de limite de tempo de espera.
- **Heurística do vizinho mais próximo com orientação temporal:** heurística de inserção com critérios de proximidade e tempo.
- **Heurística de inserção I1:** minimiza o acréscimo de tempo e distância causados pela inserção de um cliente.
- **Heurística de inserção I2:** minimiza o tempo e a distância total da rota.
- **Heurística de inserção I3:** semelhante à heurística I1, com o acréscimo de um critério de urgência de atendimento ao cliente.
- **Heurística da varredura com orientação temporal:** utiliza o Procedimento de Primeiro Agrupar e Depois Roteirizar, valendo-se da heurística de Gillet e Miller (1974) para a primeira parte e da heurística de inserção I1 para a segunda parte.

Solomon testou as heurísticas propostas em 56 problemas que diferiam quanto à posição dos clientes, porcentagem de clientes com janelas de tempo e horizonte de planejamento. Sua conclusão foi que a heurística de inserção I1 apresentou os melhores resultados, seguida da heurística de inserção I2. Segue abaixo a heurística de inserção I1 de Solomon (1987) *apud* Dullaert (2000) em mais detalhes:

Heurística de Inserção I1

Nesta heurística de inserção sequencial, para cada rota primeiro seleciona-se o cliente inicial através de um critério de inicialização e depois monta-se a rota com a inserção de um cliente ainda não alocado entre dois clientes da rota em criação.

Os critérios de inicialização são exclusivos (apenas um é usado) e podem ser:

- cliente mais distante ainda não alocado.
- cliente com o menor valor de fim da janela de tempo.

Em seguida, insere-se sequencialmente clientes u entre dois clientes adjacentes i_{p-1} e i_p da rota em construção $(i_0, i_1, \dots, i_m)$, sendo i_0 (origem) e i_m (destino) o depósito central. Considera-se como conhecidos os valores de início do serviço em cada cliente, $b(i_r)$, e os tempos de espera

$w(i_r)$. Para inserir um cliente u entre os clientes i e j , o método utiliza 2 critérios a cada iteração, os critérios $c_1(i, u, j)$ e $c_2(i, u, j)$.

O critério c_1 , uma generalização do método de Mole e Jameson (1976), é usado para calcular, para cada cliente u ainda não alocado, a melhor posição de inserção (considerando o custo espacial-temporal de inserção) entre dois clientes i e j da rota em desenvolvimento. Ele é calculado da seguinte forma:

$$c_1(i(u), u, j(u)) = \min |c_1(i_{p-1}, u, i_p)| \quad p = 1 \dots m \quad (3.21)$$

onde

$$c_1(i, u, j) = \alpha_1 c_{11}(i, u, j) + \alpha_2 c_{12}(i, u, j) \quad \alpha_1, \alpha_2 \geq 0 \text{ e } \alpha_1 + \alpha_2 = 1 \quad (3.22)$$

$$c_{11}(i, u, j) = d_{iu} + d_{uj} - \mu d_{ij} \quad \mu \geq 0 \quad (3.23)$$

$$c_{12}(i, u, j) = b'_j - b_j \quad (3.24)$$

O subcritério c_{11} representa o acréscimo na distância total da rota pela inserção de u . O subcritério c_{12} representa o atraso no atendimento do cliente j após inserção de u . Desta forma, o critério c_1 representa uma média ponderada entre o acréscimo de distância e tempo na rota pela inserção de u e busca-se encontrar as posições $p-1$ e p que minimizem esse acréscimo ponderado.

Dada a melhor posição de inserção para cada cliente u ainda não alocado, o critério c_2 , por sua vez, é usado para determinar o melhor cliente u^* a ser inserido na rota em criação. Ele é calculado da seguinte forma:

$$c_2(i(u^*), u^*, j(u^*)) = \max |c_2(i(u), u, j(u))| \quad u \text{ factível e ainda não roteirizado} \quad (3.25)$$

$$c_2(i, u, j) = \lambda d_{0u} - c_1(i, u, j) \quad \lambda \geq 0 \quad (3.26)$$

O critério c_2 é uma generalização do algoritmo de economia de Clarke e Wright (1964), na medida em que maximiza o benefício gerado pela inserção de um cliente na rota em construção ao invés de numa nova rota direta. Além da maximização do critério c_2 , vale ressaltar que o

cliente inserido u deve ser factível em termos de janelas de tempo da rota e capacidade do veículo.

*

A heurística de inserção I1, por ter obtido os melhores resultados, foi extensamente utilizada por diversos autores, seja como base de comparação de resultados, seja como ponto de partida para melhorias incrementais.

Solomon, Baker e Schaffer (1988) desenvolveram heurísticas de melhoria para o PRVJT com frota homogênea, restrições de capacidade e janelas de tempo. A metodologia é baseada em trocas de arco *k-opt*, com procedimentos que reduzem o tempo computacional através da eliminação de testes de viabilidade desnecessários.

Thompson e Psaraftis (1993) implementaram um procedimento de melhoria por troca de arcos baseado na transferência cíclica para o PRV e PRVJT. O método foi aplicado ao conjunto de problemas de Solomon (1987) e obteve resultados superiores aos de Solomon, Baker e Schaffer (1988).

Potvin e Rousseau (1993) adaptaram a heurística de inserção I1 introduzindo um método de construção paralela. Dessa forma, a qualidade das rotas dos últimos clientes alocados é melhorada. Aplicando o método aos problemas de Solomon (1987), os autores concluíram que houve melhoria nos resultados para problemas com clientes mais dispersos geograficamente.

Potvin e Rousseau (1995) desenvolveram heurísticas de melhoria baseadas em trocas de arco tomando como ponto de partida a heurística de inserção I1.

Russell (1995) inseriu métodos de melhoria global em um algoritmo construtivo similar ao de Potvin e Rousseau (1993). O autor concluiu que esta abordagem híbrida é superior ao método tradicional de duas fases (construir e melhorar).

Cunha e Gualda (1997) desenvolveram três heurísticas baseadas na relaxação lagrangiana. As restrições de atendimento a clientes uma única vez foram relaxadas e utilizou-se um algoritmo de etiquetamento permanente. Uma das três heurísticas, baseada no agrupamento e alocação sequencial, obteve resultados superiores aos de Solomon (1987). As outras duas obtiveram resultados superiores para alguns conjuntos de problema.

Dullaert (2000) argumenta que o algoritmo de inserção I1 subestima o incremento de tempo da inserção de um cliente entre o depósito e a primeira entrega da rota. Dessa forma, o autor propõe um novo critério de inserção de tempo para o algoritmo, obtendo resultados a partir de 50% superiores, segundo Bräysy e Gendreau (2005).

Cordone e Calvo (2001) propõem uma heurística de melhoria baseada em trocas *k-opt* combinada com um procedimento de minimização do número de rotas. Os autores partem de uma solução inicial obtida pela heurística de inserção I1 e usam técnicas de implementação que reduzem o tempo computacional.

Bräysy (2001) utiliza uma abordagem de três fases para uma heurística de melhoria baseada em busca local. Na primeira fase, soluções iniciais com diferentes parâmetros são criadas com o uso dos trabalhos de Solomon (1987) e Russell (1995). Na segunda fase, o número de rotas é reduzido através de uma abordagem baseada em Clarke e Wright (1964). Por fim, na terceira fase, a distância total é minimizada.

Além de métodos heurísticos, recentemente diversos autores desenvolveram métodos meta-heurísticos de resolução, como Gehring e Homberger (2001), Repoussis *et al* (2009), Nagata *et al* (2010) e Vidal *et al* (2013).

Segue na Tabela 2 uma lista (não exaustiva) dos principais trabalhos relacionados a heurísticas para o PRVJT:

Tabela 2 - Principais aplicações de heurísticas para o PRVJT

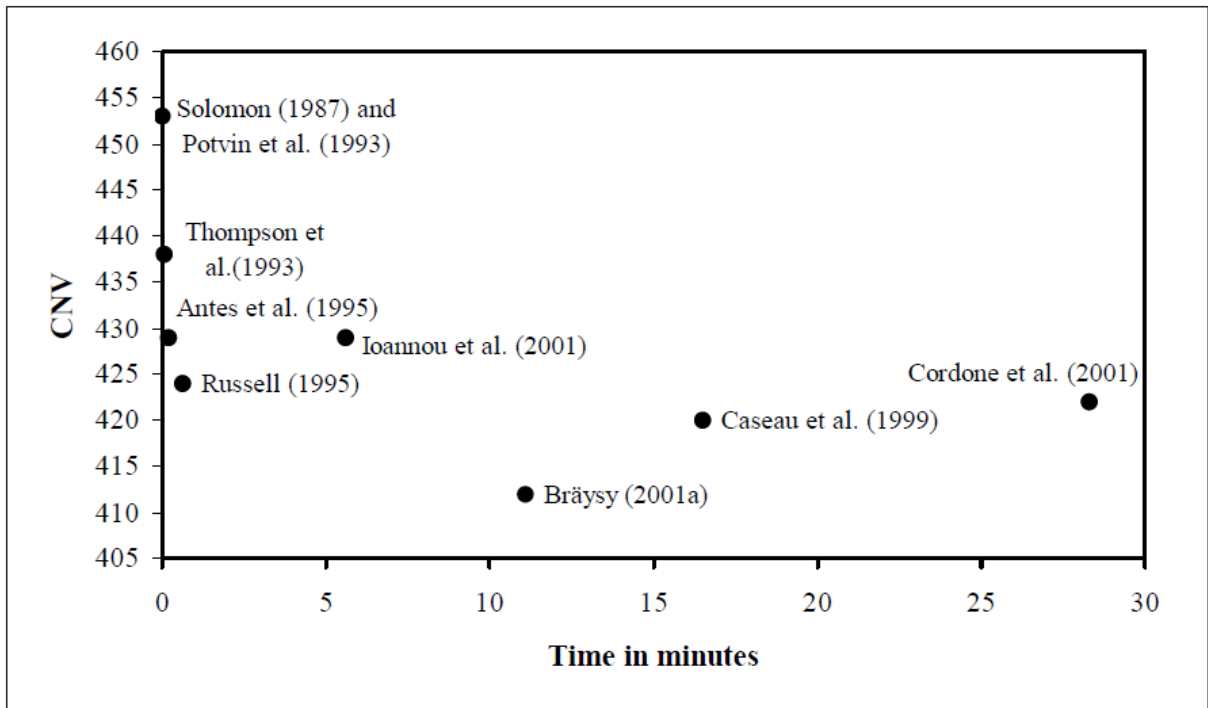
Ano	Pesquisador	Trabalho
1986	Solomon	Heurísticas construtivas e de melhoria
1987	Solomon	7 heurísticas construtivas
1988	Solomon, Baker e Schaffer	Heurísticas de melhoria
1993	Thompson e Psaraftis	Transferência cíclica
1993	Potvin e Rousseau	Algoritmo de construção paralela
1993	Balakrishnan	Algoritmos heurísticos
1993	Thangiah et al.	Heurística de duas fases
1995	Russell	Algoritmo híbrido
1995	Potvin e Rousseau	Heurísticas de melhoria
1995	Antes e Derigs	Heurística de construção e melhoria
1995	Frizzell e Giffin	Heurísticas de construção e melhoria

1997	<i>Kohl e Madsen</i>	Relaxação lagrangeana
1997	<i>Marshall et al.</i>	Algoritmos de otimização
1997	<i>Cunha e Gualda</i>	Relaxação lagrangeana
1998	<i>Shaw</i>	<i>Large Neighbourhood Search</i>
1999	<i>Schulze e Fahle</i>	Algoritmo paralelo
1999	<i>Caseau e Laburthe</i>	Algoritmo heurístico
2000	<i>Dullaert</i>	Algoritmo de inserção
2001	<i>Cordone e Calvo</i>	Algoritmo heurístico
2001	<i>Bräysy</i>	Busca em vizinhança variável
2001	<i>Ioannou et al.</i>	Algoritmo guloso
2001	<i>Tan et al.</i>	Algoritmos heurísticos
2004	<i>Ioannou e Kritikos</i>	Síntese de várias heurísticas de alocação

Fonte: Elaborado pelo autor

De acordo com Bräysy e Gendreau (2005), as heurísticas mais eficientes para o PRVJT são as de Russell (1995) e Bräysy (2001). No Gráfico 2 pode ser observada uma comparação de alguns algoritmos heurísticos aplicados a um mesmo conjunto de problemas.

Gráfico 2 - Comparação de algoritmos heurísticos quanto à eficiência (ordenada) e complexidade¹ (abscissa). Quanto menor o valor da ordenada, maior é a eficiência do algoritmo.



Fonte: Bräysy e Gendreau (2005)

¹ O termo CNV (abscissa) representa o número acumulado de veículos necessário para a resolução do conjunto de problemas (*Cumulative Number of Vehicles*).

4 MODELAGEM EM PROGRAMAÇÃO LINEAR INTEIRA MISTA

Num primeiro momento, um método exato de resolução será utilizado para a obtenção das soluções do problema. A modelagem do mesmo tomará como base a revisão da literatura do capítulo anterior e será adaptada ao problema específico da *startup*. Em seguida, os parâmetros de entrada serão coletados e processados para posterior uso nos testes computacionais.

Como o método exato retorna a solução ótima do problema, ele é preferível em relação a métodos heurísticos. Porém, nem sempre é possível ou viável sua utilização na prática devido a questões de tempo de processamento e recursos.

Conforme explicado na seção 3.4, o problema de roteirização de veículos é classificado como NP-difícil, sendo portanto complexo do ponto de vista computacional. Desta forma, a hipótese inicial é de que o uso do método exato será inviável na prática.

Caso isso se mostre verdadeiro, o desenvolvimento deste capítulo não será inútil. Como a solução ótima do problema é alcançada, ela servirá como base de comparação para avaliar a eficácia de métodos heurísticos.

Assim, o presente capítulo tem como objetivo testar a hipótese de inviabilidade do uso do método exato e, eventualmente, servir como base de avaliação da eficácia de métodos aproximados.

4.1 Formulação do Problema

Como explicado na seção 2.2, a roteirização das entregas é feita em dois períodos por dia e apresenta as seguintes características:

- Busca-se minimizar o custo total das entregas.
- O número de veículos é ilimitado e todos são de mesma capacidade.
- Não há custo fixo por veículo utilizado.
- Todas as entregas partem de um único depósito.
- Não é necessário voltar ao depósito após a última entrega, exceto quando o veículo deva retornar com a máquina de cartão de vale-refeição.

- Os custos dependem do ponto de origem e destino e são fornecidos pela empresa terceirizada de entregas.
- Os clientes estão geograficamente dispersos, sendo que o número de clientes e localização não é fixo a cada roteirização, podendo variar de um dia para o outro.
- A demanda é determinística.
- Apenas uma entrega é feita por cliente.
- O tempo de cada trajeto é dado pelo tempo de deslocamento somado ao tempo de serviço.
- Alguns clientes devem receber o produto dentro de um intervalo de tempo pré-estabelecido por eles.
- O tempo total da rota não pode ultrapassar um limite superior, para cada veículo.

Dadas as características acima descritas, pode-se concluir que o modelo mais adequado para a situação atual é baseado no Problema de Roteirização de Veículos com Janelas de Tempo. Assim, a modelagem de Programação Linear Inteira Mista descrita em Arenales *et al.* (2007) e detalhada na seção 3.3 será utilizada neste capítulo.

No entanto, esta modelagem deve ser adaptada à situação real da *startup*, levando em conta o retorno do veículo ao depósito central somente quando houver o transporte de máquina de cartão pelo mesmo.

Caso pelo menos um dos clientes da rota solicitar máquina de vale-refeição para efetuar o pagamento do pedido, o veículo deverá retornar ao depósito central para a devolução da máquina, implicando na cobrança de uma taxa de retorno. Desta forma, a existência da taxa de retorno não depende do último cliente, mas de todos os clientes da rota. Assim, é necessário saber, para cada rota (ou analogamente para cada veículo), se existe algum cliente que solicitou máquina de cartão. Para tanto, basta criar uma variável binária de decisão r_k que vale 1 se a rota do veículo k possui ao menos um cliente que solicitou máquina de cartão e 0, caso contrário.

Para o cálculo de r_k , um novo parâmetro de entrada deve ser inserido, correspondente às solicitações de máquina de vale-refeição pelo cliente. Assim, o parâmetro binário v_i deve ser adicionado ao modelo, valendo 1 se o cliente i solicitou a máquina de cartão e 0, caso contrário.

A matriz-custo é um parâmetro do modelo e, portanto, é gerada antes do conhecimento das rotas da solução. Por este motivo, ela considera os custos de retorno para todos os clientes. No

entanto, estes custos de retorno só devem ser somados na função objetivo caso o veículo transporte a máquina de vale-refeição ($r_k = 1$). Assim, uma alternativa é adicionar restrições à variável de decisão x_{ijk} de forma a retirar os trajetos de retorno ao depósito quando o veículo não transportar máquina de vale-refeição.

Segue abaixo formulação adaptada para o problema:

Parâmetros

N = número de nós (clientes) do problema (nó 0 e $n+1$ correspondem ao depósito central)

K = número de veículos utilizados

Q_k = capacidade do veículo $k = 1 \dots K$

d_i = demanda do cliente $i = 1 \dots N$

c_{ij} = custo do trajeto com origem no nó i e destino no nó j , incluindo as taxas de retorno

t_{ij} = tempo de viagem do nó i ao nó j , incluindo o tempo de serviço em i

$[a_i, b_i]$ = intervalo de tempo de chegada do veículo ao nó i

v_i = binário que vale 1 se o cliente i solicitou máquina de cartão e 0, caso contrário

Variáveis de decisão

$$x_{ijk} \begin{cases} 1, & \text{se o arco } (i,j) \text{ é percorrido pelo veículo } k \\ 0, & \text{caso contrário} \end{cases}$$

$$r_k \begin{cases} 1, & \text{se o veículo } k \text{ transportar máquina de vale-refeição} \\ 0, & \text{caso contrário} \end{cases}$$

s_{ik} = instante em que o veículo k chega ao nó i

Função objetivo

$$\min \sum_{i=0}^N \sum_{j=1}^{N+1} \sum_{k=1}^K c_{ij} x_{ijk} \quad (4.1)$$

Restrições

$$\sum_{k=1}^K \sum_{j=0}^{N+1} x_{ijk} = 1 \quad i = 1 \dots N \quad (4.2)$$

$$\sum_{i=1}^N d_i \sum_{j=0}^{N+1} x_{ijk} \leq Q_k \quad k = 1 \dots K \quad (4.3)$$

$$\sum_{j=1}^{N+1} x_{0jk} = 1 \quad k = 1 \dots K \quad (4.4)$$

$$\sum_{i=0}^{N+1} x_{ihk} - \sum_{j=0}^{N+1} x_{hjk} = 0 \quad h = 1 \dots N, k = 1 \dots K \quad (4.5)$$

$$s_{ik} + t_{ij} \leq s_{jk} + (1 - x_{ijk})M_{ij} \quad \forall (i,j), k = 1 \dots K, M_{ij} = \max \{b_i + s_{ik} + t_{ij}, 0\} \quad (4.6)$$

$$a_i \leq s_{ik} \leq b_i \quad i = 1 \dots N+1, k = 1 \dots K \quad (4.7)$$

$$r_k \leq \sum_{i=1}^N v_i \sum_{j=1}^{N+1} x_{ijk} \quad k = 1 \dots K \quad (4.8)$$

$$r_k \geq (\sum_{i=1}^N v_i \sum_{j=1}^{N+1} x_{ijk})/N \quad k = 1 \dots K \quad (4.9)$$

$$x_{i0k} \leq 1 - r_k \quad i = 1 \dots N, k = 1 \dots K \quad (4.10)$$

$$x_{ijk}, r_k \in \{0,1\}, s_{ik} \in R \quad (4.11)$$

Os conjuntos de restrições (4.2) a (4.7) são equivalentes aos enunciados no capítulo anterior.

Retomando de forma breve, o conjunto de restrições (4.2) garante que cada cliente seja atendido apenas uma vez, o conjunto (4.3) garante que a demanda atendida por cada veículo não ultrapasse sua capacidade, o conjunto (4.4) estabelece que o nó 0 seja utilizado exatamente uma vez em cada rota, o conjunto (4.5) apresenta restrições de fluxo em rede e, por fim, os conjuntos (4.6) e (4.7) garantem que as janelas de tempo sejam respeitadas.

