

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

MATEUS MIRANDA DO NASCIMENTO

**PROPOSTA DE UM ALGORITMO SIMULATED ANNEALING PARA A
ROTEIRIZAÇÃO PERIÓDICA DE VEÍCULOS EM UMA EMPRESA DE COLETA DE
RESÍDUOS SÓLIDOS**

LONDRINA

2022

MATEUS MIRANDA DO NASCIMENTO

**PROPOSTA DE UM ALGORITMO SIMULATED ANNEALING PARA A
ROTEIRIZAÇÃO PERIÓDICA DE VEÍCULOS EM UMA EMPRESA DE COLETA DE
RESÍDUOS SÓLIDOS**

**Proposal of a simulated annealing algorithm for the periodic routing of vehicles
in a solid waste collection company**

Trabalho de conclusão de curso de graduação
apresentado como requisito para obtenção do título de
Bacharel em Engenharia de Produção da Universidade
Tecnológica Federal do Paraná (UTFPR).
Orientador: Prof. Dr. Rogerio Tondato.

LONDRINA

2022



[4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Esta licença permite remixe, adaptação e criação a partir do trabalho, para fins não comerciais, desde que sejam atribuídos créditos ao(s) autor(es) e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

MATEUS MIRANDA DO NASCIMENTO

**PROPOSTA DE UM ALGORITMO SIMULATED ANNEALING PARA A
ROTEIRIZAÇÃO PERIÓDICA DE VEÍCULOS EM UMA EMPRESA DE COLETA DE
RESÍDUOS SÓLIDOS**

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do título de
Bacharel em Engenharia de Produção da Universidade
Tecnológica Federal do Paraná (UTFPR).

Data de aprovação: 22/novembro/2022

Rogério Tondato
Doutor
Universidade Tecnológica Federal do Paraná

Bruno Samways dos Santos
Doutor
Universidade Tecnológica Federal do Paraná

Rafael Henrique Palma Lima
Doutor
Universidade Tecnológica Federal do Paraná

LONDRINA

2022

RESUMO

A lei nº 12.305 de 2 de agosto de 2010 teve entre seus desdobramentos a emergência do mercado de empresas de coleta privada de resíduos sólidos, como a analisada neste estudo, que atende estabelecimentos não abrangidos pela coleta pública municipal de resíduos. Visando reduzir os custos de deslocamento, mão-de-obra, manutenção, entre outros, propõe-se um algoritmo *Simulated Annealing* para a otimização das rotas para visita aos 356 pontos de coleta atendidos pela empresa, rotas essas planejadas no horizonte de uma semana para múltiplos veículos de capacidade homogênea. No problema, são consideradas restrições de capacidade, jornada de trabalho e a periodicidade de atendimento aos pontos. Utilizando-se da relocação entre rotas como mecanismo de exploração de vizinhança e de uma estrutura baseada na heurística do vizinho mais próximo para obtenção da solução inicial, o algoritmo mostrou-se eficaz para otimização das rotas, onde foram encontrados percursos de custo global 5,2% menor que os atualmente utilizados para atendimento a estes mesmos pontos.

Palavras-chave: simulated annealing; roteirização de veículos; análise combinatória; coleta de resíduos.

ABSTRACT

Law No. 12,305 of August 2, 2010 had, among its consequences, the emergence of the market for private solid waste collection companies, such as the one analyzed in this study, which serves establishments not covered by municipal garbage collection service. In order to reduce travel, labor and maintenance costs, among others, a Simulated Annealing algorithm is proposed for the optimization of routes to visit the 356 collection points served by the company, routes planned within a week to multiple vehicles of homogeneous capacity. For the problem, capacity restrictions, working hours and the frequency of attendance to the points are considered. Using the relocation between routes as a neighborhood exploration mechanism and a structure based on the nearest neighbor heuristic to obtain the initial solution, the algorithm proved to be effective for route optimization, where paths with a global cost of 7% were found lower than those currently used to service these same points.

Keywords: simulated annealing; vehicle routing problem; combinatorial analysis; waste collection.