Além da criação da variável de decisão r_k , três novos conjuntos de restrições foram criados: (4.8) a (4.10).

No primeiro conjunto, (4.8), a somatória da parte direita da desigualdade representa o número de clientes que solicitaram máquina de cartão, para cada rota. Assim, se pelo menos 1 cliente solicitou a máquina, ela é maior que 0, senão ela vale 0. Portanto, o conjunto de restrições (4.8) atribui o valor 0 a r_k para rotas sem transporte de máquina de cartão.

Já no segundo conjunto, (4.9), a somatória da parte direita da desigualdade é semelhante à do conjunto (4.8), mas é dividida pelo número de clientes N . Desta forma, ela vale 0 para rotas sem o transporte de máquina de cartão e é maior que 0 e menor ou igual a 1 para rotas com este

transporte. Portanto, o conjunto de restrições (4.9) atribui o valor 1 para rotas com transporte de máquina de cartão.

Por fim, é preciso eliminar os trajetos de retorno ao depósito central quando o veículo não transportar máquina de vale-refeição ($r_k = 0$). Porém, para que as restrições de fluxo em rede (3.14) sejam satisfeitas, os trajetos que não retornem ao depósito devem retornar para algum nó. Desta forma, considera-se dois nós de retorno, $j = 0$ e $j = N+1$, e convencionam-se que se o trajeto termina em 0, o veículo não retorna ao depósito central e se o trajeto termina em $N+1$, o veículo retorna e a taxa de retorno é aplicada. Assim, o conjunto de restrições (4.10) determina que, se um veículo retorna para o depósito, o nó final não poderá ser o 0. Dado que o nó final deve ser necessariamente ou 0 ou $N+1$, isso equivale a dizer que o nó final deverá ser $N+1$. Não é preciso fixar o nó 0 como final para veículos que não retornem, pois isto é feito automaticamente pela função objetivo de minimização do custo total.

4.2 Coleta e Processamento de Dados

Para a realização dos experimentos computacionais do modelo de otimização, é preciso fornecer os parâmetros de entrada do modelo, sendo eles: o número de clientes a serem roteirizados, a demanda de cada cliente, os custos de transporte entre cada par de clientes, a

capacidade de cada veículo, o número total de veículos, o tempo de deslocamento e serviço entre cada par de clientes e as janelas de entrega de cada cliente.

4.2.1 *Número de clientes*

Na maioria das aplicações do problema de roteirização de veículos, os pontos a serem atendidos são fixos. Assim, o problema atual se diferencia dos demais na medida em que os pontos de entrega variam de um dia para o outro.

Desta forma, o número de clientes deve ser sempre atualizado de acordo com a demanda diária. Atualmente, este número situa-se em torno de 90 clientes por dia, como mencionado anteriormente.

4.2.2 *Número de veículos*

A frota de veículos é ilimitada e não há custo fixo associado a cada veículo. Desta forma, pode-se usar quantos veículos forem necessários para minimizar o custo total.

O número máximo de veículos necessários para atender uma malha de clientes assumindo carga não fracionada é equivalente ao número de clientes. Nessa situação, cada veículo realizaria apenas uma entrega. Assim, este parâmetro de entrada é fornecido com o valor igual ao número de clientes, mesmo que a solução obtida não utilize todos os veículos

4.2.3 *Capacidade dos veículos*

Os veículos utilizados para o transporte dos produtos são motocicletas fornecidas por uma empresa terceirizada. Segundo a *Liv Up*, a capacidade máxima de cada uma é de cerca de R\$700. Dado que os preços dos produtos variam pouco e se encontram em torno de R\$8,50 reais, a capacidade máxima é equivalente a cerca de 82 itens.

4.2.4 *Demanda*

A demanda de cada cliente é definida pela ordem de pedido realizada pelo mesmo na plataforma *online*. Esses valores são extraídos de um banco de dados da *startup*.

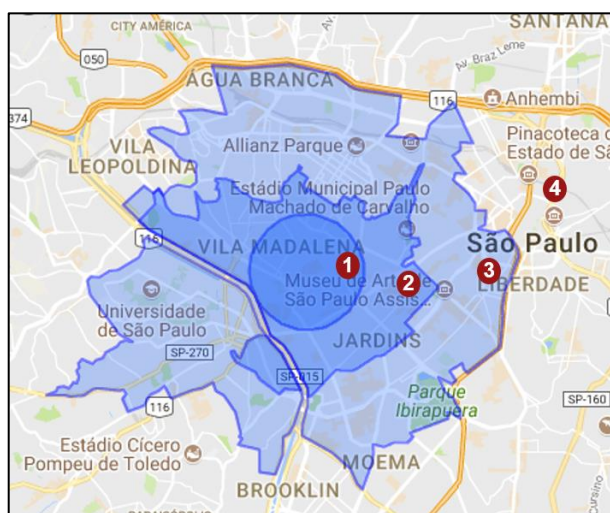
Atualmente, o ticket médio encontra-se em torno de R\$220, representando cerca de 24 itens por cliente.

4.2.5 Matriz-custo

A matriz-custo do problema informa o custo de transporte entre todos os pares de pontos de entrega da malha de clientes a ser considerada. Como os pontos de entrega não são fixos, isso implica que a matriz-custo também não é fixa, devendo ser atualizada todos os dias.

A função custo do problema é fornecida pela empresa terceirizada de entregas. Diferentemente do que é comumente visto em problemas de roteirização, o custo de transporte neste caso não é sempre proporcional à distância. Assim, a empresa de entregas usa um sistema de zoneamento da região em torno do depósito central, como ilustra a Figura 7.

Figura 7 - Mapa ilustrativo das 4 zonas de entrega da empresa terceirizada



Fonte: Elaborado pelo autor

Desta forma, a região é dividida em quatro zonas e para entregas dentro das três primeiras zonas, o custo é tabelado conforme a Tabela 3:

Tabela 3 - Custos de entrega por zona de chegada

	Zona 1	Zona 2	Zona 3
Primeira entrega	R\$11,50	R\$12,90	R\$15,10
Demais entregas	R\$8,90	R\$10,90	R\$14,50

Fonte: Elaborado pelo autor

É importante ressaltar que o custo de transporte independe da zona de saída do veículo, dependendo somente da zona de chegada. Assim, as colunas da tabela acima correspondem à zona de chegada do veículo. Caso o veículo saia do depósito, considera-se como primeira entrega. Caso contrário, usa-se os valores das demais entregas.

Exemplificando, se o veículo sai do armazém (primeira entrega) em direção a um cliente situado na zona 3, esse trajeto custará R\$15,10. Caso o veículo saia de qualquer cliente em direção a um cliente situado na zona 1, o custo do trajeto será de R\$8,90, independente da zona de saída ser a 1, 2, 3 ou fora das zonas.

Caso a zona de chegada do cliente situe-se fora das 3 zonas, o custo é proporcional à distância de acordo com a seguinte expressão:

$$\text{Custo (em R\$)} = 4 + 2 \times \text{Distância (em Km)}$$

Na maioria dos casos, não é necessário o retorno do veículo ao depósito, não havendo portanto custo associado a esse deslocamento. No entanto, em algumas entregas o cliente pode solicitar pagamento através de máquina de vale-refeição. Isso implica no retorno do veículo ao depósito para devolução da máquina e na cobrança de uma taxa de retorno, como pode-se observar na Tabela 4:

Tabela 4 - Taxas de retorno por zona de saída

Zona 1	Zona 2	Zona 3	Fora das zonas
R\$3,90	R\$6,90	R\$10,90	R\$12,80

Fonte: Elaborado pelo autor

Vale ressaltar que, ao contrário das taxas de entrega da Tabela 3, as taxas de retorno dependem da zona de saída do veículo e não de chegada, pois o destino é sempre o depósito central. Assim, a zona que o último cliente da rota se situa vai definir a taxa de retorno aplicada, no caso de uma rota com transporte da máquina de cartão.

Assim, o problema de obtenção da matriz-custo pode ser dividido em três partes: custo para zonas de chegada 1, 2 ou 3, custo para zonas de chegada situadas fora das 3 zonas e taxas de retorno.

Custo para zonas de chegada 1, 2 ou 3

Dado a zona onde cada cliente se situa, a matriz custo é facilmente obtida através da tabela de custos da empresa terceirizada de entregas. Assim, considerando uma matriz $n+2 \times n+2$, sendo n o número de pontos de entrega, sendo as linhas os pontos de saída e as colunas os pontos de chegada, o custo de cada célula da matriz corresponde ao custo tabelado entre o ponto de saída e de chegada correspondente.

Exemplo:

Para $n=3$, temos a seguinte matriz-custo:

Tabela 5 - Exemplo de matriz-custo 4x4

Saída	Chegada				
	0	C_{01}	C_{02}	C_{03}	C_{04}
	C_{10}	0	C_{12}	C_{13}	C_{14}
	C_{20}	C_{21}	0	C_{23}	C_{24}
	C_{30}	C_{31}	C_{32}	0	C_{34}
	C_{40}	C_{41}	C_{42}	C_{43}	0

Fonte: Elaborado pelo autor

A primeira coluna representa os trajetos saindo de qualquer ponto e chegando no ponto 0 (depósito). Como a convenção adotada na seção 3.1 indica que o retorno ao nó 0 significa a ausência de taxa de retorno, essa coluna vale 0 para todas as células.

A primeira linha representa os trajetos saindo do armazém e chegando a qualquer ponto, logo representa todas as primeiras entregas. Assim, os custos de primeira entrega devem ser utilizados. O custo c_{02} , por exemplo, vale R\$11,50 se o cliente 2 estiver na zona 1, R\$12,90 se o cliente 2 estiver na zona 2 e R\$15,10 se o cliente 2 estiver na zona 3.

As demais linhas representam os demais trajetos e, analogamente à primeira linha, deve-se usar os custos tabelados de acordo com as zonas das colunas.

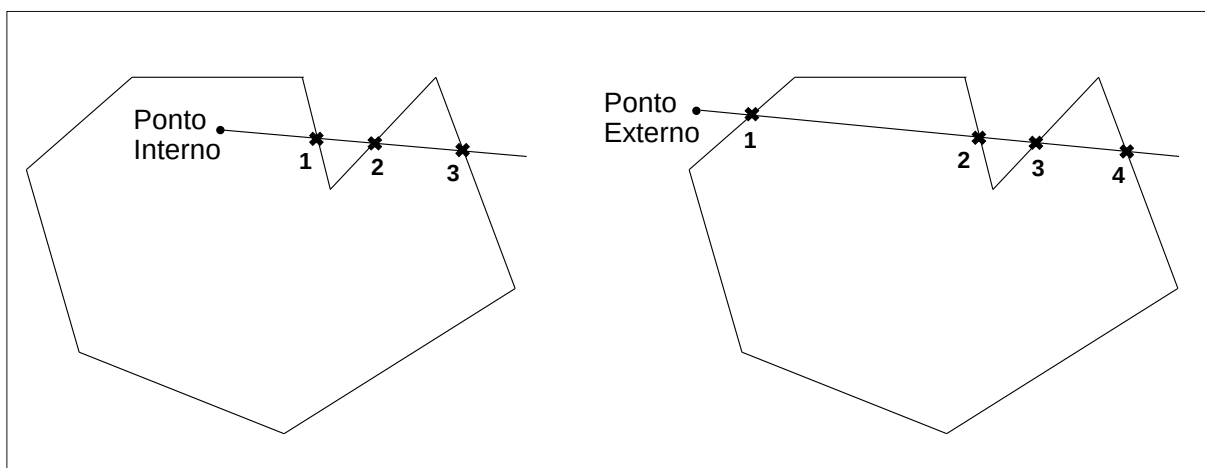
Neste exemplo, fica claro que sabendo a zona onde cada cliente se situa, obtém-se de forma simples os custos dos trajetos. Porém, a base de dados da *Liv Up* informa somente as coordenadas do endereço dos clientes, sendo necessário plotá-los num mapa para identificar

manualmente as zonas às quais eles pertencem. Para contornar esse problema, um algoritmo em linguagem VBA foi desenvolvido (ver Anexo A).

Esse algoritmo é uma função que, dadas as coordenadas do cliente e as coordenadas dos vértices da fronteira das zonas, retorna o número da zona à qual o cliente pertence, sendo 4 se o cliente se situar fora das 3 zonas.

O algoritmo em questão é derivado do algoritmo de teste interior/exterior de um ponto em relação a um polígono (Figueiredo; Carvalho, 1991). Traçando-se uma semireta em qualquer direção partindo-se do ponto, se a semireta cruzar a fronteira do polígono um número ímpar de vezes, o ponto é interno ao polígono. Caso contrário, o ponto é externo ao polígono, como ilustrado na Figura 8.

Figura 8 - Exemplo de pontos interno e externo a um polígono



Fonte: Elaborado pelo autor

As coordenadas dos clientes são obtidas através da base de dados da *Liv Up*. As coordenadas dos vértices das zonas são disponibilizadas pela empresa terceirizada de entregas em arquivos de extensão *.kml*.

Segue na Figura 9 o algoritmo desenvolvido:

Figura 9 - Descrição do algoritmo de verificação da zona

Programa Principal

Executar Rotina Ponto Interno Externo nas fronteiras da Zona 1, 2 e 3

Se o ponto é interno à fronteira da Zona 1 **então** o ponto pertence à Zona 1

Senão

Se o ponto é interno à fronteira da Zona 2 **então** o ponto pertence à Zona 2

Senão

Se o ponto é interno à fronteira da Zona 3 **então** o ponto pertence à Zona 3

Senão o ponto pertence à Zona 4

Fim se

Fim se

Fim se

Início Rotina Ponto Interno Externo

Enquanto houver uma aresta do polígono não verificada **faça**

Executar Rotina Intersecção Ponto Segmento

Se ponto está contido na aresta **então**

Fim enquanto

Senão

Incrementar contagem de intersecções

Fim se

Fim enquanto

Caso

O ponto está contido em alguma aresta: o ponto é interno ao polígono

O número de intersecções é ímpar: o ponto é interno ao polígono

O número de intersecções é par: o ponto é externo ao polígono

Fim caso

Fim Rotina Ponto Interno Externo

Início Rotina Intersecção Ponto Segmento

Traçar uma semirreta partindo do ponto, no sentido leste

Verificar se a semirreta cruza o segmento de reta ou se o ponto está contido no segmento

Fim Rotina Intersecção Ponto Segmento

Fonte: Elaborado pelo autor

Custo para zonas de chegada situadas fora das 3 zonas

Os custos para zonas de chegada fora das 3 zonas são diretamente proporcionais à distância. Desta forma, deve-se calcular primeiramente as distâncias saindo de todos os pontos (0 a n) e chegando a todos os pontos situados fora da zona.

Para esse cálculo, uma interface de programação (*API*¹, em inglês) da *Google* foi utilizada. A linha de código desse *API* pode ser consultada no Anexo B.

Uma vez obtidas todas as distâncias, aplica-se a fórmula mostrada anteriormente para chegar ao custo do trajeto.

Taxas de retorno

As taxas de retorno só devem ser aplicadas nas rotas em que algum cliente solicitou a máquina de cartão. Porém, no momento do cálculo da matriz-custo, não se sabe ainda as rotas da solução do problema. Deste modo, a matriz-custo gerada considera os custos de retorno na coluna correspondente, sendo que as restrições do problema definirão se a taxa de retorno será ou não somada à função objetivo.

Dada a zona onde cada cliente se situa, é possível completar a matriz-custo através da tabela de taxas de retorno da empresa terceirizada de entregas. Assim, considerando na Tabela 6 o mesmo exemplo anterior, temos:

Tabela 6 - Exemplo análogo à Tabela 5

0	C_{01}	C_{02}	C_{03}	C_{04}
C_{10}	0	C_{12}	C_{13}	C_{14}
C_{20}	C_{21}	0	C_{23}	C_{24}
C_{30}	C_{31}	C_{32}	0	C_{34}
C_{40}	C_{41}	C_{42}	C_{43}	0

Fonte: Elaborado pelo autor

¹ API é um conjunto de instruções, rotinas e padrões de programação usadas para que se possa acessar um aplicativo baseado na internet.

A última coluna representa os trajetos saindo de qualquer ponto e chegando no ponto 4 ($n+1$), equivalente ao depósito. Neste exemplo, os custos c_{14} , c_{24} e c_{34} correspondem à taxa de retorno e o custo c_{04} vale 0. Assim, caso o cliente 2 situe-se na zona 3, o custo c_{24} vale R\$10,90.

Tendo em vista que atualmente o número de clientes por dia é cerca de 90, é preciso calcular 8372 custos (92 elevado ao quadrado subtraído de 92 valores da diagonal principal). Com o crescimento deste número de entregas, a matriz se torna ainda maior, sendo necessário um processo automatizado de geração da matriz. Atendendo às necessidades da *Liv Up*, foi desenvolvida uma planilha em Excel suficiente para este objetivo.

4.2.6 Matriz-tempo

A matriz-tempo informa o tempo do trajeto entre todos os pares de pontos da malha. Ele inclui o tempo de deslocamento de um ponto ao outro assim como o tempo de serviço no ponto de saída.

Para o cálculo do tempo, três informações são necessárias: distância entre os pontos, velocidade média do veículo e tempo de serviço.

Distância entre os pontos

O problema da distância entre os pontos já foi abordado na subseção 4.2.5. No entanto, a interface da *Google* usada para o cálculo tem um tempo de processamento alto, o que inviabiliza seu uso para todos os pares de pontos (no item anterior, somente as distâncias para pontos de chegada na zona 4 são calculados).

Como as janelas de tempo dos clientes têm uma amplitude elevada, não é necessária uma alta precisão nos valores de tempo. Desse modo, pode-se utilizar uma distância aproximada, calculada através da multiplicação da distância linear por um fator de correção. Abaixo encontra-se a fórmula da distância linear entre dois pontos desprezando-se a curvatura da Terra:

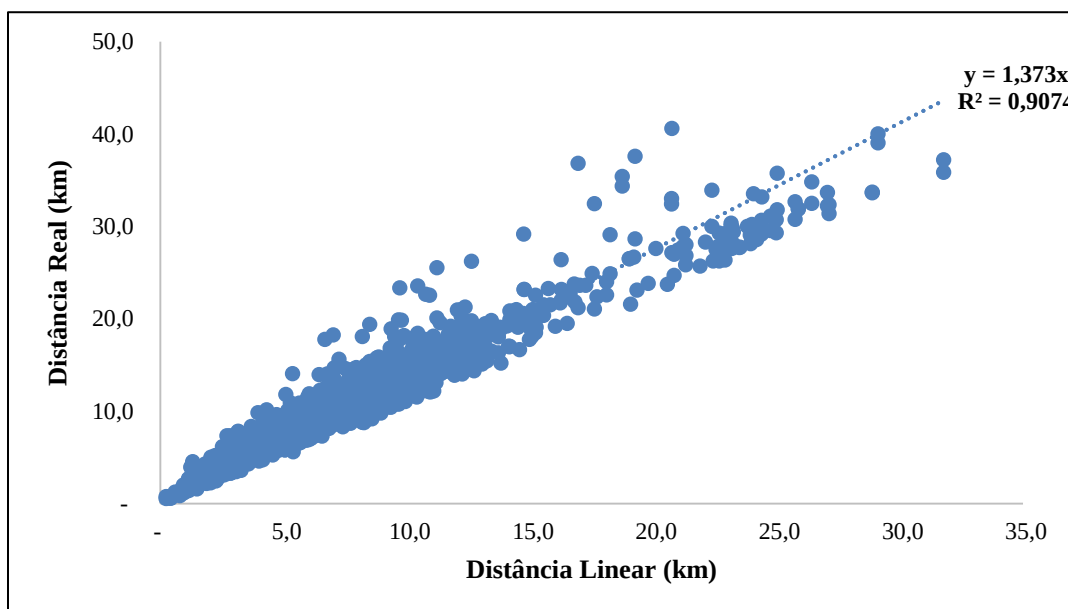
$$DL = f \sqrt{(X1 - X2)^2 + (Y1 - Y2)^2}, \text{ sendo:} \quad (4.12)$$

DL = distância linear, f = fator de conversão de graus para km = 111,12 km/grau

(Xn, Yn) = coordenada geográfica (em graus) do ponto

O fator de correção foi calculado através da correlação entre as distâncias reais e distâncias lineares, para cerca de 800 pares de pontos correspondentes a clientes da *Liv Up*. Segue abaixo o gráfico obtido:

Gráfico 3 - Regressão linear da distância real vs distância linear



Fonte: Elaborado pelo autor

Para distâncias reais de até 25km, o que corresponde à maior parte dos casos, é razoável aplicar um fator de 1,37 na distância linear. Além disso, a regressão linear apresenta alto coeficiente de determinação R^2 , o que demonstra alta aderência do modelo linear à realidade.

Velocidade média do veículo

Como não há informações suficientes para o cálculo da velocidade média dos veículos, assume-se um valor de 25km/h como premissa do modelo.

Tempo de serviço

O tempo de serviço também é uma premissa do modelo, com o valor de 0,1h.

Através da multiplicação da distância linear corrigida pela velocidade média dos veículos, somado ao tempo de serviço, chega-se ao tempo do trajeto entre dois pontos.

4.2.7 Janelas de entrega

As janelas de entrega representam um intervalo de tempo dentro do qual o cliente deve receber o produto. Como o produto se mantém congelado fora de refrigeração somente até 6 horas, alguns clientes optam por essa restrição para conseguirem receber o produto ainda congelado.

Os valores do intervalo são fornecidos pelo cliente no momento do pedido e podem ser extraídos do banco de dados da *Liv Up*. Atualmente, cerca de 10% dos clientes utilizam essa restrição, e o tamanho do intervalo é em torno de 1h30min.

4.2.8 Solicitações de máquina de vale-refeição

Como explicado anteriormente, é preciso saber se um cliente solicitou máquina de vale-refeição pois isso implica na cobrança de taxa de retorno pela empresa de transporte.

Essa informação é fornecida pelo cliente no momento da compra online e pode ser extraída da base de dados da *startup*. A partir dessa informação, deve-se convertê-la em um vetor binário que vale 1 caso o cliente solicitou a máquina e 0, caso contrário.

Atualmente, cerca de 40% dos clientes solicitam a máquina de vale-refeição como forma de pagamento.

4.3 Resultados do Modelo de Programação Linear Inteira Mista

Nesta seção, o modelo de Programação Linear Inteira Mista do Problema de Roteirização de Veículos com Janelas de Tempo será testado através de métodos computacionais que fornecem a solução ótima.

O computador utilizado possui processador Intel® Core™ i7, 2.0GHz, com 8GB de memória RAM e sistema operacional Windows 10 (64 bits).

A plataforma escolhida foi o *software* ILOG CPLEX², versão 12.7.1, com configurações no modo padrão. O CPLEX possui interface com diversas linguagens de programação, como C, C++, *Python*, *Matlab* e *OPL*. Devido à proximidade da linguagem com a formulação

² Software de otimização desenvolvido pela IBM.

matemática, a *OPL (Optimization Programming Language)* foi escolhida para a programação. As linhas de código podem ser consultadas no Anexo D.

Conforme mencionado no início do capítulo, o objetivo dos experimentos computacionais é o teste da hipótese de inviabilidade do uso do método exato na obtenção das soluções do problema. Para tanto, testes incrementais foram realizados, partindo de uma malha de 1 cliente e adicionando 1 cliente à malha a cada teste. O custo total, tempo de processamento e roteiro obtido foram registrados para posterior análise dos resultados.

As informações fornecidas como parâmetros do modelo correspondem a dados reais de um dia de entregas e seguem nas Tabela 7 a 9:

Tabela 7 - Instâncias do modelo			
Id Cliente	Demanda (R\$)	Janela de entregas³ (h)	Vale-refeição⁴
1	115	0-3	0
2	346	0-3	0
3	159	0-3	0
4	72	0-3	1
5	274	0-3	0
6	167	0-3	0
7	406	0-3	0
8	162	0-3	0
9	241	0-3	1
10	132	0-3	0
11	77	0-3	1
12	104	0-3	1
13	213	2-3	0
14	151	0-3	1

³ O instante 0 corresponde à partida do veículo. 0-3 significa que o veículo pode chegar no cliente entre o instante 0 e 3h após a partida do veículo.

⁴ O valor 1 significa que o cliente solicitou a máquina de vale-refeição. O valor 0 corresponde ao caso contrário.

15	152	0-3	1
16	132	0-3	0

Fonte: Elaborado pelo autor

Tabela 8 - Matriz-custo (em R\$)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	-	15,1	12,9	22,2	25,6	25,0	15,1	23,8	15,1	44,8	15,1	15,1	15,1	15,1	30,5	34,1	25,1	-
1	-	-	10,9	21,6	30,6	14,6	14,5	23,2	14,5	50,3	14,5	14,5	14,5	14,5	27,6	29,4	22,0	10,9
2	-	14,5	-	24,2	30,4	29,8	14,5	25,8	14,5	49,5	14,5	14,5	14,5	14,5	30,2	32,0	21,7	6,9
3	-	14,5	10,9	-	18,1	10,6	14,5	13,4	14,5	36,0	14,5	14,5	14,5	14,5	20,6	19,9	15,4	12,8
4	-	14,5	10,9	18,8	-	30,7	14,5	14,1	14,5	31,9	14,5	14,5	14,5	14,5	9,3	16,6	9,9	12,8
5	-	14,5	10,9	12,2	21,5	-	14,5	21,1	14,5	41,0	14,5	14,5	14,5	14,5	25,5	25,2	18,8	12,8
6	-	14,5	10,9	16,9	12,1	21,5	-	13,2	14,5	38,0	14,5	14,5	14,5	14,5	17,3	19,0	8,0	10,9
7	-	14,5	10,9	12,9	15,6	18,2	14,5	-	14,5	31,6	14,5	14,5	14,5	14,5	16,2	11,5	14,2	12,8
8	-	14,5	10,9	20,8	29,9	11,6	14,5	22,4	-	42,3	14,5	14,5	14,5	14,5	26,9	28,6	21,2	10,9
9	-	14,5	10,9	37,8	26,1	47,8	14,5	31,3	14,5	-	14,5	14,5	14,5	14,5	22,8	25,2	33,5	12,8
10	-	14,5	10,9	21,6	30,7	14,7	14,5	23,2	14,5	50,4	-	14,5	14,5	14,5	27,7	29,4	22,0	10,9
11	-	14,5	10,9	19,7	14,6	24,4	14,5	21,3	14,5	38,8	14,5	-	14,5	14,5	25,8	23,0	10,9	10,9
12	-	14,5	10,9	18,4	11,4	23,0	14,5	20,0	14,5	37,3	14,5	14,5	-	14,5	16,7	19,8	8,2	10,9
13	-	14,5	10,9	13,6	25,6	11,1	14,5	18,1	14,5	38,0	14,5	14,5	14,5	-	22,6	24,3	14,5	10,9
14	-	14,5	10,9	19,2	10,5	29,3	14,5	12,7	14,5	22,7	14,5	14,5	14,5	14,5	-	13,3	14,9	12,8
15	-	14,5	10,9	20,5	15,4	22,6	14,5	10,7	14,5	22,7	14,5	14,5	14,5	14,5	12,3	-	18,7	12,8
16	-	14,5	10,9	14,8	8,8	24,0	14,5	12,4	14,5	28,5	14,5	14,5	14,5	14,5	13,1	16,7	-	12,8
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Fonte: Elaborado pelo autor

Tabela 9 - Matriz-tempo (em horas)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	0,1	0,2	0,1	0,4	0,5	0,4	0,3	0,4	0,3	0,8	0,2	0,2	0,3	0,3	0,5	0,6	0,3	-
1	-	0,1	0,3	0,5	0,6	0,3	0,3	0,5	0,2	1,0	0,1	0,3	0,4	0,1	0,6	0,6	0,5	0,3
2	-	0,3	0,1	0,5	0,7	0,6	0,4	0,5	0,3	1,0	0,3	0,4	0,4	0,3	0,6	0,7	0,5	0,2
3	-	0,3	0,5	0,1	0,4	0,2	0,4	0,3	0,4	0,7	0,3	0,4	0,4	0,3	0,4	0,4	0,3	0,5
4	-	0,5	0,6	0,4	0,1	0,6	0,3	0,3	0,6	0,7	0,5	0,3	0,3	0,5	0,2	0,4	0,2	0,5
5	-	0,3	0,5	0,3	0,4	0,1	0,4	0,4	0,2	0,8	0,3	0,5	0,5	0,3	0,5	0,5	0,4	0,5
6	-	0,3	0,4	0,4	0,3	0,4	0,1	0,3	0,4	0,8	0,3	0,2	0,1	0,3	0,4	0,4	0,2	0,4
7	-	0,5	0,6	0,3	0,3	0,4	0,3	0,1	0,5	0,6	0,5	0,4	0,3	0,4	0,3	0,2	0,3	0,6
8	-	0,2	0,3	0,4	0,6	0,3	0,4	0,5	0,1	0,9	0,2	0,4	0,4	0,2	0,6	0,6	0,4	0,4
9	-	0,9	1,1	0,8	0,5	1,0	0,7	0,6	1,0	0,1	0,9	0,8	0,7	0,9	0,5	0,5	0,8	0,9
10	-	0,1	0,3	0,5	0,6	0,3	0,3	0,5	0,2	1,0	0,1	0,3	0,4	0,1	0,6	0,6	0,5	0,3
11	-	0,3	0,4	0,4	0,3	0,5	0,2	0,4	0,4	0,8	0,3	0,1	0,2	0,3	0,5	0,5	0,2	0,3
12	-	0,4	0,4	0,4	0,2	0,5	0,1	0,4	0,4	0,8	0,4	0,2	0,1	0,3	0,4	0,4	0,2	0,4
13	-	0,1	0,3	0,3	0,5	0,3	0,3	0,4	0,2	0,8	0,1	0,3	0,3	0,1	0,5	0,5	0,3	0,4
14	-	0,6	0,7	0,4	0,2	0,6	0,3	0,3	0,6	0,5	0,6	0,4	0,3	0,5	0,1	0,3	0,3	0,6
15	-	0,6	0,8	0,4	0,3	0,5	0,5	0,2	0,7	0,6	0,6	0,6	0,5	0,6	0,3	0,1	0,4	0,7
16	-	0,4	0,5	0,3	0,2	0,5	0,2	0,3	0,5	0,6	0,4	0,3	0,2	0,4	0,3	0,4	0,1	0,4
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0,1

Fonte: Elaborado pelo autor

Os clientes foram adicionados um a um na malha na mesma ordem mostrada na Tabela 7. Os resultados obtidos podem ser observados na Tabela 10:

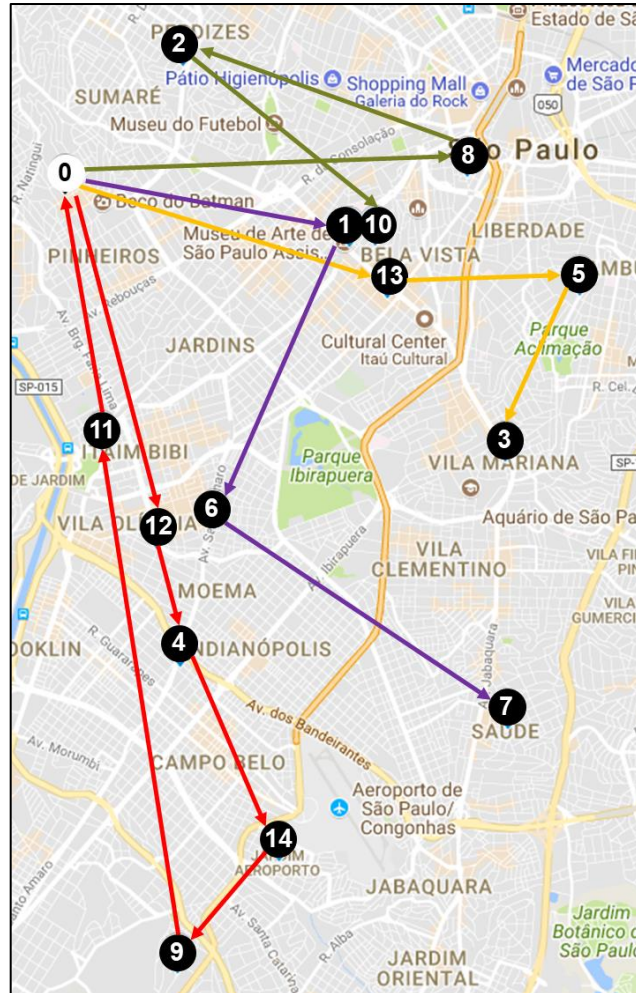
Tabela 10 - Soluções geradas pelo software CPLEX (D = depósito central)

# clientes	Rotas geradas	Custo (R\$)	Tempo (s)
1	D-1	15,10	0,09
2	D-1-2	26,00	0,10
3	D-1-3-2	47,57	0,12
4	D-1-3-4-2-D	72,58	0,15
5	D-1-5-3-D; D-4-2	85,35	0,23
6	D-6-4-2-D; D-1-5-3	86,94	0,26
7	D-6-4-7-D; D-2; D-1-5-3	108,95	0,39
8	D-1-2; D-6-4-7-D; D-8-5-3	119,03	0,62
9	D-8-5-3; D-6-7-1; D-4-9-2-D	157,08	0,81
10	D-4-9-2-D; D-8-5-3; D-10; D-1-6-7	172,18	1,34
11	D-1-2; D-8-5-3; D-11-4-9-10-12-D; D-6-7	180,85	4,62
12	D-10-2; D-8-5-3; D-11-12-4-9-1-D; D-6-7	191,51	19,86
13	D-10-8-2; D-13-5-3; D-1-12-4-9-11-D; D-6-7	205,49	60,99
14	D-12-4-14-9-11-D; D-8-2-10; D-13-5-3; D-1-6-7	205,52	57,23
15	D-11-4-14-15-9-D; D-8-1-10; D-13-5-3; D-6-7; D-12-2-D	231,45	965,37
16	-	-	> 12600

Fonte: Elaborado pelo autor

Uma visualização gráfica dos resultados para o caso de 14 clientes pode ser observada na Figura 10:

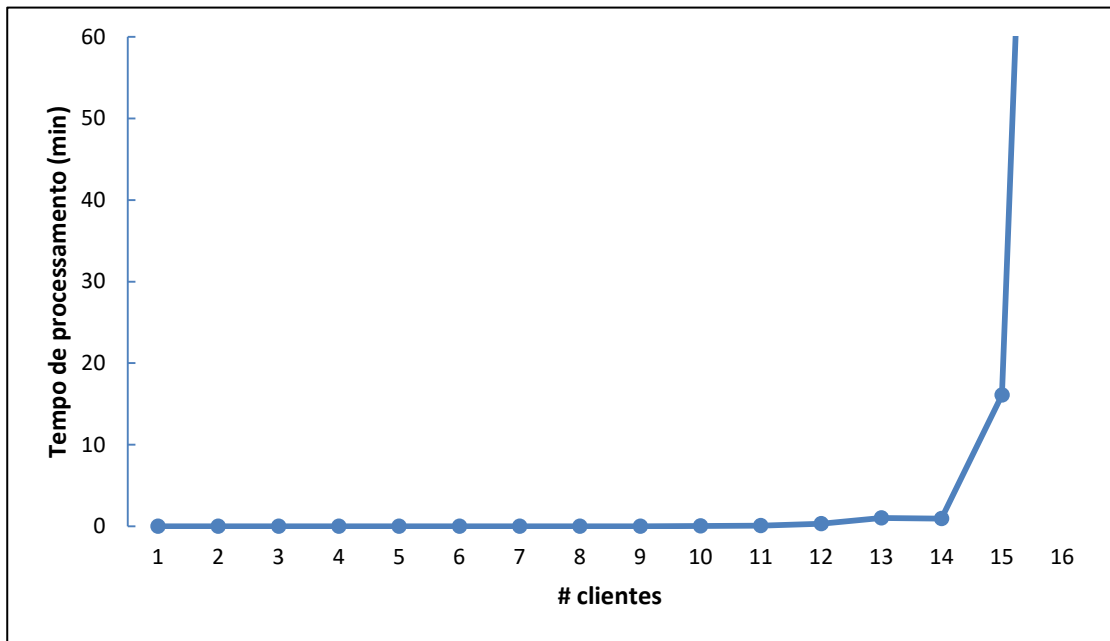
Figura 10 - Exemplo de rotas geradas pelo método exato para 14 clientes



Fonte: Elaborado pelo autor

4.4 Análise dos Resultados

Com o objetivo de testar a hipótese inicial de inviabilidade da solução exata do problema, foram realizados testes incrementais do modelo. Começando com uma única entrega e adicionando um cliente a cada teste, obteve-se um gráfico que retrata o tempo de processamento em função do número de clientes.

Gráfico 4 - Evolução do tempo de processamento em função do número de clientes

Fonte: Elaborado pelo autor

Observa-se que, até 14 pontos de entrega, a solução exata possui um tempo de processamento próximo de 0. Com 15 pontos de entrega, o tempo de processamento passa à ordem dos minutos. Com 16 pontos de entrega, o tempo de processamento aumenta consideravelmente, sendo superior a 3,5h.

Como o processo manual de roteirização leva atualmente 1 hora para ser feito, a solução exata só é viável do ponto de vista de tempo até 15 clientes. No entanto, o número de clientes a serem roteirizados atualmente é cerca de 45 por período de entrega, com perspectivas de crescimento a cada mês. Assim, a hipótese de inviabilidade do uso do método exato de resolução mostra-se verdadeira, sendo necessário o uso de métodos heurísticos para a resolução do problema num intervalo de tempo viável.

5 MÉTODO HEURÍSTICO DE RESOLUÇÃO

Conforme visto no capítulo anterior, a resolução do problema através de método exato não se mostrou viável do ponto de vista de tempo computacional. Isto pode ser explicado devido à alta complexidade computacional do Problema de Roteirização de Veículos, incluindo aqueles com janela de tempo, o que foi explicado no capítulo 3. Desta forma, a estratégia de solução que adota-se nesses casos é o uso de métodos heurísticos e, se necessário, meta-heurísticas de melhoria do método inicial.

Alguns dos principais métodos heurísticos de resolução do PRVJT foram abordados na seção 3.4. Observou-se que a principal obra neste tema é a de Solomon (1987), que desenvolveu 7 heurísticas e as testou em 56 problemas, concluindo que a heurística de inserção I1 apresentou os melhores resultados. A partir deste trabalho, diversos autores desenvolveram adaptações da heurística I1 ou utilizaram os 56 problemas como base de comparação para novas heurísticas desenvolvidas para o problema.

No caso deste trabalho, também optou-se por adaptar a heurística I1 de Solomon para o caso específico do problema. A razão pela qual heurísticas mais recentes ou meta-heurísticas não foram utilizadas deve-se ao fato de que a heurística de Solomon é rápida e possui poucos parâmetros, não acrescentando complexidade à resolução. Assim, acredita-se que o uso de métodos mais complexos não se justifica para o cenário atual da *startup*, sendo a heurística I1 suficiente para o objetivo do trabalho.

Desta forma, o presente capítulo organiza-se em duas partes. Primeiramente, as adaptações feitas à heurística I1 são explicadas e o algoritmo desenvolvido é descrito. Em seguida, a eficácia deste algoritmo é testada através da comparação com a solução ótima para um conjunto de problemas.

5.1 Adaptação do algoritmo de inserção I1 de Solomon

A heurística de inserção sequencial I1 de Solomon (descrita na seção 3.4), como o próprio nome já diz, é um algoritmo que cria rotas através da inserção de clientes, um em seguida do outro, de acordo com critérios de inicialização e de inserção.

Os critérios de inicialização são exclusivos (apenas um é usado) e podem ser o cliente mais distante ainda não alocado ou o cliente com o menor valor de fim da janela de tempo.

Existem 2 critérios para inserção, c_1 e c_2 . O critério c_1 determina a melhor posição de inserção na rota para cada cliente não alocado. Este critério é composto por 2 subcritérios, c_{11} e c_{12} , que avaliam, respectivamente, o impacto da inserção na distância e no tempo. O critério c_2 determina o cliente que maximiza o benefício da inserção.

No entanto, o problema deste trabalho difere-se dos problemas tratados por Solomon (1987) quanto a dois aspectos: não é necessário minimizar a distância total e só existe custo de retorno quando um veículo transporta máquina de cartão. Assim, algumas adaptações ao método proposto por Solomon (1987) devem ser feitas.

Três adaptações foram desenvolvidas no algoritmo, referentes às distâncias, ao retorno e ao critério de inicialização.