LISTA DE ILUSTRAÇÕES

Figura 1 - Exemplo de vetor solução.....	33
Figura 2 - Pseudocódigo para a definição da agenda de coleta da solução inicial.....	34
Figura 3 - Pseudocódigo para o algoritmo de agrupamento dos pontos.....	35
Figura 4 – Pseudocódigo para a geração da solução inicial.....	36
Figura 5 – Estrutura da terceira etapa para geração da solução inicial.....	37
Figura 6 – Pseudocódigo para a obtenção do vizinho mais próximo.	38
Figura 7 – Pseudocódigo para gerar várias soluções iniciais.	38
Figura 8 – Pseudocódigo para a obtenção do custo total das rotas.	39
Figura 9 – Pseudocódigo para obtenção das distâncias e horas extras.....	39
Figura 10 – Pseudocódigo para a estrutura de vizinhança.	41
Figura 11 – Pseudocódigo para frequência igual a 1.....	41
Figura 12 – Pseudocódigo para frequência igual a 0.....	42
Figura 13 – Pseudocódigo para frequência igual a 6.....	42
Figura 14 – Pseudocódigo para frequências de 2 à 5.	43
Figura 15 – Pseudocódigo para o <i>Simulated Annealing</i>	45
Figura 16 – Fluxograma para o <i>Simulated Annealing</i>	47
Figura 17 – Valor objetivo da solução inicial.	51
Figura 18 - Evolução do valor objetivo.....	52
Figura 19 - Evolução da distância percorrida.	53
Figura 20 - Evolução do total de horas extras.	53

LISTA DE TABELAS

Tabela 1 – Distância em quilômetros entre uma amostra de pontos de coleta.	48
Tabela 2 – Tempos de deslocamento em horas entre uma amostra de pontos de coleta.....	49
Tabela 3 - Resumo dos resultados	54

SUMÁRIO

1	INTRODUÇÃO	9
1.1	Problema de pesquisa e pergunta de partida	9
1.2	Objetivos	10
1.3	Justificativa	10
2	REFERENCIAL TEÓRICO	12
2.1	Pesquisa operacional	12
2.2	Roteirização de veículos	13
2.3	Modelos de Roteirização	15
2.4	<i>Simulated Annealing</i>	23
3	MÉTODOS E TÉCNICAS DE PESQUISA	26
4	PROPOSTA DE MODELO MATEMÁTICO	27
4.1	Descrição do problema	27
4.2	Modelo matemático	28
4.2.1	Variáveis de decisão	28
4.2.2	Parâmetros, variáveis auxiliares e conjuntos.	29
4.2.3	Função objetivo e restrições.....	30
4.3	Algoritmo proposto	33
4.3.1	Codificação da solução	33
4.3.2	Geração da solução inicial	33
4.3.3	Cálculo do custo total das rotas.....	38
4.3.4	Estrutura de vizinhança – realocação entre rotas.....	39
5	APLICAÇÃO DO MODELO MATEMÁTICO	48
5.1	Dados do problema	48
5.2	Execução do algoritmo	50
5.2.1	Parâmetros do <i>Simulated Annealing</i>	50
5.3	Resultados	52
6	DISCUSSÃO DOS RESULTADOS	54
7	CONCLUSÃO	56
	REFERÊNCIAS	58
	APÊNDICE A - Dados dos pontos de coleta	61
	APÊNDICE B - Rotas obtidas pelo SA	71
	APÊNDICE C - Rotas usadas pela empresa	76

1 INTRODUÇÃO

No Brasil, a lei nº 12.305 de 2 de agosto de 2010 define como responsabilidade dos municípios e do Distrito Federal o manejo e a destinação dos resíduos sólidos urbanos, no âmbito local. Além disso, a lei condiciona o acesso aos recursos da união destinados a serviços e iniciativas relacionadas à limpeza urbana e à elaboração de um plano municipal de gestão integrada de resíduos sólidos, que, dentre outras implicações, estabelece limites à responsabilidade do poder público quanto ao manejo destes (Brasil, 2010).

Sendo assim, em muitos municípios nos quais a política nacional definida pela lei já é aplicada, a legislação municipal estabelece um limite de geração, seja definido por meio do volume, massa ou outras características dos dejetos, de forma que esta deixe de ser qualificada como geração domiciliar e o responsável seja constituído como grande gerador, obrigando-o a elaboração de um plano de gestão de resíduos sólidos. Dessa forma, a responsabilidade pelo gerenciamento dos próprios resíduos passa ao gerador, ficando este fora do escopo da coleta pública (POLETTI, 2018).

Por sua vez, os grandes geradores, em sua maioria companhias privadas, de forma a cumprir o plano de gestão elaborado, tendem a buscar por prestadores de serviços de segregação, coleta, transporte e destinação final de resíduos, o que cria um nicho de mercado formado por empresas do ramo. Usualmente, a prestação de tais serviços de coleta é efetuada por meio de caminhões compactadores, veículos estes que passam em cada um dos estabelecimentos, recolhem os materiais e os transporta até o local de destinação final ou de armazenamento temporário, seja um aterro sanitário, cooperativa de reciclagem, usina de compostagem, dentre outros.