A adaptação referente às distâncias diz respeito ao subcritério c_{11} e ao critério c_2 . No algoritmo original, c_{11} representa o acréscimo da inserção do cliente u na distância da rota, como observa-se em (5.1). Já o critério c_2 representa o benefício causado pela inserção do cliente u , tomando como base a distância do depósito ao cliente u , como é visto em (5.2).

$$c_{11}(i, u, j) = d_{iu} + d_{uj} - \mu d_{ij} \quad \mu \geq 0 \quad (5.1)$$

$$c_2(i, u, j) = \lambda d_{0u} - c_1(i, u, j) \quad \lambda \geq 0 \quad (5.2)$$

No entanto, no problema atual, não deseja-se minimizar a distância total, e sim o custo total. Como o custo não é sempre proporcional à distância, não faz sentido o uso de distâncias no algoritmo. Assim, as distâncias foram substituídas pelos custos, como pode-se observar em (5.3) e (5.4).

$$c_{11}(i, u, j) = c_{iu} + c_{uj} - \mu c_{ij} \quad \mu \geq 0 \quad (5.3)$$

$$c_2(i, u, j) = \lambda c_{0u} - c_1(i, u, j) \quad \lambda \geq 0 \quad (5.2)$$

A segunda adaptação refere-se às especificidades do retorno ao depósito. O subcritério c_{11} representa o acréscimo de custo pela inserção do cliente u na rota. Porém, se um cliente que

solicitou máquina de cartão é inserido numa rota em que ainda não tenha nenhum cliente que solicitou a máquina, além do acréscimo de custo causado pela posição de inserção, um custo de retorno deve ser adicionado à rota. Dessa forma, o termo c_{1l} foi adaptado, como pode-se observar em (5.4).

$$c_{1l}(i, u, j, k) = c_{iu} + c_{uj} - \mu c_{ij} + v_u (1 - r_k) c_{l, N+1} \quad \mu \geq 0 \quad (5.4)$$

Onde:

$v_u = 1$ se o cliente u solicitou máquina de cartão e 0, caso contrário.

$r_k = 1$ se o veículo da rota k em construção deve retornar ao depósito e 0, caso contrário.

l = último cliente da rota em construção.

Por fim, uma última adaptação foi feita quanto ao critério de inicialização. Assim, considera-se tanto a distância do cliente ao depósito quanto o fim de sua janela de tempo. Primeiro, todos os clientes são ordenados da maior distância para a menor. Em seguida, ordena-se pelo menor fim da janela de tempo. Desta forma, os clientes que devem ser atendidos antes são iniciados nas rotas e, caso possuam o mesmo fim da janela de tempo, aquele de maior distância ao depósito é priorizado.

Dadas as adaptações descritas acima, foi desenvolvido um algoritmo em linguagem VBA (ver anexo E) cujos procedimentos podem ser vistos na Figura 11:

Figura 11 - Descrição do algoritmo adaptado da heurística de inserção I1 de Solomon

Programa principal
Executar Rotina <u>Vetor_inicialização</u>
Enquanto existe cliente ainda não alocado faça
Se nova rota deve ser criada então
Atualizar número de veículos
Inicializar número de clientes da rota, posição inicial, instante inicial e capacidade utilizada
Adicionar primeiro cliente u do vetor de inicialização na posição 1
Atualizar instantes de chegada da rota
Inicializar vetor de clientes factíveis com vetor de inicialização

Senão**Executar** Rotina Inserção_cliente**Fim se**Verificar se após inserção de cliente u o veículo deve retornar ao depósito

Atualizar capacidade utilizada do veículo

Atualizar número de clientes da rota

Remover cliente u de vetor de inicialização e de vetor de clientes factíveis**Fim enquanto**

Calcular custo total

Início Rotina Vetor_inicialização

Ordenar clientes por ordem decrescente de distância em relação ao depósito

Ordenar, em seguida, por ordem crescente do fim da janela de tempo

Fim Rotina Vetor_inicialização**Início** Rotina Inserção_cliente**Se** não existe cliente factível **então**

Nova rota deve ser criada

Senão

Retirar do vetor de clientes factíveis aqueles infactíveis em relação à capacidade

Se não houver cliente factível **então**

Nova rota deve ser criada

Senão**Para** cada cliente u factível **faça****Para** cada posição p da rota **faça****Executar** Rotina Critério_c1Atualizar valor mínimo de c_1 **Fim para****Fim para****Se** não houver inserção factível **então**

Nova rota deve ser criada

Senão**Para** cada cliente factível **faça**Calcular o termo c_2

Atualizar valor máximo de c_2

Fim para

Atualizar posições da rota após inserção de cliente u que maximiza c_2

Atualizar instantes da rota

Fim se

Fim se

Fim se

Fim Rotina Inserção_cliente

Início Rotina Critério_c1

Calcular novos instantes de chegada após inserção de cliente u na posição p

Para cada cliente localizado após o cliente u (u inclusive), **faça**

Verificar se restrição de janela de tempo ainda é atendida

Fim para

Se alguma restrição de janela de tempo não for atendida **então**

Retornar que inserção de cliente u na posição p é infactível

Senão

Se o veículo não precise retornar ao depósito e o cliente u solicitou máquina de vale-refeição e posição $p \neq n+1$ **então**

Calcular termo c_{11} e adicionar taxa de retorno do último cliente ao termo

Senão

Calcular termo c_{11}

Fim se

Calcular termo c_{12}

Retornar critério $c_1 = \alpha_1 c_{11} + \alpha_2 c_{12}$

Fim se

Fim Rotina Critério_c1

Fonte: Elaborado pelo autor

5.2 Eficácia do algoritmo desenvolvido

Conforme explicado no capítulo 4, apesar de o método exato de resolução não ser viável, seu desenvolvimento foi útil para servir como base de comparação e avaliação inicial da heurística.

Desta forma, esta seção tem como objetivo avaliar a eficácia do algoritmo desenvolvido através da comparação do custo total obtido com o custo total ótimo.

Vale ressaltar que, como o número máximo viável de clientes a serem roteirizados no método exato é 15, não será possível avaliar a eficácia da heurística em cenários reais de mais de 50 clientes. No entanto, os resultados deste capítulo fornecem uma visão inicial da eficácia do algoritmo desenvolvido, mesmo que para situações de menor escala.

Por motivos de tempo de processamento, decidiu-se por efetuar testes com 11 clientes. Para tanto, 3 conjuntos de problemas foram testados: conjunto A, B e C. Cada conjunto possui 15 problemas diferentes. As instâncias dos problemas do conjunto A podem ser consultadas no Anexo F.

O conjunto A de problemas foi obtido através de amostras de 11 clientes retiradas aleatoriamente da base histórica real de roteiros. Desta forma, este conjunto representa o grupo de controle. Neste conjunto, cerca de 30% dos clientes solicitam máquina de cartão e cerca de 10% dos clientes possuem restrições de janela de tempo de amplitude entre 1 e 2 horas.

O conjunto B de problemas é idêntico ao conjunto A em todas as características exceto nas solicitações de máquina de cartão. Neste conjunto, considera-se que nenhum cliente solicitou a máquina.

O conjunto C de problemas é idêntico ao conjunto A em todas as características exceto nas restrições de janela de tempo. Neste conjunto, considera-se que 50% dos clientes possuem restrições de janela de tempo (de mesma amplitude de A).

As características de cada conjunto são resumidas na Tabela 11:

Tabela 11 - Características dos conjuntos de problemas A, B e C

Conjunto	# clientes	Solicitações de máquina	Restrição de janela de tempo
A	11	30%	10%
B	11	0%	10%
C	11	30%	50%

Fonte: Elaborado pelo autor

Assim, nas subseções seguintes, pretende-se avaliar a eficácia do algoritmo e o impacto da mudança de cenários. Para avaliar a eficácia, o conjunto A de problemas será testado e o parâmetro α do algoritmo será analisado. Para avaliar os impactos da mudança de cenários, os conjuntos B e C serão testados e comparados com A.

5.2.1 Análise da eficácia para o grupo de controle com parâmetro α fixo

Primeiramente, a eficácia do algoritmo será avaliada para os problemas do conjunto A, que representam o grupo de controle das análises e possuem as características reais do problema. Para tanto, foi calculado o desvio percentual do custo total da heurística em relação ao ótimo. O custo total ótimo para os 15 problemas foi obtido da mesma maneira que no capítulo 4.

Conforme visto anteriormente, o critério de inserção c_1 da heurística I1 é uma média ponderada dos subcritérios c_{11} e c_{12} . Esta ponderação é feita pelos parâmetros α_1 e α_2 , que podem ser sintetizados em um único parâmetro α tal que $\alpha_1 = \alpha$ e $\alpha_2 = 1 - \alpha$. Nesta subseção, o parâmetro α utilizado para todos os problemas é o mesmo: $\alpha = 0,5$.

Segue na Tabela 12 os resultados obtidos:

Tabela 12 - Análise da eficácia da heurística para o conjunto A e $\alpha = 0,5$

Problema	Custo Total Ótimo (R\$)	Custo Total Obtido (R\$)	Desvio
A1	223,20	312,31	39,9%
A2	204,40	214,13	4,8%
A3	173,44	211,95	22,2%

A4	210,74	232,04	10,1%
A5	261,91	263,17	1,0%
A6	235,90	238,38	1,1%
A7	247,11	283,97	14,9%
A8	260,91	281,45	7,9%
A9	220,92	227,98	3,2%
A10	222,19	247,59	11,4%
A11	155,31	186,11	19,8%
A12	194,49	206,01	5,9%
A13	198,88	219,27	10,3%
A14	216,14	220,99	2,2%
A15	203,28	207,55	3,5%
Média			10,6%

Fonte: Elaborado pelo autor

Os resultados acima indicam que, na média, o algoritmo retorna um custo em torno de 10% superior ao ótimo para 11 clientes.

No entanto, o parâmetro α pode ser alterado, podendo ou não resultar numa melhoria na eficácia.

5.2.2 Impacto do parâmetro α na eficácia

Na seção anterior, utilizou-se o parâmetro $\alpha = 0,5$. O objetivo desta subseção é verificar se a alteração do α afeta a eficácia do algoritmo.

Para tanto, o mesmo conjunto de problemas A será testado. Neste caso, o parâmetro α do algoritmo será variado de 0 a 1 em incrementos de 0,1. O α escolhido, em cada problema, será aquele que forneça o menor custo total entre os α testados, e será chamado de α^* .

Segue na Tabela 13 os resultados obtidos:

Tabela 13 - Impacto do parâmetro α na eficácia do algoritmo					
Problema	Custo Total Ótimo (R\$)	Custo Total Obtido (R\$)	Desvio α^*	Desvio $\alpha=0,5$	α^*
A1	223,20	225,24	0,9%	39,9%	1
A2	204,40	214,13	4,8%	4,8%	1
A3	173,44	189,05	9,0%	22,2%	1
A4	210,74	232,04	10,1%	10,1%	1
A5	261,91	263,17	1,0%	1,0%	1
A6	235,90	238,38	1,1%	1,1%	1
A7	247,11	279,05	12,9%	14,9%	1
A8	260,91	281,45	7,9%	7,9%	1
A9	220,92	227,05	2,8%	3,2%	1
A10	222,19	236,00	6,2%	11,4%	0,7
A11	155,31	172,96	11,4%	19,8%	1
A12	194,49	204,64	5,2%	5,9%	1
A13	198,88	210,85	6,0%	10,3%	1
A14	216,14	220,99	2,2%	2,2%	0,7
A15	203,28	207,55	3,5%	3,5%	1
Média			5,7%	10,6%	

Fonte: Elaborado pelo autor

Os resultados obtidos permitem indicar que houve uma melhoria significativa na eficácia do algoritmo através da variação do parâmetro α . De 10,6%, o desvio médio passou a 5,7%, uma melhoria de 4,9 pontos percentuais ou cerca de 50%.

Também nota-se que os parâmetros α^* encontram-se próximos de 1, o que significa que o termo c_{11} (acréscimo de custo) é mais relevante que c_{12} (acréscimo de tempo).

Vale ressaltar que, em muitos casos, mais de um valor de α fornecia o menor custo total, como observa-se na Tabela 14. Assim, o maior α entre eles foi escolhido.

Tabela 14 - Custo Total Obtido em função de α para o problema A14

α	Custo Total Obtido (R\$)
0	260,97
0,1	245,71
0,2	245,71
0,3	248,57
0,4	220,99
0,5	220,99
0,6	220,99
0,7	220,99
0,8	226,62
0,9	226,62
1	226,62

Fonte: Elaborado pelo autor

Em conclusão, foi possível reduzir o desvio do algoritmo através da alteração do parâmetro α . Uma possível explicação é o fato de que as características dos problemas, como percentual de janelas de tempo, amplitude das janelas e percentual de clientes que solicitam máquina definem o peso que se deve atribuir aos subcritérios c_{11} e c_{12} de forma a minimizar o custo total. Uma outra interpretação é que, como variando-se o α obtém-se rotas diferentes, isto é equivalente a realizar trocas de clientes entre rotas. Assim, variando-se o α , é possível obter um leque maior de soluções e escolher a melhor. Nesta última interpretação, faria sentido a relaxação da restrição $\alpha_1 + \alpha_2 = 1$.

5.2.3 Impacto do retorno do veículo na eficácia

Uma das adaptações do algoritmo original de Solomon diz respeito aos retornos dos veículos, como explicado anteriormente. Assim, é interessante avaliar se perde-se em eficácia em cenários com retorno de veículo.

Portanto, a subseção atual tem como objetivo avaliar se a presença de clientes que solicitam máquina de cartão impacta na eficiência do algoritmo.

Neste caso, o conjunto B (idêntico ao A, porém sem retorno de veículos) será testado e a eficiência calculada será comparada com o conjunto de problemas A. O α utilizado nos dois casos é o α^* . Segue na Tabela 15 os resultados:

Tabela 15 - Impacto do retorno do veículo na eficácia do algoritmo

Problema	Custo Total Ótimo (R\$)	Custo Total Obtido (R\$)	Desvio B	Desvio A	α^*
B1	212,11	216,40	2,0%	0,9%	1
B2	182,60	190,43	4,3%	4,8%	1
B3	165,46	175,44	6,0%	9,0%	1
B4	188,94	195,54	3,5%	10,1%	1
B5	238,21	239,47	0,5%	1,0%	1
B6	199,40	212,78	6,7%	1,1%	1
B7	227,75	232,27	2,0%	12,9%	0,7
B8	216,61	243,05	12,2%	7,9%	0,5
B9	190,55	190,55	0,0%	2,8%	1
B10	179,08	184,80	3,2%	6,2%	0,7
B11	151,41	151,41	0,0%	11,4%	0,9
B12	181,69	186,56	2,7%	5,2%	1
B13	178,45	182,77	2,4%	6,0%	0,6
B14	205,34	210,19	2,4%	2,2%	0,7
B15	159,50	164,15	2,9%	3,5%	1

Média 3,4% 5,7%

Fonte: Elaborado pelo autor

Dos resultados acima, observa-se que em 11 dos 15 problemas o desvio aumenta com a presença de clientes que solicitam máquina de cartão. Considerando a média dos desvios, tem-se um aumento de 2,3 pontos percentuais. No entanto, em valores absolutos, ambos os desvios encontram-se num patamar aceitável.

Desta forma, conclui-se que a presença de clientes que solicitam máquina de vale-refeição na roteirização reduz a eficácia da heurística desenvolvida. Apesar de o resultado não ser desejável, ele é importante para a identificação das limitações do algoritmo e oportunidades de melhoria.

5.2.4 Impacto das janelas de tempo na eficácia

Analogamente à subseção anterior, é interessante avaliar se o aumento da proporção de clientes com restrição de janelas de tempo impacta na eficácia do algoritmo, já que no futuro é possível que a *Liv Up* ofereça a possibilidade de escolha de uma janela de tempo a todos os clientes no próprio *website* (como visto no capítulo 2, esta escolha é feita atualmente pelo telefone).

Neste caso, o conjunto C de problemas (50% de restrições de janela de tempo) foi testado e comparado com o conjunto A (10% de restrições de janela de tempo). O parâmetro α^* foi utilizado nos dois casos. Segue na Tabela 16 resultados:

Tabela 16 - Impacto do aumento de restrições de janela de tempo na eficácia do algoritmo

Problema	Custo Total Ótimo (R\$)	Custo Total Obtido (R\$)	Desvio C	Desvio A	α^*
C1	223,20	280,12	25,5%	0,9%	1
C2	204,40	241,39	18,1%	4,8%	0,5
C3	176,22	189,05	7,3%	9,0%	1
C4	210,74	240,24	14,0%	10,1%	0,9

C5	261,91	271,21	3,5%	1,0%	1
C6	237,80	266,08	11,9%	1,1%	0,9
C7	254,66	282,33	10,9%	12,9%	1
C8	284,86	297,03	4,3%	7,9%	0,7
C9	222,31	227,05	2,1%	2,8%	1
C10	229,87	251,16	9,3%	6,2%	0,6
C11	155,31	175,21	12,8%	11,4%	0,5
C12	194,49	204,63	5,2%	5,2%	1
C13	198,88	217,45	9,3%	6,0%	0,9
C14	216,14	226,13	4,6%	2,2%	1
C15	203,28	213,62	5,1%	3,5%	0,9
Média			9,6%	5,7%	

Fonte: Elaborado pelo autor

Os resultados obtidos indicam que o algoritmo perde em eficácia com o aumento na proporção de clientes com restrição de janela de tempo. De uma média de desvio de 5,7% no conjunto A, este desvio passa a 9,6%.

Como na situação atual da *startup* a proporção de clientes com restrição de janela de tempo é semelhante ao conjunto A, o algoritmo mostra-se relevante para este cenário. Caso a *Liv Up* opte por melhorar o nível de serviço da operação, deve se atentar ao fato de que a eficácia é reduzida, embora esteja ainda em patamar aceitável.

Nota-se também que 8 dos 15 α^* encontram-se abaixo de 1, o que indica que o termo c_{12} relativo ao tempo tem maior relevância que anteriormente nesse conjunto de problemas.

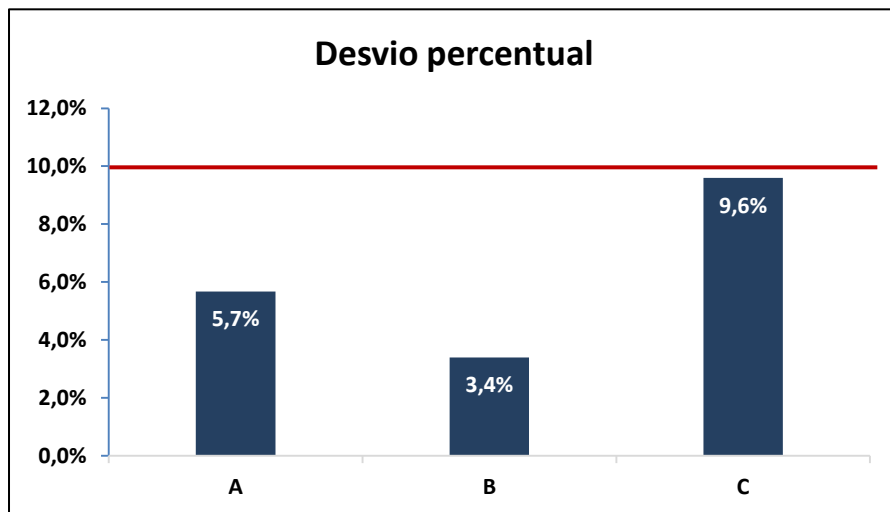
*

A partir das análises de desvio dos algoritmos em diferentes conjuntos de problema, é possível obter-se uma visão inicial de sua eficácia, mesmo que para um número de clientes reduzido. Uma comparação entre os três conjuntos é retratada no Gráfico 5.

Em conclusão, existe uma pequena perda de eficácia em problemas com retorno de veículos e uma maior perda para problemas com uma maior proporção de clientes com restrição de janela de tempo, o que pode ser visto como oportunidade de melhoria do algoritmo.

No entanto, para os 3 conjuntos de problemas, o desvio encontrou-se abaixo de 10% em relação ao ótimo. De acordo com Vidal *et al.* (2013), heurísticas construtivas (categoria em que se enquadra a heurística desenvolvida neste trabalho) geralmente são capazes de produzir soluções com desvios de 10% a 15% quando aplicadas a problemas de roteirização. Assim, a eficácia do algoritmo pode ser considerada satisfatória para o cenário atual da *startup*.