A operação de tais empresas de coleta, ao utilizar-se do transporte rodoviário, depara-se com os principais custos de tal modal, dentre eles o custo com combustível, lubrificação, pneus e manutenção, todos estes associados a distância percorrida pelos veículos. Além disso, pode-se citar ainda o custo com mão de obra, diretamente associado ao tempo necessário para atendimento da demanda.

1.1 Problema de pesquisa e pergunta de partida

O gasto com combustível é o principal custo associado ao transporte rodoviário, de maneira que este está diretamente associado ao quanto o veículo se

desloca: ou seja, quanto maior a distância percorrida por um veículo, maior seu consumo de combustível. Além disso, outros custos também estão diretamente relacionados ao quanto um veículo se desloca, como os de lubrificação, pneus, manutenções diversas, etc. Isto posto, reduzir o deslocamento total do caminhão durante a jornada é um dos desafios para baratear o serviço e torná-lo mais eficiente, assim como a busca por atender estes mesmos pontos utilizando-se, por exemplo, de uma quantidade menor de equipes de coleta ou ainda reduzindo o total de horas extras necessárias.

Em que medida, portanto, a aplicação de técnicas de roteirização é capaz de reduzir o custo dispendido mensalmente por uma operação de coleta de resíduos sólidos de grandes geradores em uma empresa do Paraná?

1.2 Objetivos

O objetivo geral deste trabalho é aplicar métodos de roteirização por meio de técnicas computacionais no processo de coleta de grandes geradores em uma empresa do Paraná. Como objetivos específicos, têm-se:

- Desenvolver um referencial teórico sobre pesquisa operacional e técnicas de roteirização.
- Realizar a coleta de dados junto a empresa através do sistema ERP e de rastreamento de veículos desta.
- Aplicar as técnicas de roteirização e analisar os resultados obtidos.

1.3 Justificativa

No âmbito do gerenciamento de resíduos sólidos, de 60% a 80% dos custos são relacionados à coleta e transporte (LIFSET, 1992; *apud* TIRKOLAE; MAHDAVI; ESFAHANI, 2018). Dada a preocupação crescente em todo o mundo com a gestão de resíduos, é, portanto, de grande importância o problema da otimização de rotas para tal setor (HANNAN et al., 2018). Por tais razões, foram várias as pesquisas cujo o enfoque é a otimização de rotas em operações de coleta de resíduos.

Dessa maneira, em tais pesquisas pode-se observar que este tipo de operação se beneficia da aplicação de técnicas de roteirização, tendo em vista a redução de custos por meio da otimização dos trajetos percorridos, como mostraram Tirkolae et al. (2019). No estudo, os autores realizaram o desenvolvimento e aplicação de um

algoritmo para o problema de roteirização de veículos na coleta de lixo urbano. Dessa maneira, foi possível a redução dos custos com a coleta e transporte de resíduos em 13%.

Por sua vez, Bautista, Fernández e Pereira (2008) realizaram a aplicação da heurística do Vizinho Mais Próximo e da heurística da Inserção do Mais Próximo na coleta de lixo urbano de Sant Boi de Llobregat, região metropolitana de Barcelona, na Espanha. Com a aplicação das ferramentas, obteve-se uma redução de até 37% no comprimento das rotas. Além disso, os autores destacam que, com a diminuição do tempo em coleta, há o aumento da disponibilidade dos funcionários, permitindo-os realizar outros tipos de serviço associados a limpeza urbana.

Portanto, dada a eficácia da aplicação de métodos de roteirização para a redução de deslocamentos em operações de coleta resíduos sólidos urbanos, a presente pesquisa justifica-se à medida que, assim como os estudos mostrados, também busca soluções para a redução de custos operacionais desses serviços e demonstrar como a aplicação de tais técnicas proporciona ganho de eficiência para as empresas do ramo.

2 REFERENCIAL TEÓRICO

2.1 Pesquisa operacional

A origem da Pesquisa Operacional remonta à Inglaterra, durante a Segunda Guerra Mundial, onde o contexto exigia ferramentas capazes de definir sobre a utilização mais eficiente e eficaz de recursos militares escassos, seja no âmbito da logística, da tática ou da estratégia militar. Os bons resultados atingidos pelos cientistas tiveram como consequência a disseminação das técnicas utilizadas, principalmente nos Estados Unidos, culminando no desenvolvimento do método Simplex em 1947. Tal área do conhecimento vem, desde então, sendo aplicada em diversos segmentos, sejam industriais ou comerciais, e em várias áreas, como logística, operações, *marketing* e finanças (FÁVERO, 2012).