Gráfico 5 - Comparação do desvio em relação ao ótimo para os conjuntos A, B e C de problemas



Fonte: Elaborado pelo autor

6 ANÁLISE DOS RESULTADOS OBTIDOS PELA HEURÍSTICA ADAPTADA

No capítulo anterior, alguns testes do algoritmo foram realizados para verificar a eficácia do mesmo, medida pelo desvio em relação à solução ótima. Num universo de 11 clientes por roteirização, a heurística desenvolvida mostrou-se satisfatória. Porém, o cenário real da *startup* envolve, atualmente, cerca de 50 clientes por período de entrega, com perspectivas de crescimento desse número. Nesse caso, não se conhece a solução ótima do problema.

No entanto, a finalidade da ferramenta heurística é trazer melhorias para a *startup*, resolvendo os problemas mencionados no início deste trabalho. Desta forma, este capítulo tem como objetivo testar a heurística desenvolvida no cenário real da *startup* e avaliar os benefícios trazidos pelo seu uso. Primeiramente, será feita uma comparação entre os resultados obtidos pelo algoritmo e dados reais. Em seguida, análises de sensibilidade em relação às janelas de tempo e retorno dos veículos serão realizadas, de forma a auxiliar na tomada de decisões. Por fim, o impacto para a *Liv Up* dos resultados obtidos será avaliado, tomando como base os problemas identificados no segundo capítulo.

6.1 Comparação com solução atual

A fim de avaliar o impacto no custo de logística, o algoritmo será testado num conjunto de problemas reais da *startup*. Os resultados obtidos serão, em seguida, comparados com os resultados reais e a diferença entre os dois será quantificada. O custo total real foi calculado partindo-se das rotas efetivamente realizadas, disponibilizadas pela *Liv Up*.

O conjunto de problemas testado será denominado R, e é composto de 15 problemas reais referentes ao histórico da *startup*.

Conforme visto no capítulo 5, é possível melhorar os resultados através da variação do parâmetro α . Desta forma, um primeiro teste foi feito usando o α^* . Segue na Tabela 17 os resultados:

Tabela 17 - Comparação do custo real e calculado pelo algoritmo, para $0 < \alpha < 1$

Problema	# clientes	Custo Total Real (R\$)	Custo Total Obtido (R\$)	Diferença Percentual	α^*
R1	55	923,95	911,02	1,4%	0,8
R2	57	889,06	918,08	-3,3%	1
R3	61	1089,57	1046,31	4,0%	0,9
R4	51	943,83	941,57	0,2%	0,6
R5	43	733,37	757,56	-3,3%	0,9
R6	41	705,9	658,84	6,7%	0,8
R7	44	807,16	770,40	4,6%	1
R8	64	1103,39	1083,72	1,8%	0,9
R9	54	908,49	873,25	3,9%	0,6
R10	52	818,68	811,15	0,9%	0,9
R11	58	1113,91	1097,16	1,5%	0,8
R12	53	960,08	977,21	-1,8%	1
R13	58	939,93	912,85	2,9%	0,7
R14	51	902,82	889,44	1,5%	1
R15	51	956,53	929,02	2,9%	1
Média				1,6%	

Fonte: Elaborado pelo autor

Conforme resultados da tabela acima, observa-se que o custo calculado pelo algoritmo proporciona economias na maioria dos problemas. Para o conjunto de problemas testado, as diferenças variaram de -3,3% a 6,7%, com um valor médio de 1,6% e desvio-padrão de 2,8%. Novamente, percebe-se que o α^* indica que o critério c_{11} tem maior peso nos problemas.

Apesar de o algoritmo obter reduções na maioria dos casos, em alguns problemas ele se mostrou inferior à solução real. Assim, ainda é possível atingir resultados melhores. Como visto no

capítulo anterior, a variação do parâmetro α pode ser interpretada simplesmente como a criação de novas configurações de rota. Assim, se variarmos este parâmetro sem a restrição de $\alpha_1 + \alpha_2 = 1$, pode-se encontrar eventualmente resultados melhores.

Como o α^* encontrou-se próximo de 1 nos resultados obtidos na Tabela 17, pode-se dizer que α_1 tem maior peso em relação a α_2 . Assim, pode-se fixar $\alpha_2 = 0$ e trabalhar somente com α_1 , sem muito impacto nos resultados.

Desta forma, novos testes foram realizados para o mesmo conjunto de problemas, com $0 < \alpha_1 < 10$ e $\alpha_2 = 0$. Segue na Tabela 18 resultados obtidos:

Tabela 18 - Comparação do custo real e calculado pelo algoritmo, para $0 < \alpha_1 < 10$ e $\alpha_2 = 0$

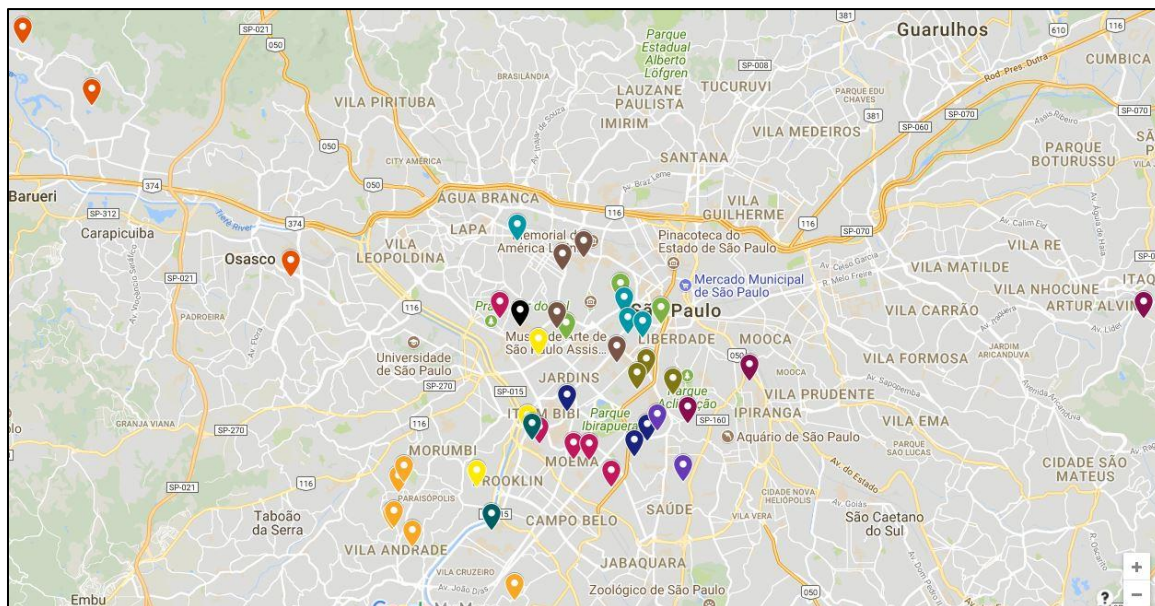
Problema	# clientes	Custo Total Real (R\$)	Custo Total Obtido (R\$)	Diferença Percentual	α_1
R1	55	923,95	885,13	4,2%	1,7
R2	57	889,06	862,53	3,0%	3,1
R3	61	1089,57	1036,89	4,8%	2,1
R4	51	943,83	925,72	1,9%	4,5
R5	43	733,37	723,23	1,4%	1,3
R6	41	705,9	658,84	6,7%	0,8
R7	44	807,16	770,40	4,6%	1
R8	64	1103,39	1070,63	3,0%	1,5
R9	54	908,49	861,92	5,1%	0,6
R10	52	818,68	806,10	1,5%	1,5
R11	58	1113,91	1087,97	2,3%	4,1
R12	53	960,08	936,79	2,4%	4,2
R13	58	939,93	911,14	3,1%	1,2
R14	51	902,82	864,89	4,2%	1,7
R15	51	956,53	900,97	5,8%	2,7
Média				3,6%	

Fonte: Elaborado pelo autor

Observa-se que houve uma melhoria incremental nos resultados através do aumento da amplitude de variação do parâmetro α . Neste caso, as reduções variaram de 1,4% a 6,7%, com um valor médio de 3,6% e desvio-padrão de 1,6%.

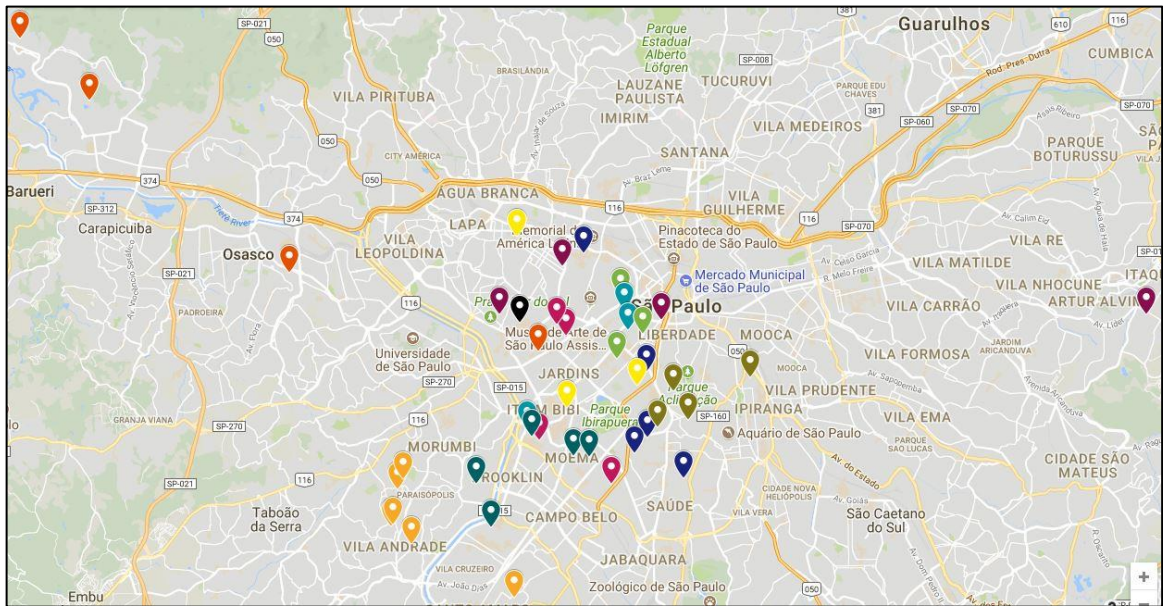
Nas figuras 12 e 13, pode-se comparar a rota gerada pelo algoritmo com a rota efetivamente realizada para o problema R6. Cada cor representa uma rota diferente.

Figura 12 - Roteiro efetivamente realizado para o problema R6



Fonte: Elaborado pelo autor

Figura 13 - Roteiro gerado pelo algoritmo para o problema R6



Fonte: Elaborado pelo autor

A principal diferença notada é a existência de menos rotas e mais clientes por rota no cenário gerado pelo algoritmo, já que a inserção de clientes é feita até se atingir a capacidade máxima do veículo e o limite de tempo.

Abaixo encontra-se um comparativo entre os dois testes realizados:

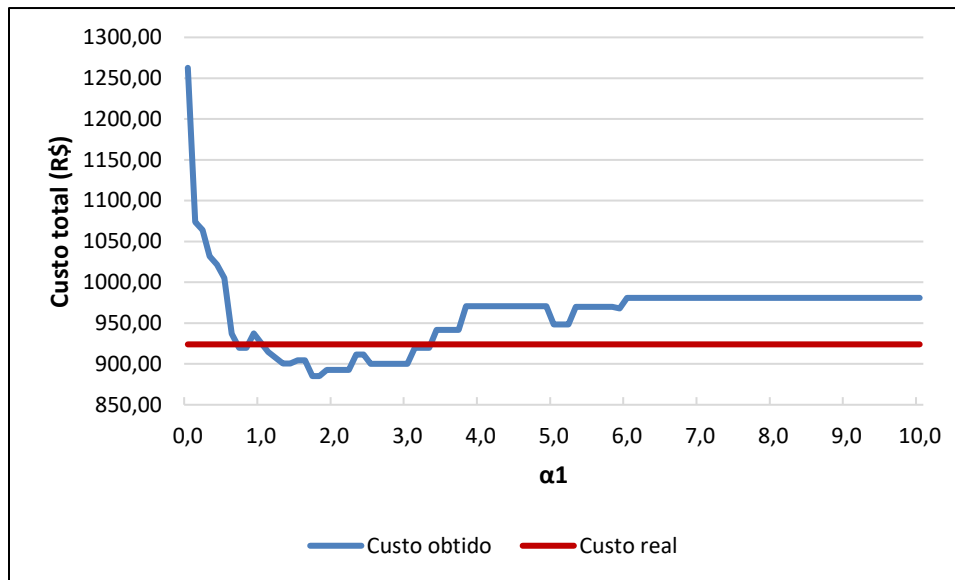
Tabela 19 - Comparativo entre resultados em função de α

	$0 < \alpha < 1$	$0 < \alpha_1 < 10$ e $\alpha_2 = 0$
Redução mín	-3,3%	1,4%
Redução média	1,6%	3,6%
Redução máx	6,7%	6,7%
Desvio-padrão	2,8%	1,6%
Custo/entrega médio	R\$17,13	R\$16,78

Fonte: Elaborado pelo autor

Para ilustrar o efeito da variação de α_1 no custo total, o Gráfico 7 foi obtido para o problema R1:

Gráfico 6 - Custo total em função de α para o problema R1



Fonte: Elaborado pelo autor

6.2 Análises de sensibilidade

Através do uso do algoritmo de roteirização, existe um potencial de redução nos custos de transporte para a *Liv Up*. Porém, além dessa redução, o algoritmo tem a vantagem de tempo de processamento, que encontra-se na ordem de 10 segundos.

Por este motivo, o algoritmo pode ser facilmente utilizado em análises de sensibilidade para a tomada de decisões. Nesta seção, duas análises serão estudadas: o aumento na proporção de clientes com restrição de janelas de tempo e a redução nos pagamentos com máquina de vale-refeição. Ambos os cenários estudados possuem relevância para a *startup*.

Vale ressaltar também que estas análises diferem das análises feitas no capítulo anterior. Enquanto as análises do capítulo 5 referem-se ao comportamento do desvio do algoritmo com a variação dos cenários, estas análises referem-se ao impacto no custo obtido pelo algoritmo (não necessariamente ótimo) com a mudança de cenários, independentemente do desvio do mesmo.

6.2.1 Aumento na proporção de restrições de janela de tempo

Como foi explicado anteriormente, os clientes têm a possibilidade de escolher um horário de recebimento dentro de uma janela de, geralmente, 1 a 2 horas. Atualmente, menos de 10% dos clientes possuem esta restrição e esta escolha é feita por telefone.

Assim, seria interessante para a *startup* oferecer este serviço para um maior número de pessoas, implementando esta opção no *website* no momento do pagamento. Isto implicaria num aumento no nível de serviço e diferenciação em relação à concorrência.

No entanto, isto também implicaria num maior custo de roteirização, sendo interessante quantificar este aumento no custo pelo aumento no nível de serviço.

Desta forma, 5 cenários foram testados no mesmo conjunto R de problemas: nenhuma restrição de janela de tempo, 25% de restrições de janela de tempo, 50%, 75% e 100%, todas com amplitudes em torno de 1 a 2 horas. Os resultados podem ser observados na Tabela 20:

Tabela 20 - Custo calculado pelo algoritmo para diferentes cenários de janela de tempo

Problema	Custo (R\$) 0%	Custo (R\$) 25%	Custo (R\$) 50%	Custo (R\$) 75%	Custo (R\$) 100%
R1	908,02	908,60	910,00	922,77	930,50
R2	872,95	854,32	887,84	868,78	871,38
R3	1034,96	1053,22	1040,41	1057,56	1050,63
R4	915,00	923,86	939,31	945,42	922,94
R5	723,23	717,04	699,28	696,86	728,29
R6	658,84	671,02	664,00	664,32	667,85
R7	770,40	781,00	793,68	779,60	796,17
R8	1081,86	1084,54	1099,17	1113,06	1095,33
R9	861,05	877,57	860,05	872,27	871,98
R10	811,15	815,30	822,59	812,83	807,65
R11	1091,08	1102,55	1128,29	1132,63	1125,34
R12	907,83	972,32	916,08	960,09	981,98

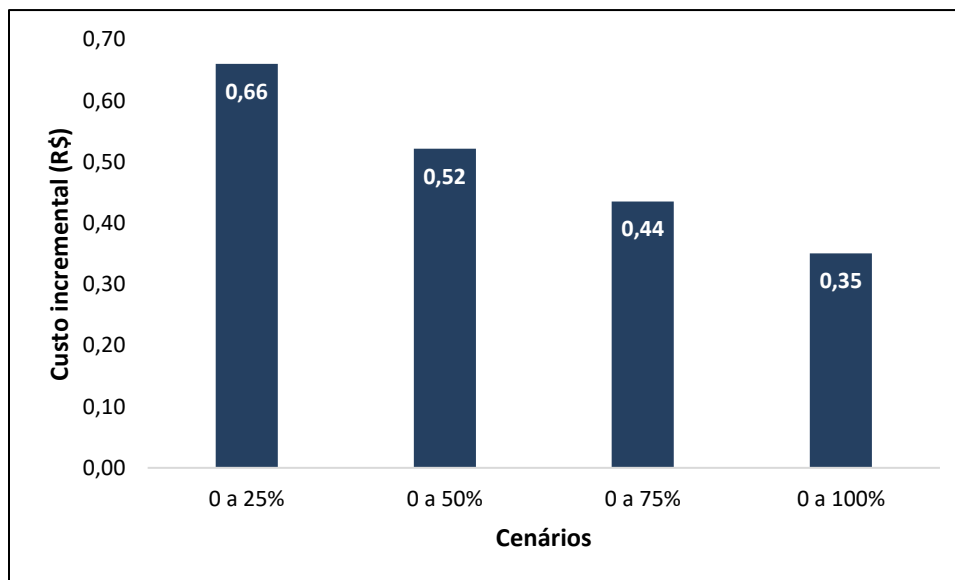
R13	892,31	917,60	931,29	952,99	939,37
R14	871,56	842,59	907,31	876,29	877,40
R15	897,32	907,30	913,10	920,25	911,62

Fonte: Elaborado pelo autor

Nota-se que, em muitos casos, um aumento no número de restrições de janela de tempo reduziu o custo obtido pelo algoritmo. Isto pode ser explicado pelo fato de as restrições alterarem a ordem de inserção do algoritmo, o que eventualmente pode levar a uma melhoria da solução.

A partir dos dados da tabela acima, é interessante calcular qual o custo incremental de adicionar uma restrição de janela de tempo no problema. Para tanto, basta calcular a razão entre a diferença de custos entre cenários e o número de restrições adicionadas. Segue abaixo graficamente esse resultado, partindo-se do cenário de nenhuma restrição para cada um dos outros 4 cenários:

Gráfico 7 - Custo incremental da adição de restrição de janela de tempo para 4 cenários



Fonte: Elaborado pelo autor

Observa-se no gráfico acima que, partindo-se de um cenário de nenhuma restrição de janela de tempo, quanto maior foi a proporção de restrições no cenário final, menor foi o custo

incremental do algoritmo, e em todos os casos houve um aumento no custo, devido à redução do espaço de soluções.

Assim, o algoritmo indica que o aumento no número de restrições implica num custo incremental que reduz com o aumento na proporção de restrições. Em média, esses custos encontram-se abaixo de R\$1,00 por restrição adicionada. Dado que o *ticket* médio encontra-se atualmente em torno de R\$220,00, o custo do aumento no nível de serviço mais do que justifica-se.

6.2.2 Redução de pagamentos com máquina de vale-refeição

Outra análise interessante para a *startup* refere-se à redução de pagamentos feitos através da máquina de vale-refeição, já que esta forma de pagamento implica no retorno do veículo ao depósito e, conseqüentemente, numa taxa de retorno.

Atualmente, a maioria dos clientes efetua o pagamento no próprio *website* no momento da compra. No entanto, ainda não há suporte para pagamento *online* através de vale-refeição, sendo necessário o transporte da máquina ao cliente. Uma alternativa seria o investimento na implantação do pagamento *online* de vale-refeição, o que eliminaria todas as taxas de retorno da roteirização.

Para este cálculo, o conjunto R de problemas foi utilizado. O custo original do algoritmo foi comparado com o custo eliminando-se todas as solicitações de máquina de cartão. Segue na Tabela 21 os resultados:

Tabela 21 - Economia incremental na eliminação das taxas de retorno no conjunto R de problemas

Problema	# solicitações	Custo original (R\$)	Custo sem solicitações (R\$)	Economia incremental (R\$)
R1	13	911,02	827,39	6,43
R2	11	918,08	790,40	11,61
R3	13	1046,31	928,98	9,03
R4	14	941,57	809,88	9,41
R5	7	757,56	642,31	16,46

R6	5	658,84	618,87	7,99
R7	13	770,40	697,28	5,62
R8	25	1083,72	915,62	6,72
R9	11	873,25	781,11	8,38
R10	16	811,15	696,61	7,16
R11	24	1097,16	923,72	7,23
R12	19	977,21	800,99	9,27
R13	16	912,85	806,24	6,66
R14	10	889,44	796,42	9,30
R15	12	929,02	803,99	10,42
Média				8,78

Fonte: Elaborado pelo autor

Para o conjunto R de problemas, houve economia incremental variando de R\$ 5,62 a R\$ 16,46, com média de R\$ 8,78 por solicitação retirada. Dado que as taxas de retorno encontram-se entre R\$ 3,90 e R\$ 12,80 e geralmente mais de um cliente que solicitou máquina situa-se na mesma rota, existe provavelmente uma economia tanto devido à eliminação das taxas de retorno quanto ao aumento da eficácia do algoritmo, o que é consistente com os resultados obtidos na análise de eficácia do capítulo anterior.

6.3 Impacto dos resultados para a *startup*

O objetivo deste trabalho é, mais do que desenvolver uma ferramenta de roteirização, trazer benefícios para a *Liv Up* através da resolução dos problemas encontrados.

Como mencionado na seção 2.2, quatro tipos de problema foram identificados: problema de tempo de processamento, problema de custo, problema de nível de serviço e problema de gestão. Com o uso da ferramenta proposta neste trabalho, existe um impacto positivo em cada um desses quatro problemas identificados, o que será abordado nesta seção.

6.3.1 Impacto no tempo de processamento

Um dos problemas identificados refere-se ao tempo do processo de roteirização. Como explicado anteriormente, este processo é feito manualmente e leva, atualmente, cerca de 1 hora para ser concluído. Com o uso de uma ferramenta automatizada, existe uma redução significativa no tempo total de roteirização.

O tempo de processamento do algoritmo pode ser dividido em três partes: tempo de processamento dos dados de entrada, tempo de construção das rotas e tempo de variação do parâmetro α .

Para o processamento dos dados de entrada, é preciso extrair as informações dos clientes (demanda, localização, restrições etc) da base de dados e inserir na ferramenta de roteirização. Isto leva cerca de 1 minuto para ser feito. Além disso, é preciso obter a matriz de distâncias através do *API* do *Google*, como explicado no capítulo 4. Isto leva cerca de 5 minutos. Assim, o processamento dos dados de entrada leva cerca de 6 minutos no total.

O tempo de construção das rotas refere-se ao tempo de processamento do algoritmo, dado um parâmetro α fixo. Para cerca de 50 clientes, este tempo é inferior a um segundo.

Por fim, a solução final é obtida através da construção de rotas para vários parâmetros α_1 e seleção daquela de menor custo. Percorre-se assim os valores de α_1 de 0 a 10, o que leva em torno de 10 segundos.

Desta forma, o tempo total de processamento para 2 roteirizações por dia é cerca de 12 minutos. Acrescentado-se uma margem de 3 minutos, tem-se um tempo total de 15 minutos. Comparando este valor com o atual, existe uma economia de tempo de 75%. Isto representa 45 minutos a menos por dia, ou 0,09 FTE (*Full Time Equivalent*), que podem ser utilizados em tarefas produtivas para a *startup*.

6.3.2 Impacto no custo

Outro problema indentificado refere-se ao custo de transporte das entregas. Como o processo de roteirização atual é feito manualmente, algumas configurações de rota de menor custo podem passar despercebidas.

Com o uso do algoritmo de roteirização proposto, existe indício de redução do desvio em relação ao custo ótimo e economia, mesmo que incremental, no custo total. Como visto no início do capítulo, esta economia foi, em média, de 3,6%.

Assim, dado que o número médio de entregas por dia é 90 e o custo médio por entrega é R\$ 19,20, temos que o custo de entrega é, por dia, R\$ 1.728,00 e, por ano, R\$ 435.456,00 (dias úteis).

Com a economia de 3,6% nos custos, tem-se uma economia de cerca de R\$15.676,00 anuais.

A redução proporcionada pelo algoritmo pode ser considerada de pequena magnitude. Isso pode ser explicado pelo fato de, atualmente, haver poucas restrições de janela de tempo e poucos clientes roteirizados. Além disso, a própria estrutura de custos, dependente na maioria das vezes apenas da zona de chegada do cliente, faz com que diferentes roteiros possuam o mesmo custo, reduzindo as possibilidades de economia. Por fim, deve-se considerar o fato de que os pontos fracos do algoritmo ainda podem ser melhor explorados através de melhorias, resultando em desempenho ainda melhor.

Assim, admitindo-se um crescimento acelerado da startup e um aumento no número de restrições de janela de tempo, espera-se que a complexidade da roteirização aumente, resultando numa diferença maior entre o custo calculado e o manual e numa economia relativa e absoluta maiores.

6.3.3 Impacto no nível de serviço

O nível de serviço da área de Logística e Expedição é afetado, entre outras coisas, pela entrega dos produtos no horário definido pelo cliente. Como o processo de roteirização é manual, não é possível oferecer a possibilidade de escolha de horário a todos os clientes no momento do pagamento, pois isto implicaria num aumento na complexidade da roteirização.

Assim, a ferramenta de roteirização, por ter um tempo de processamento baixo e economia no custo total, permite que este serviço seja oferecido a todos os clientes sem comprometer significativamente o custo total.

Isto é benéfico para a *startup*, uma vez que a possibilidade de escolha de um horário de recebimento do produto aumenta a qualidade percebida do serviço pelos clientes e diferencia a *Liv Up* dos demais concorrentes.

6.3.4 Impacto na gestão

Por fim, existe um impacto significativo na gestão da operação logística.

É evidente que um processo de roteirização que leva 1 hora para ser concluído não é ideal para ser usado em testes e simulações de cenários. Assim, o baixo tempo de processamento do processo automatizado acaba se desdobrando em diversas potenciais análises para a tomada de decisão de gestores.

Dois exemplos de análises foram propostos neste capítulo: a análise referente às janelas de entrega e ao retorno dos veículos.

A primeira análise permitiu atribuir um custo incremental de uma nova restrição de janela de tempo e concluiu-se que vale à pena o aumento no nível de serviço.

A segunda análise avaliou o impacto da implantação de pagamento *online* de vale-refeição no custo total obtido. Obteve-se um valor de R\$ 8,78 de economia incremental pela retirada de uma solicitação de pagamento com máquina de cartão. Dado que atualmente cerca de 30% dos clientes solicitam este meio de pagamento, teríamos uma redução de 27 solicitações por dia, ou R\$ 237,00. Anualizada, a economia seria de R\$ 59 724,00, maior que a economia com o uso do algoritmo.

Além dessas duas análises, diversas outras serão possíveis. Logo, fica clara a importância de uma ferramenta rápida para a tomada de decisão.

7 CONCLUSÃO

7.1 Considerações Finais

A *Liv Up*, *startup* na qual este trabalho se baseia, encontra-se no início da fase de crescimento acelerado, já que seu modelo de negócios já foi validado no mercado. Assim, precisa estar preparada para operar em grande escala. Por este motivo, procurou-se desenvolver uma solução de escalabilidade para sua operação logística, mais especificamente a de roteirização das entregas.

Alguns problemas foram identificados no que se refere ao custo, tempo do processo, nível de serviço e gestão da operação. Baseando-se nesses problemas e em características específicas da logística atual, inicialmente construiu-se um modelo de Programação Linear Inteira Mista para representar o Problema de Roteirização de Veículos com Janelas de Tempo, adaptado às especificidades da operação logística atual. Dado o forte caráter combinatório do problema, sua solução só foi viável para até 15 clientes, o que representa cerca de 1/3 do número de clientes roteirizados num período de entregas.

Para contornar esse problema, foi preciso desenvolver um método heurístico de resolução, que apesar de não garantir o custo ótimo, reduz significativamente a complexidade computacional do problema. A partir de uma revisão da literatura, optou-se pelo uso adaptado da heurística de inserção I1 de Solomon (1987), extensamente utilizada em trabalhos posteriores por diversos autores.

A partir de testes computacionais em alguns conjuntos de problema, verificou-se que a eficácia do algoritmo desenvolvido neste trabalho encontrou-se num nível satisfatório, com desvio inferior a 10% em relação ao ótimo.

Aplicando o algoritmo em um conjunto de 15 problemas, foi possível obter uma economia considerável de em média 3,6% para a *startup*, o que representa atualmente cerca de R\$ 15.676,00 anuais.

Além do custo, foi possível trazer impactos positivos em outros aspectos. Em relação ao tempo do processo, reduziu-se de 1 hora para 15 minutos, permitindo um uso mais produtivo do tempo do analista responsável pela roteirização. Em relação ao nível de serviço, a ferramenta facilita o atendimento às restrições de horário de entrega sem comprometer o custo total. Por fim, em

relação à gestão da operação, a ferramenta é um importante instrumento de auxílio na tomada de decisão, como foi observado nas análises de aumento da proporção de janelas de entrega e introdução de pagamentos *online* de vale-refeição.

Desta forma, pode-se concluir que este trabalho cumpriu com seu objetivo. Através do uso da ferramenta desenvolvida, espera-se auxiliar a *Liv Up* em sua fase de crescimento e contribuir para que ela se torne uma referência em seu setor. Além dos benefícios trazidos à *Liv Up*, vale ressaltar a importância deste trabalho no incentivo e apoio ao empreendedorismo.

7.2 Trabalhos Futuros

Durante o desenvolvimento do algoritmo de roteirização, algumas escolhas foram feitas de forma a simplificar o método de solução. Assim, embora existam trabalhos mais recentes na literatura sobre métodos aproximativos do Problema de Roteirização de Veículos com Janelas de Tempo, optou-se pelo uso da heurística de Solomon, de 1987.

Para a situação atual da *startup*, acredita-se que a heurística de inserção I1 seja suficiente para substituir o método manual sem adicionar complexidades desnecessárias. No entanto, como foi visto na avaliação da eficácia do algoritmo, a mudança em algumas características do problema pode acarretar perda de eficácia. Assim, para trabalhos futuros, seria interessante os pontos fracos deste algoritmo, como por exemplo o aumento na proporção de janelas de tempo, através do desenvolvimento de heurísticas mais recentes ou, até mesmo, de meta-heurísticas, capazes de reduzir ainda mais o desvio do algoritmo.

Além disso, como mencionado anteriormente, a ferramenta proposta neste trabalho é um importante instrumento de auxílio na tomada de decisão. Assim, é muito interessante para a *startup* explorar este benefício em análises de cenário. Alguns trabalhos sugeridos podem envolver o estudo de viabilidade logística de novos mercados ou de troca da frota terceirizada por frota própria, através da adaptação de parâmetros como velocidade média do veículo, capacidade do veículo e método de custeio.

REFERÊNCIAS BIBLIOGRÁFICAS

- ANTES, J.; DERIGS, U. **A New Parallel Tour Construction Algorithm for the Vehicle Routing Problem with Time Windows**. 1995. 11f. Monografia – Department of Economics and Computer Science, University of Köln, Colônia. 1995.
- ARENALES, M. N. *et al.* **Pesquisa Operacional**. Rio de Janeiro: Elsevier, 2007.
- BALAKRISHNAN, N. Simple heuristics for the Vehicle Routing Problem with Soft Time Windows. **Journal of the Operational Research Society**, v.44, n.3, p.279-287, 1993.
- BALLOU, R. H. **Gerenciamento da Cadeia de Suprimentos: Planejamento, Organização e Logística Empresarial**. 4. Ed. São Paulo: Bookman, 2001.
- BASSI, S. **Pesquisa Operacional Aplicada à Área de Logística de Transportes Rodoviários em Projetos de Grande Porte**. 2009. 113 f. Trabalho de Formatura – Escola Politécnica, Universidade de São Paulo, São Paulo. 2009.
- BEDÊ, M. A. **Sobrevivência das empresas no Brasil**. Brasília: Sebrae, 2016.
- BELFIORE, P. P. **Scatter Search para Problemas de Roteirização de Veículos com Frota Heterogênea, Janelas de Tempo e Entregas Fracionadas**. 2006. 111 f. Tese (Doutorado em Engenharia de Produção) – Escola Politécnica, Universidade de São Paulo, São Paulo. 2006.
- BELFIORE, P. P.; FAVERO, L. P. L. Problema de roteirização de veículos com janelas de tempo: revisão da literatura. In: Simpósio de Engenharia de Produção, 13., 2006. Bauru. **Anais...** Bauru, 2006.
- BODIN, L.; GOLDEN, B. Classification in vehicle routing and scheduling. **Networks**, v.11, n.2, p.97-108, 1981.
- BOTT, K.; BALLOU, R. H. Research Perspectives in Vehicle Routing and Scheduling. **Transportation Research Part A: Policy and Practice**, v.20A, n.3, p.571-574, 1990.
- BRÄYSY, O. Fast Local Searches for the Vehicle Routing Problem with Time Windows. **INFOR: Information Systems and Operational Research**, v.40, n.4, p.319-330, 2001.
- BRÄYSY, O.; GENDREAU, M. Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms. **Transportation Science**, v.39, n.1, p.104-118, 2005.
- BRESSLAU, F. L. **Distribuição de Produtos Lácteos no Interior do Estado de São Paulo**. 2010. 75 f. Trabalho de Formatura – Escola Politécnica, Universidade de São Paulo, São Paulo. 2010.
- CASEAU, Y. LABURTHER, F. Heuristics for Large Constrained Vehicle Routing Problems. **Journal of Heuristics**, v.5, n.3, p.281-303, Outubro, 1999.

CHRISTOFIDES, N. The Vehicle Routing Problem. **Revue française d'automatique, informatique, recherche opérationnelle** : Recherche Opérationnelle, v.10, n.2, p.55-70, Fevereiro, 1976.

CLARKE, G.; WRIGHT, J. W. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. **Operations Research**, v.12, n.4, p.568-581, 1964.

CORDONE, R.; CALVO, R. W. A Heuristic for the Vehicle Routing Problem with Time Windows. **Journal of Heuristics**, v.7, n.2, p.107-129, Março, 2001.

CUNHA, C. B. **Uma contribuição para o Problema de Roteirização de Veículos com Restrições Operacionais**. 1997. 111 f. Tese (Doutorado em Engenharia de Transportes) – Escola Politécnica, Universidade de São Paulo, São Paulo. 1997.

CUNHA, C. B.; GUALDA, N. D. F. Heurísticas baseadas em relaxação lagrangiana para o problema de roteirização de veículos com restrições operacionais. In: Congresso de Pesquisa e Ensino em Transportes, 11., 1997. Rio de Janeiro. **Anais...** Rio de Janeiro: ANPET, 1997. v.2, p.843-855.

DANTZIG, G. B.; RAMSER, J.H. The Truck Dispatching Problem. **Management Science**, v.6, n.1, p.80-91, Outubro, 1959.

DESROCHERS, M.; LAPORTE, G. Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints. **Operations Research Letters**, v.10, n.1, p.27-36, Fevereiro, 1991.

DULLAERT, W. **A sequential insertion heuristic for the Vehicle Routing Problem with Time Windows with relatively few customers per route**. 2000. 21f. Monografia – *Faculty of Applied Economics, University of Antwerp*, Antuérpia. 2000.

FIGUEIREDO, L. H.; CARVALHO, P. C. P. **Introdução à Geometria Computacional**. Rio de Janeiro: IMPA, 1991.

FIORDE NEWS. **Venda de produtos orgânicos cresce em 2016**. Disponível em: <<http://www.fiorde.com.br/wordpress/blog/venda-de-produtos-organicos-cresce-em-2016>>. Acesso em: 10 de set. 2017.

FISHER, M.; JAIKUMAR, R. A Generalized Assignment Heuristics for Vehicle Routing. **Networks**, v11, n.2, p.109-124, 1981.

FISHER, M. L.; JÖRNSTEN, K. O.; MADSEN, O. B. G. Vehicle Routing with Time Windows: Two Optimization Algorithms. **Operations Research**, v.45, n.3, p.488-492, 1997.

FRIZZELL, P. W.; GIFFIN, J. W. The Split Delivery Vehicle Scheduling Problem with Time Windows and Grid Network Distances. **Computer & Operations Research**, v.22, n.6, p.655-667, 1995.

GAREY, M. R.; JOHNSON, D. S. **Computers and Intractability: A Guide to the Theory of NP-completeness**. Nova Iorque: W. H. Freeman, 1979.

GEHRING, H.; HOMBERGER, J. Parallelization of a Two-Phase Metaheuristic for Routing Problems with Time Windows. **Asia-Pacific Journal of Operational Research**, v.18, n. 1, p.35-47, Maio, 2001.

GILLET, B. L.; MILLER, L. A heuristic algorithm for the vehicle dispatch problem. **Operations Research**, v.22, n.4, p.340-349, 1974.

GOLDBARG, M. C.; LUNA, H. P. L. **Otimização combinatória e programação linear: modelos e algoritmos**. 2. Ed. Rio de Janeiro: Elsevier, 2005.

GOLDEN, B. L.; ASSAD, A. OR Forum – Perspectives on Vehicle Routing: Exciting New Developments. **Operations Research**, v.34, n.5, p.803-810, Outubro, 1986.

GOLDEN, B.; BALL, M.; BODIN, L. Current and future research directions in network optimization. **Computers & Operations Research**, v.8, n.2, p.71-81, 1981.

GOOGLE. **Google Maps Distance Matrix API**. Disponível em: <<https://developers.google.com/maps/documentation/distance-matrix/intro?hl=pt-br>>. Acesso em: 10 de set. 2017.

GUIMARÃES, A. B. S. *et al.* **Demografia das empresas**. Rio de Janeiro: IBGE, 2014. (Estudos e pesquisas. Informação econômica, ISSN 1679-480X ; n.27).

GRECO, S. M. S. S. *et al.* **Empreendedorismo no Brasil 2015**. Curitiba: *Global Entrepreneurship Monitor/IBPQ*, 2015.

IOANNOU, G.; KRITIKOS, M. N.; PRASTACOS, G. A greedy look-ahead heuristic for the vehicle routing problem with time windows. **Operations Research Society**, v.52, n.5, p.523-537, Maio, 2001.

IOANNOU, G.; KRITIKOS, M. N. A Synthesis of Assignment and Heuristic Solutions for Vehicle Routing with Time Windows. **The Journal of the Operational Research Society**, v.55, n.1, p.2-11, Janeiro, 2004.

KOHL, N.; MADSEN, O. B. G. An Optimization Algorithm for the Vehicle Routing Problem with Time Windows Based on Lagrangian Relaxation. **Operations Research**, v.45, n.3, p.395-406, 1997.

LENSTRA, J. K.; RINNOOY KAN, A. H. G. Complexity of Vehicle and Scheduling Problems. **Networks**, v.11, n.2, p.221-227, 1981.

MARQUES, V. J. A. **Um método heurístico de distribuição**. Estudo de caso: distribuição de sementes a partir de um Centro de Distribuição. 2007. 51 f. Dissertação (Mestrado em Engenharia de Produção) – Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2007.

MILLER, C. E.; TUCKER, A. W.; ZEMLIN, R. A. Integer programming formulations of travelling salesman problems. **Journal of the ACM**, v.7, n.4, p.326-329, Outubro, 1960.

MOLE, R. H.; JAMESON, S. R. A sequential route route-building algorithm employing a generalized savings criterion. **Operational Research Quarterly**, v.27, n.2, p.503-511, 1976.

NAGATA, Y.; BRÄYSY, O.; DULLAERT, W. A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows. **Computers & Operations Research**, v.37, n.4, p.724-737, Abril, 2010.

NEWTON, R. M.; THOMAS, W. H. Bus Routing in a Multi-School System. **Computers & Operations Research**, v.1, n.2, p.213-222, August, 1974.

NICHOLSON, T. A. J. **Optimization in industry: Optimization techniques**. Chicago: Aldine Pub.Co., 1974. v.1.

POTVIN, J. Y.; ROUSSEAU, J. M. A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. **European Journal of Operational Research**, v.66, n.3, p.331-340, 1993.

POTVIN, J. Y.; ROUSSEAU, J. M. An exchange heuristic for routing problems with time windows. **European Journal of Operational Research**, v.46, n.12, p.1433-1446, 1995.

REEVES, C. R. **Modern Heuristic Techniques for Combinatorial Problems**. New York: John Wiley & Sons Inc. 1993.

REPOUSSIS, P.; TARANTILIS, C.; IOANNOU, G. Arc-guided evolutionary algorithm for the vehicle routing problem with time windows. **IEEE Transactions on Evolutionary Computation**, v.13, n.3, p.624-647, Junho, 2009.

RUSSELL, R. A. Hybrid heuristics for the vehicle routing problem with time windows. **Transportation Science**, v.29, n.2, p.156-166, 1995.

SCHULZE, J.; FAHLE, T. A parallel algorithm for the vehicle routing problem with time window constraints. **Annals of Operations Research**, v.86, n.0, p.585-607, Janeiro, 1999.

SHAW, P. Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems. In: International Conference on Principles and Practice of Constraint Programming, 4., 1998. Pisa. **Anais...** Londres: Springer-Verlag, 1998. p.417-431.

SIPSER, M. **Introduction to the Theory of Computation**. 2. Ed. Boston: Thomson Course Technology, 2006.

SOLOMON, M. M. On the worst-case performance of some heuristics for the vehicle routing and scheduling with time windows constraints. **Networks**, v.16, n.2, p.161-174, 1986.

SOLOMON, M. M. Algorithms for the Vehicle Routing and Scheduling Problems with Time Windows Constraints. **Operations Research**, v.35, n.2, p.254-265, 1987.

SOLOMON, M. M.; BAKER, E. K.; SCHAFFER, J. R. Vehicle routing and scheduling problems with time window constraints: Efficient implementations of solution improvement procedures. In: GOLDEN, B. L.; ASSAD, A. A. (Eds.). **Vehicle routing: Methods and Studies**. Amsterdam: North-Holland, 1988. p. 85-106.

TAN, K. C.; LEE, L. H.; ZHU, Q. L. ; OU, K. Heuristic Methods for vehicle routing problem with time windows. **Artificial Intelligence in Engineering**, v.15, n.3, p.281-295, 2001.

THANGIAH, S. R.; OSMAN, I. H.; VINAYAGAMOORTHY, R.; SUN, T. Algorithms for the Vehicle Routing Problems with Time Deadlines. **American Journal of Mathematical and Management Sciences**, v.13, n.3-4, p.323-355, 1993.

VIDAL, T.; CRAINIC, T. G.; GENDREAU, M.; PRINS, C. Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. *European Journal of Operational Research*, v.23, n.1, p.1-21, Novembro, 2013.

THOMPSON, P. M.; PSARAFTIS, H. N. Cyclic Transfer Algorithms for Multivehicle Routing and Scheduling Problems. **Operations Research**, v.41, n.5, p.935-946, 1993.

TSUDA, D. S. **Modelo de Roteirização de Veículos em uma Empresa Importadora de Produtos Japoneses**. 2007. 72 f. Trabalho de Formatura – Escola Politécnica, Universidade de São Paulo, São Paulo. 2007.

VIDAL, T.; CRAINIC, T.; GENDREAU, M.; PRINS, C. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. **Computers & Operations Research**, v.40, n.1, p. 475-489, Janeiro, 2013.

WATERS, C. D. J. Vehicle Scheduling Revisited. **Journal of the Operational Research Society**, v.35, n.2, p.145-148, Fevereiro, 1984.

ANEXO A – CÓDIGO VBA DO ALGORITMO DE CÁLCULO DA ZONA DE UMA COORDENADA

```

' Função que, dada uma coordenada e os parâmetros e fronteiras de um
segmento de reta, traça uma semirreta horizontal no sentido leste, partindo
da coordenada, e verifica se ela cruza o segmento.
Function Interception(ByRef x As Double, ByRef y As Double, ByRef j As
Double, ByRef l As Double, ByRef a As Double, ByRef b As Double)

    ' Esta condição verifica as ordenadas dos extremos do segmento (j e l)
e atribui o menor valor a n e o maior a p
    If j < l Then
        n = j
        p = l
    Else
        n = l
        p = j
    End If

    ' Se a semirreta horizontal leste cruzar o segmento de reta, a função
retorna 1. Se a coordenada estiver contida no segmento, a função retorna 2.
Caso contrário, retorna 0
    If n <= y And y <= p Then
        If a * x + b = y Then
            Interception = 2
        Else
            If x < (y - b) / a Then
                Interception = 1
            Else
                Interception = 0
            End If
        End If
    Else
        Interception = 0
    End If

End Function

' Esta função verifica se uma coordenada é um ponto interno ou externo a um
polígono
Function Interno(ByRef poligono As range, ByRef x As Double, ByRef y As
Double)

    fronteira = 0
    num = 0

    ' Este laço verifica, para todos os segmentos de reta do polígono, se a
semirreta horizontal leste intercepta o segmento. O número de
interceptações é calculado caso a coordenada não pertença a algum segmento.
Senão, a função retorna que a coordenada é interna ao polígono.
    For i = 1 To poligono.Rows.Count - 1
        Select Case Interception(x, y, poligono.Cells(i, 3),
poligono.Cells(i + 1, 3), poligono.Cells(i, 4), poligono.Cells(i, 5))
            Case 2
                fronteira = 1
            Exit For
        End Select
    Next i

End Function

```

```

        Case 1
            num = num + 1
        End Select
    Next i

    Select Case Interception(x, y, poligono.Cells(i, 3), poligono.Cells(1,
3), poligono.Cells(i, 4), poligono.Cells(i, 5))
        Case 2
            fronteira = 1
        Case 1
            num = num + 1
    End Select

    ' Se a coordenada pertence a algum segmento do polígono, retorna que a
    coordenada é interna ao polígono. Senão, verifica se o número de
    intercepções é par ou ímpar (par é externo e ímpar é externo ao polígono.
    If fronteira = 1 Then
        Interno = 1
    Else
        Interno = num Mod 2
    End If

End Function

' Esta função verifica, para cada zona, se a coordenada é interna ou
externa a ela, e retorna a zona à qual a coordenada está inserida.
Function Zona(ByRef zonal As range, ByRef zona2a As range, ByRef zona2b As
range, ByRef zona3a As range, ByRef zona3b As range, ByRef x As Double,
ByRef y As Double)

    If Interno(zonal, x, y) = 1 Then 'Verifica se a coordenada está dentro
da zona 1
        Zona = 1
    Else
        If Interno(zona2a, x, y) = 1 Or Interno(zona2b, x, y) = 1 Then
'Verifica se a coordenada está dentro da zona 2
            Zona = 2
        Else
            If Interno(zona3a, x, y) = 1 Or Interno(zona3b, x, y) = 1 Then
'Verifica se a coordenada está dentro da zona 3
                Zona = 3
            Else
                Zona = 4 'Caso a coordenada não esteja em nenhuma das 3
zonas, ela estará na zona 4
            End If
        End If
    End If

End Function

```

ANEXO B – GOOGLE MAPS DISTANCE MATRIX API

```
' Através da URL
http://maps.googleapis.com/maps/api/distancematrix/json?origins=A&destinations=B", sendo A a coordenada da origem e B a do destino, este código
retorna a distância real entre A e B proveniente do servidor do Google

Function GetDistance(start As String, dest As String)
    Dim firstVal As String, secondVal As String, lastVal As String
    firstVal =
"http://maps.googleapis.com/maps/api/distancematrix/json?origins="
    secondVal = "&destinations="
    lastVal = "&key=AIzaSyCgXSC0VDRqdtR-4eHK3QsN8Z6qrypHqX0"
    Set objHTTP = CreateObject("MSXML2.ServerXMLHTTP")
    URL = firstVal & Replace(Replace(start, ",", "."), ";", ",") &
secondVal & Replace(Replace(dest, ",", "."), ";", ",")
    objHTTP.Open "GET", URL, False
    objHTTP.send ("")
    If InStr(objHTTP.responseText, """"distance"" : {"") = 0 Then GoTo
ErrorHandler
    Set regex = CreateObject("VBScript.RegExp"): regex.Pattern =
""""value"".*?([0-9]+)": regex.Global = False
    Set matches = regex.Execute(objHTTP.responseText)
    tmpVal = Replace(matches(0).SubMatches(0), ".",
Application.International(xlListSeparator))
    GetDistance = CDBl(tmpVal)
    Exit Function
ErrorHandler:
    GetDistance = -1
End Function
```


Figura 16 - Aba de instâncias relacionadas aos clientes

	A	B	C	D	E	F	G	H	I	J
1		LivUp	Demanda	X	Y	Zona	Horário mín	Horário máx		
2		0	-	-46,691	-23,557	1	0,00	0,00		
3		55	-	-46,691	-23,557	1	0,00	3,00		
4										
5		Client ID	Demanda	X	Y	Zona	Horário mín	Horário máx	VA/VR	
6		1	115	-46,652	-23,562	3	0,00	3,00	0	
7		2	346	-46,677	-23,541	2	0,00	3,00	0	
8		3	159	-46,631	-23,593	4	0,00	3,00	0	
9		4	72	-46,679	-23,620	4	0,00	3,00	1	
10		5	274	-46,626	-23,568	4	0,00	3,00	0	
11		6	167	-46,672	-23,595	3	0,00	3,00	0	
12		7	406	-46,645	-23,619	4	0,00	3,00	0	
13		8	162	-46,641	-23,554	3	0,00	3,00	0	
14		9	241	-46,687	-23,690	4	0,00	3,00	1	
15		10	132	-46,652	-23,562	3	0,00	3,00	0	
16		11	77	-46,685	-23,583	3	0,00	3,00	1	
17		12	104	-46,676	-23,596	3	0,00	3,00	1	
18		13	213	-46,651	-23,568	3	2,00	3,00	0	
19		14	151	-46,668	-23,631	4	0,00	3,00	1	
20		15	152	-46,643	-23,643	4	0,00	3,00	1	
21		16	132	-46,668	-23,605	4	0,00	3,00	0	
22		17	185	-46,741	-23,614	4	0,00	3,00	1	
23		18	274	-46,668	-23,601	4	0,00	3,00	0	
24		19	477	-46,732	-23,630	4	0,00	3,00	1	
25		20	53	-46,657	-23,552	3	0,00	3,00	0	
26		21	340	-46,663	-23,566	2	1,50	3,00	1	
27		22	59	-46,653	-23,541	3	0,00	3,00	0	
28		23	65	-46,654	-23,548	3	0,00	3,00	1	
29		24	301	-46,677	-23,621	4	0,00	3,00	1	
30		25	76	-46,650	-23,577	3	0,00	3,00	0	

Painel de controle
Database >>
DB Zonas
DB Clientes

Fonte: Elaborado pelo autor

ANEXO D – CÓDIGO INSERIDO NO SOFTWARE CPLEX

```

/*****
* OPL 12.7.1.0 Model
* Author: fellipecfm
* Creation Date: 16 juin 2017 at 15:00:02
*****/

/* Declaração de parâmetros*/

int n_clientes = ...; /* número de clientes */
range n = 1..n_clientes;
range l = 0..n_clientes+1;
range i = 0..n_clientes;
range j = 1..n_clientes+1;
range k = n;

float demanda[n]=...; /* demanda de cada cliente, em R$ */
float custo[l][l]=...; /* matriz custo sem retorno (custo do trajeto para cada
par de nós) */
float tempo[l][l]=...; /* matriz tempo (tempo do trajeto para cada par de nós) */
float inicio[j]=...; /* vetor com o instante mínimo que o veículo deve chegar no
nó */
float fim[j]=...; /* vetor com o instante máximo que o veículo deve chegar no nó
*/
int va_vr[n]=...; /*vetor binário que vale 1 se o cliente precisa de máquina de
cartão e 0, caso contrário */

int capacidade = 700; /* capacidade dos veículos, em R$ */
int M = 10; /*

/* ----- */

/* Modelagem do problema */

dvar boolean x[i][l][k]; /* variável de decisão binária que vale 1 se o trajeto
i-j é efetuado pelo veículo k e 0, caso contrário*/
dvar float s[l][k]; /* variável de decisão real que representa o instante em que
o veículo k chega no cliente l */
dvar boolean retorno[k]; /* variável de decisão binária que vale 1 se o veículo k
transportar máquina de vale-refeição e 0, caso contrário */

dexpr float objetivo = sum(a in i, b in j, c in k) custo[a][b]*x[a][b][c]; /*
função-objetivo (4.1) */

minimize objetivo;

subject to {

forall (a in n) sum(b in l, c in k) x[a][b][c] == 1; /* (4.2) */

forall (c in k) sum(a in n) demanda[a]*sum(b in l) x[a][b][c] <= capacidade; /*
(4.3) */

forall (c in k) sum(b in j) x[0][b][c] == 1; /* (4.4) */

```

```

forall (h in n, c in k) sum(a in i) x[a][h][c] == sum(b in l) x[h][b][c]; /*
(4.5) */

forall (c in k) s[0][c]==0; /* conjunto de restrições opcional que define o
instante 0 como o inicial */

forall (a in j, c in k) s[a][c] >= inicio[a]; /*(4.7) */

forall (a in j, c in k) s[a][c] <= fim[a]; /* (4.7) */

forall (a in i, b in j, c in k) s[a][c] + tempo[a][b] <= s[b][c] + (1-
x[a][b][c])*M; /* (4.6) */

forall (c in k) retorno[c] <= sum(a in n) va_vr[a]*sum(b in l) x[a][b][c]; /*
(4.8) */

forall (c in k) retorno[c] >= (sum(a in n) va_vr[a]*sum(b in l)
x[a][b][c])/n_clientes; /* (4.9) */

forall (a in n, c in k) x[a][0][c] <= 1 - retorno[c]; /* (4.10) */

}

/* Código para imprimir as rotas da solução */

int n_veiculo = 1;
int z;
int p;
int q;

execute DISPLAY {
    for(var c in k)
        if(x[0][n_clientes+1][c]==0) {
            writeln("Veículo ",n_veiculo,": 0");
            p=0;
            while(p<n_clientes+1){
                z=0;
                for(q=0;q<=n_clientes+1 && z==0;q++){
                    if(x[p][q][c]==1){
                        writeln(" > ", q);
                        if(q==0)
                            q=n_clientes+1
                        z=1;}}
                p=q-1;
            }
            n_veiculo++;}
}

```

ANEXO E – ALGORITMO ADAPTADO DA HEURÍSTICA I1 DE SOLOMON

```

Type MyType
    client As Integer
    position As Integer
    c1 As Variant
    c2 As Double
End Type

Sub Roteirizar()

    Dim n As Integer 'número de clientes
    n = ['DB Clientes'!B3].Value - 1

    Dim client_id(), end_window(), dist()
    ReDim client_id(1 To n), end_window(1 To n), dist(1 To n)
    Sheets("DB Clientes").Select
    client_id = range_to_array(range([B6], [B6].End(xlDown)), n)
    end_window = range_to_array(range([H6], [H6].End(xlDown)), n) 'instante
    final da janela de tempo
    Sheets("Matriz Distância").Select
    dist = range_to_array(range([E6], range([E6],
[E6].End(xlToRight)).Cells(1, n).Address), n) 'distâncias do depósito aos
    clientes

    Dim sort()
    ReDim sort(1 To n, 1 To 3)
    For i = 1 To n
        sort(i, 1) = client_id(i)
        sort(i, 2) = dist(i)
        sort(i, 3) = end_window(i)
    Next i

    'Ordenamento dos clientes por ordem decrescente da distância e, em
    seguida, por ordem crescente do fim da janela de tempo
    Call merge_sort(sort, 3, 2, 1, n, False)
    Call merge_sort(sort, 3, 3, 1, n, True)

    Dim init_order() As Integer
    ReDim init_order(1 To n) As Integer
    For i = 1 To n
        init_order(i) = sort(i, 1) 'ordem de inicialização das rotas
    Next i

    Dim num_free_clients As Integer, num_capfeas As Integer, num_feas As
    Integer, create_route As Integer, u As Integer, n_vehic As Integer
    Dim num_clients() As Integer, position() As Integer, cap_feasible() As
    Integer, return_route() As Integer
    num_free_clients = n 'número de clientes ainda não roteirizados
    create_route = 1 'variável que indica se é necessário criar nova rota
    n_vehic = 0 'número de veículos/rotas utilizados

    Dim capacity As Double, total_cost As Double, alpha As Double, mi As
    Double, lambda As Double, instant() As Double, cap_util() As Double
    capacity = 700
    total_cost = 0
    alpha = 0.5
    mi = 1

```

```

lambda = 1

Dim demand(), start_window(), cost_matrix(), rcost_matrix(),
time_matrix(), vr_machine()
ReDim demand(1 To n), start_window(1 To n), cost_matrix(0 To n + 1, 0
To n + 1), rcost_matrix(0 To n + 1, 0 To n + 1), time_matrix(0 To n + 1, 0
To n + 1)
ReDim vr_machine(1 To n)
Sheets("DB Clientes").Select
demand = range_to_array(range([C6], [C6].End(xlDown)), n)
start_window = range_to_array(range([G6], [G6].End(xlDown)), n)
Sheets("Matriz Custo").Select
cost_matrix = range_to_matrix(range([H3], range([H3],
[H3].End(xlToRight).End(xlDown)).Cells(n + 2, n + 2).Address), n)
Sheets("Matriz Custo Retorno").Select
rcost_matrix = range_to_matrix(range([H3], range([H3],
[H3].End(xlToRight).End(xlDown)).Cells(n + 2, n + 2).Address), n)
Sheets("Matriz Tempo").Select
time_matrix = range_to_matrix(range([G3], range([G3],
[G3].End(xlToRight).End(xlDown)).Cells(n + 2, n + 2).Address), n)
Sheets("DB Clientes").Select
ReDim Preserve start_window(1 To n + 1), end_window(1 To n + 1)
start_window(n + 1) = [G3].Value
end_window(n + 1) = [H3].Value
vr_machine = range_to_array(range([I6], [I6].End(xlDown)), n) 'vetor
binário que indica se cliente solicitou máquina de cartão (=1) ou não (=0)

Dim cl_matrix() As MyType, clc2_matrix() As MyType

Do While num_free_clients > 0

    ReDim Preserve init_order(1 To num_free_clients) As Integer
    If create_route Then 'criação de nova rota
        n_vehic = n_vehic + 1

        ReDim Preserve num_clients(1 To n_vehic) As Integer
        num_clients(n_vehic) = 0
        ReDim Preserve instant(0 To n + 1, 1 To n_vehic) As Double
        instant(0, n_vehic) = 0
        ReDim Preserve position(0 To n + 1, 1 To n_vehic) As Integer
        position(0, n_vehic) = 0
        ReDim Preserve cap_util(1 To n_vehic) As Double
        cap_util(n_vehic) = 0
        ReDim Preserve return_route(1 To n_vehic) As Integer

        u = init_order(1)
        position(1, n_vehic) = u
        position(2, n_vehic) = n + 1
        instant(u, n_vehic) = calc_inst(start_window, instant,
time_matrix, position, 1, n_vehic)
        instant(n + 1, n_vehic) = calc_inst(start_window, instant,
time_matrix, position, 2, n_vehic)

        num_capfeas = num_free_clients
        ReDim cap_feasible(1 To num_capfeas) As Integer
        cap_feasible = init_order
        create_route = 0
    Else 'inserção de clientes

```

```

'Filtra os clientes infactíveis em relação à restrição de
capacidade. Se não houver cliente factível, uma nova rota será iniciada
If num_capfeas = 0 Then
    create_route = 1
    GoTo Final_loop_line
Else
    ReDim Preserve cap_feasible(1 To num_capfeas) As Integer
    i = 1
    Do While i <= num_capfeas
        aux = cap_feasible(i)
        If cap_util(n_vehic) + demand(aux) > capacity Then
            i = i - 1
            Call remove_client(cap_feasible, aux, num_capfeas)
            num_capfeas = num_capfeas - 1
            If num_capfeas = 0 Then
                create_route = 1
                Exit Do
            End If
            ReDim Preserve cap_feasible(1 To num_capfeas) As
Integer
        End If
        i = i + 1
    Loop
End If
If create_route = 1 Then
    GoTo Final_loop_line
End If

'Para cada cliente factível, calcula o termo c1 da heurística
I1 de Solomon (1987) e a respectiva posição de inserção
For i = 1 To num_capfeas
    new_min = c1(cost_matrix, rcost_matrix, time_matrix,
position, 1, instant, start_window, end_window, n_vehic,
num_clients(n_vehic), cap_feasible(i), mi, alpha,
vr_machine(cap_feasible(i)), return_route(n_vehic))
    p_min = 1
    For j = 2 To num_clients(n_vehic) + 1
        aux_min = c1(cost_matrix, rcost_matrix, time_matrix,
position, j, instant, start_window, end_window, n_vehic,
num_clients(n_vehic), cap_feasible(i), mi, alpha,
vr_machine(cap_feasible(i)), return_route(n_vehic))
        If new_min = "NA" Or (aux_min <> "NA" And aux_min <=
new_min) Then
            new_min = aux_min
            p_min = j
        End If
    Next j
    ReDim Preserve c1_matrix(1 To i) As MyType
    c1_matrix(i).client = cap_feasible(i)
    c1_matrix(i).position = p_min
    c1_matrix(i).c1 = new_min
Next i

'Se não houver cliente factível quanto à restrição de janelas
de tempo, cria uma nova rota
For i = 1 To num_capfeas
    If c1_matrix(i).c1 <> "NA" Then
        Exit For
    End If
Next i
If i = num_capfeas + 1 Then

```

```

        create_route = 1
        GoTo Final_loop_line
        'Senão, identifica o cliente u a ser inserido na rota
    Else
        'Calcula o termo c2 da heurística I1 de Solomon(1987) para
cada cliente factível
        j = 0
        For i = 1 To num_capfeas
            If cl_matrix(i).c1 <> "NA" Then
                j = j + 1
                ReDim Preserve clc2_matrix(1 To j) As MyType
                clc2_matrix(j).client = cl_matrix(i).client
                clc2_matrix(j).position = cl_matrix(i).position
                clc2_matrix(j).c1 = cl_matrix(i).c1
                clc2_matrix(j).c2 = lambda * cost_matrix(0,
cap_feasible(i)) - cl_matrix(i).c1
            End If
        Next i
        num_feas = j
        'Identifica o cliente u com o valor máx de c2
        new_max = clc2_matrix(1).c2
        p_max = clc2_matrix(1).position
        u = clc2_matrix(1).client

        If num_feas > 1 Then
            For i = 2 To num_feas
                If clc2_matrix(i).c2 >= new_max Then
                    new_max = clc2_matrix(i).c2
                    p_max = clc2_matrix(i).position
                    u = clc2_matrix(i).client
                End If
            Next i
        End If
        'Atualiza as posições da rota
        For i = num_clients(n_vehic) + 1 To p_max Step -1
            position(i + 1, n_vehic) = position(i, n_vehic)
        Next i
        position(p_max, n_vehic) = u
        'Atualiza os instantes
        For i = p_max To num_clients(n_vehic) + 2
            instant(position(i, n_vehic), n_vehic) =
calc_inst(start_window, instant, time_matrix, position, i, n_vehic)
        Next i
    End If

End If

If return_route(n_vehic) = 0 Then
    return_route(n_vehic) = vr_machine(u)
End If
cap_util(n_vehic) = cap_util(n_vehic) + demand(u)
num_clients(n_vehic) = num_clients(n_vehic) + 1
Call remove_client(init_order, u, num_free_clients)
Call remove_client(cap_feasible, u, num_capfeas)
num_free_clients = num_free_clients - 1
num_capfeas = num_capfeas - 1

Final_loop_line:
Loop

total_cost = 0

```



```

    For i = 1 To n_vehic
        total_cost = total_cost + route_cost(i, position, cost_matrix,
rcost_matrix, num_clients(i), return_route(i))
    Next i

    Sheets("Painel de controle").Select 'excel
    [B2].Value = total_cost

End Sub

'Esta função calcula o custo de uma rota, dada a ordem de atendimento
Function route_cost(n_route, position() As Integer, cost_matrix(),
rcost_matrix(), num_clients As Integer, return_route As Integer)

    route_cost = 0
    If return_route = 1 Then
        For i = 1 To num_clients + 1
            route_cost = route_cost + rcost_matrix(position(i - 1,
n_route), position(i, n_route))
        Next i
    Else
        For i = 1 To num_clients + 1
            route_cost = route_cost + cost_matrix(position(i - 1, n_route),
position(i, n_route))
        Next i
    End If

End Function

'Esta função calcula o critério c1 da heurística
Function c1(cost(), rcost(), time() As Variant, position() As Integer, p,
instant() As Double, start_window(), end_window(), n_vehic As Integer,
num_clients As Integer, u As Integer, mi As Double, alpha As Double, vr,
return_route As Integer)

    Dim instant_u As Double
    If start_window(u) > instant(position(p - 1, n_vehic), n_vehic) +
time(position(p - 1, n_vehic), u) Then
        instant_u = start_window(u)
    Else
        instant_u = instant(position(p - 1, n_vehic), n_vehic) +
time(position(p - 1, n_vehic), u)
    End If

    Dim new_instant
    new_instant = instant
    If start_window(position(p, n_vehic)) > instant_u + time(u, position(p,
n_vehic)) Then
        new_instant(position(p, n_vehic), n_vehic) =
start_window(position(p, n_vehic))
    Else
        new_instant(position(p, n_vehic), n_vehic) = instant_u + time(u,
position(p, n_vehic))
    End If

    If p <= num_clients Then
        For i = p + 1 To num_clients + 1
            If start_window(position(i, n_vehic)) > new_instant(position(i
- 1, n_vehic), n_vehic) + time(position(i - 1, n_vehic), position(i,
n_vehic)) Then

```

```

        new_instant(position(i, n_vehic), n_vehic) =
start_window(position(i, n_vehic))
    Else
        new_instant(position(i, n_vehic), n_vehic) =
new_instant(position(i - 1, n_vehic), n_vehic) + time(position(i - 1,
n_vehic), position(i, n_vehic))
    End If
Next i
End If

c1 = 0
For i = p To num_clients + 1 & c1 <> "NA"
    If new_instant(position(i, n_vehic), n_vehic) >
end_window(position(i, n_vehic)) Then
        c1 = "NA"
    End If
Next i

If c1 = 0 Then
    If return_route = 1 Then
        c11 = rcost(position(p - 1, n_vehic), u) + rcost(u, position(p,
n_vehic)) - rcost(position(p - 1, n_vehic), position(p, n_vehic)) * mi
    ElseIf vr = 0 Then
        c11 = cost(position(p - 1, n_vehic), u) + cost(u, position(p,
n_vehic)) - cost(position(p - 1, n_vehic), position(p, n_vehic)) * mi
    ElseIf p = n + 1 Then
        c11 = rcost(position(p - 1, n_vehic), u) + rcost(u, position(p,
n_vehic)) - cost(position(p - 1, n_vehic), position(p, n_vehic)) * mi
    Else
        c11 = rcost(position(p - 1, n_vehic), u) + rcost(u, position(p,
n_vehic)) + rcost(position(num_clients, n_vehic), position(num_clients + 1,
n_vehic)) - cost(position(p - 1, n_vehic), position(p, n_vehic)) * mi
    End If
    c12 = new_instant(position(p, n_vehic), n_vehic) -
instant(position(p, n_vehic), n_vehic)
    c1 = alpha * c11 + (1 - alpha) * c12
End If

End Function

'Esta função retorna o instante de chegada do veículo no cliente, dada sua
posição na rota
Function calc_inst(start_window(), instant() As Double, time_matrix(),
position() As Integer, p, n_vehic As Integer)

    If start_window(position(p, n_vehic)) > instant(position(p - 1,
n_vehic), n_vehic) + time_matrix(position(p - 1, n_vehic), position(p,
n_vehic)) Then
        calc_inst = start_window(position(p, n_vehic))
    Else
        calc_inst = instant(position(p - 1, n_vehic), n_vehic) +
time_matrix(position(p - 1, n_vehic), position(p, n_vehic))
    End If

End Function

'Função auxiliar que remove um cliente de um vetor
Sub remove_client(ByRef vector() As Integer, ByVal client_id As Integer,
ByVal n As Integer)

    For i = 1 To n

```

```
        If vector(i) = client_id Then
            Exit For
        End If
    Next i

    If n <> 1 Then
        For j = i To n - 1
            vector(j) = vector(j + 1)
        Next j
    Else
        vector(1) = 0
    End If

End Sub
```


ANEXO F – INSTÂNCIAS DO CONJUNTO A DE PROBLEMAS

Tabela 22 - Instâncias do conjunto A de problemas

PROBLEMA A1						
ID Cliente	Demanda	Latitude	Longitude	Horário mín	Horário máx	VA/VR
1	R\$295,74	-23,565	-46,650	0	3	0
2	R\$182,40	-23,630	-46,707	0	3	1
3	R\$162,55	-23,632	-46,640	0	2	0
4	R\$348,30	-23,590	-46,674	0	3	0
5	R\$293,67	-23,556	-46,464	0	3	0
6	R\$92,60	-23,611	-46,708	0	3	0
7	R\$247,10	-23,572	-46,655	0	3	0
8	R\$245,43	-23,544	-46,642	0	3	0
9	R\$254,80	-23,519	-46,497	0	3	0
10	R\$293,85	-23,562	-46,651	0	3	0
11	R\$122,00	-23,626	-46,694	0	3	0
PROBLEMA A2						
ID Cliente	Demanda	Latitude	Longitude	Horário mín	Horário máx	VA/VR
1	R\$50,10	-23,601	-46,677	0	3	0
2	R\$252,20	-23,491	-46,869	0	3	0
3	R\$163,81	-23,606	-46,663	0	3	1
4	R\$185,41	-23,565	-46,636	0	3	1
5	R\$131,20	-23,494	-46,846	0	3	0
6	R\$156,52	-23,593	-46,674	0	3	0
7	R\$104,32	-23,593	-46,678	0	3	0
8	R\$108,64	-23,608	-46,658	0	3	0
9	R\$170,00	-23,550	-46,654	0	3	0

10	R\$364,60	-23,594	-46,674	0	3	1
11	R\$372,60	-23,595	-46,678	0	3	0

PROBLEMA A3

ID Cliente	Demanda	Latitude	Longitude	Horário mín	Horário máx	VA/VR
1	R\$141,22	-23,577	-46,680	0	3	0
2	R\$456,00	-23,602	-46,660	0	3	0
3	R\$229,23	-23,583	-46,580	0	3	0
4	R\$170,83	-23,570	-46,692	0	3	1
5	R\$305,55	-23,606	-46,663	0	3	0
6	R\$71,30	-23,592	-46,686	0	3	0
7	R\$296,91	-23,530	-46,663	2	3	0
8	R\$225,54	-23,612	-46,678	0	3	0
9	R\$107,70	-23,609	-46,571	0	3	1
10	R\$230,76	-23,565	-46,666	0	3	0
11	R\$47,26	-23,609	-46,704	0	3	0

PROBLEMA A4

ID Cliente	Demanda	Latitude	Longitude	Horário mín	Horário máx	VA/VR
1	R\$254,80	-23,612	-46,683	0	3	0
2	R\$122,80	-23,541	-46,657	0	3	0
3	R\$253,60	-23,617	-46,684	0	3	0
4	R\$256,00	-23,534	-46,679	0	3	0
5	R\$357,80	-23,536	-46,667	0	3	1
6	R\$129,25	-23,627	-46,696	0	3	1
7	R\$158,00	-23,584	-46,681	2	3	0
8	R\$700,00	-23,554	-46,567	0	3	0
9	R\$67,96	-23,585	-46,679	0	3	0

10	R\$274,00	-23,618	-46,685	0	3	0
11	R\$114,30	-23,600	-46,612	0	3	1

PROBLEMA A5

ID Cliente	Demanda	Latitude	Longitude	Horário mín	Horário máx	VA/VR
1	R\$120,43	-23,635	-46,640	0	3	1
2	R\$305,55	-23,664	-46,689	0	3	0
3	R\$225,45	-23,636	-46,638	0	3	0
4	R\$399,10	-23,655	-46,692	0	3	0
5	R\$180,20	-23,536	-46,667	0	3	0
6	R\$135,00	-23,533	-46,671	0	3	0
7	R\$106,66	-23,533	-46,679	0	3	0
8	R\$221,81	-23,532	-46,781	0	3	0
9	R\$104,90	-23,506	-46,870	0	3	0
10	R\$194,30	-23,525	-46,738	0	3	0
11	R\$253,50	-23,530	-46,727	0	3	1

PROBLEMA A6

ID Cliente	Demanda	Latitude	Longitude	Horário mín	Horário máx	VA/VR
1	R\$258,40	-23,613	-46,738	0	3	0
2	R\$256,70	-23,612	-46,630	0	3	1
3	R\$352,70	-23,562	-46,635	0	3	0
4	R\$625,20	-23,552	-46,638	0	3	0
5	R\$150,31	-23,558	-46,606	0	3	0
6	R\$230,22	-23,534	-46,717	0	3	0
7	R\$184,70	-23,632	-46,706	0	3	0
8	R\$234,10	-23,648	-46,663	0	3	1
9	R\$252,20	-23,577	-46,591	0	3	1
10	R\$181,90	-23,647	-46,704	0	2	1

11	R\$167,32	-23,637	-46,639	0	3	1
----	-----------	---------	---------	---	---	---

PROBLEMA A7

ID Cliente	Demanda	Latitude	Longitude	Horário mín	Horário máx	VA/VR
1	R\$232,20	-23,643	-46,671	0	3	0
2	R\$81,90	-23,629	-46,709	0	3	0
3	R\$67,60	-23,595	-46,670	0	3	1
4	R\$473,94	-23,619	-46,605	0	3	0
5	R\$67,90	-23,616	-46,629	2	3	0
6	R\$206,70	-23,606	-46,663	0	3	0
7	R\$184,70	-23,484	-46,630	0	3	0
8	R\$68,68	-23,606	-46,716	0	3	1
9	R\$170,87	-23,668	-46,540	0	3	0
10	R\$306,18	-23,510	-46,603	0	3	1
11	R\$252,09	-23,534	-46,682	0	3	0

PROBLEMA A8

ID Cliente	Demanda	Latitude	Longitude	Horário mín	Horário máx	VA/VR
1	R\$356,80	-23,571	-46,655	0	2	0
2	R\$229,23	-23,667	-46,530	0	3	0
3	R\$94,10	-23,592	-46,634	0	3	0
4	R\$252,60	-23,620	-46,566	0	3	1
5	R\$112,06	-23,533	-46,679	0	3	0
6	R\$124,80	-23,608	-46,691	0	3	0
7	R\$292,20	-23,544	-46,567	0	3	1
8	R\$700,00	-23,610	-46,631	0	3	1
9	R\$162,10	-23,612	-46,738	0	3	1
10	R\$171,90	-23,555	-46,690	0	3	0

11	R\$267,50	-23,627	-46,663	0	3	0
----	-----------	---------	---------	---	---	---

PROBLEMA A9

ID Cliente	Demanda	Latitude	Longitude	Horário mín	Horário máx	VA/VR
1	R\$83,60	-23,648	-46,539	0	3	0
2	R\$198,20	-23,611	-46,630	0	3	0
3	R\$102,80	-23,578	-46,674	0	3	0
4	R\$290,70	-23,578	-46,674	0	3	0
5	R\$173,70	-23,530	-46,637	0	3	1
6	R\$227,07	-23,591	-46,635	0	3	1
7	R\$192,00	-23,540	-46,661	0	3	0
8	R\$210,50	-23,570	-46,642	0	3	0
9	R\$252,40	-23,602	-46,678	0	3	0
10	R\$109,70	-23,500	-46,649	2	3	0
11	R\$192,74	-23,694	-46,557	0	3	1

PROBLEMA A10

ID Cliente	Demanda	Latitude	Longitude	Horário mín	Horário máx	VA/VR
1	R\$376,50	-23,618	-46,695	0	3	1
2	R\$232,11	-23,597	-46,671	0	3	1
3	R\$263,60	-23,543	-46,559	0	3	1
4	R\$97,93	-23,623	-46,667	1	3	1
5	R\$314,19	-23,593	-46,624	0	3	0
6	R\$189,80	-23,579	-46,639	0	3	0
7	R\$84,90	-23,635	-46,640	0	3	1
8	R\$107,30	-23,588	-46,625	0	3	0
9	R\$336,10	-23,533	-46,674	0	3	0
10	R\$222,80	-23,571	-46,649	0	3	0

11	R\$211,50	-23,492	-46,636	0	3	1
PROBLEMA A11						
ID Cliente	Demanda	Latitude	Longitude	Horário mín	Horário máx	VA/VR
1	R\$133,40	-23,552	-46,679	2	3	1
2	R\$164,62	-23,598	-46,661	0	3	1
3	R\$141,22	-23,527	-46,694	0	3	0
4	R\$155,70	-23,541	-46,662	0	3	0
5	R\$267,20	-23,544	-46,652	0	3	0
6	R\$294,50	-23,559	-46,683	0	3	0
7	R\$437,60	-23,605	-46,718	0	3	0
8	R\$353,20	-23,591	-46,682	0	3	1
9	R\$39,50	-23,562	-46,683	0	1	0
10	R\$20,60	-23,574	-46,688	0	3	0
11	R\$162,46	-23,579	-46,674	0	3	0
PROBLEMA A12						

ID Cliente	Demanda	Latitude	Longitude	Horário mín	Horário máx	VA/VR
1	R\$292,40	-23,529	-46,694	0	3	0
2	R\$406,50	-23,594	-46,644	0	3	0
3	R\$336,70	-23,646	-46,670	0	3	0
4	R\$342,20	-23,690	-46,673	0	3	0
5	R\$119,50	-23,629	-46,636	0	3	1
6	R\$126,40	-23,651	-46,668	0	3	0
7	R\$363,96	-23,553	-46,652	0	3	0
8	R\$412,30	-23,551	-46,647	0	3	0
9	R\$181,09	-23,572	-46,636	0	3	0
10	R\$149,77	-23,566	-46,643	0	2	0
11	R\$230,94	-23,565	-46,649	0	3	0

PROBLEMA A13

ID Cliente	Demanda	Latitude	Longitude	Horário mín	Horário máx	VA/VR
1	R\$193,60	-23,564	-46,759	0	3	1
2	R\$281,97	-23,593	-46,686	0	3	1
3	R\$506,20	-23,560	-46,669	0	3	0
4	R\$127,60	-23,595	-46,685	0	3	1
5	R\$235,53	-23,542	-46,651	0	3	0
6	R\$109,90	-23,616	-46,686	0	3	0
7	R\$268,47	-23,543	-46,671	2	3	0
8	R\$61,79	-23,648	-46,539	0	3	0
9	R\$236,52	-23,556	-46,676	0	3	0
10	R\$312,60	-23,571	-46,713	0	3	0
11	R\$234,45	-23,622	-46,569	0	3	0

PROBLEMA A14

ID Cliente	Demanda	Latitude	Longitude	Horário mín	Horário máx	VA/VR
1	R\$187,20	-23,512	-46,656	0	3	1
2	R\$211,42	-23,557	-46,679	0	3	0
3	R\$71,56	-23,659	-46,695	0	3	1
4	R\$356,58	-23,685	-46,544	0	3	0
5	R\$155,70	-23,553	-46,676	0	3	0
6	R\$298,80	-23,541	-46,698	0	3	0
7	R\$264,50	-23,563	-46,683	0	3	0
8	R\$340,40	-23,569	-46,637	0	3	0
9	R\$230,30	-23,536	-46,672	0	3	0
10	R\$94,15	-23,573	-46,652	0	3	0
11	R\$211,42	-23,558	-46,655	0	3	0

PROBLEMA A15

ID Cliente	Demanda	Latitude	Longitude	Horário mín	Horário máx	VA/VR
1	R\$282,90	-23,611	-46,708	0	3	0
2	R\$235,89	-23,580	-46,717	0	3	1
3	R\$412,20	-23,603	-46,631	0	3	1
4	R\$345,50	-23,605	-46,659	0	2	1
5	R\$160,70	-23,614	-46,663	0	3	0
6	R\$331,70	-23,548	-46,711	0	3	0
7	R\$136,99	-23,561	-46,663	0	3	0
8	R\$139,40	-23,538	-46,689	0	3	0
9	R\$175,96	-23,610	-46,680	0	3	0
10	R\$316,90	-23,570	-46,612	0	3	1
11	R\$227,70	-23,565	-46,666	0	3	1

Fonte: Elaborado pelo autor