Segunda a Sociedade Brasileira de Pesquisa Operacional (SOBRAPO, 2021, p.1), Pesquisa Operacional consiste na “área de conhecimento que estuda, desenvolve e aplica métodos analíticos avançados para auxiliar na tomada de melhores decisões nas mais diversas áreas de atuação humana”. Longaray (2012) define a Pesquisa Operacional como um conjunto de técnicas que faz uso do método científico para auxílio no processo de tomada de decisão. Em adição, Medeiros da Silva (2017, p. 1) coloca a Pesquisa Operacional como a “descrição de um sistema organizado com o auxílio de um modelo, e através da experimentação com o modelo, na descoberta da melhor maneira de operar o sistema”. Dessa forma, como define Rodrigues (2017), a Pesquisa Operacional utiliza-se de modelos matemáticos, estruturados em fases, aplicados em situações reais, como auxílio à tomada de decisão.

Dessa maneira, aplicada à Engenharia de Produção, a Pesquisa Operacional permite resolver, por meio de modelos matemáticos e métodos computacionais, problemas solucionados antes de forma empírica ou intuitiva. Permite, portanto, a otimização e o ganho de eficiência.

Andrade (2015) destaca a existência de dois principais enfoques para a Pesquisa Operacional, ambos com diferentes abordagens, porém, complementares quanto à aplicação prática no âmbito gerencial: o enfoque clássico e o enfoque atual.

O autor define o enfoque clássico, ou tradicional, como derivativo do conceito quantitativo de pesquisa operacional, ou seja, da resolução de modelos matemáticos aplicados para a obtenção de uma solução ótima. Porém, tal conceito tende a levar à

ideia de que a Pesquisa Operacional pode ser empregada somente a determinados tipos de problema, através de modelos que são uma simplificação da realidade e que, portanto, trazem soluções de aplicabilidade limitada. Há nesse enfoque um olhar voltado à técnica e à solução em si. Por sua vez, sob a perspectiva atual, tal olhar volta-se à modelagem do problema. Em tal visão, o esforço em modelar o problema leva a uma melhor compreensão deste, suas variáveis, suas interações com elementos internos e externos ao sistema, quais os resultados possíveis e quais as informações importantes e necessárias. Tal enfoque coloca, portanto, a Pesquisa Operacional como importante elemento ao moldar a maneira como o problema é visto.

Para além da modelagem ou da aplicação de modelos matemáticos, segundo Hillier e Lieberman (2013), um estudo de pesquisa operacional pode ser dividido em seis fases: a definição do problema e coleta de dados, a formulação do modelo matemático representativo do problema, a solução do problema por meio do modelo matemático e de técnicas computacionais, o teste e aprimoramento do modelo, a preparação para aplicação do modelo e, por último, a implementação deste. Outros autores, como Moreira (2018), dividem um estudo típico de Pesquisa Operacional em cinco fases, ao invés de seis, suprimindo a fase de preparação para aplicação do modelo. O autor coloca ainda que tais fases podem misturar-se umas às outras em alguns momentos. Há ainda autores como Belfiore e Fávero (2012) que colocam em adição uma última fase, onde faz-se a avaliação se o objetivo final do estudo foi atingido.

2.2 Roteirização de veículos

O Problema de Roteamento de Veículos (PRV) ou *Vehicle Routing Problem* (VRP) básico, também conhecido como Problema de Roteamento de Veículos Capacitado (PRVC), é um problema de otimização combinatória que consiste em encontrar uma rota, ou conjunto de rotas, que atenda a demanda de todos os clientes ao menor custo possível, de forma que cada cliente deva ser atendido uma vez, por uma rota, sendo que as cargas não são fracionáveis e a capacidade de carga de cada veículo é limitada e homogênea (ELSHAER; AWAD, 2020).

O PRV também pode ser compreendido como uma generalização do *Traveling Salesman Problem* (TSP) ou Problema do Caixeiro Viajante (PCV). O Problema do Caixeiro Viajante também é um problema de otimização combinatória, de forma que este determina, para uma lista de destinos, a rota mais curta que passa

por cada um deles e retorna à origem. Dessa maneira, de forma análoga, o PRV não se limita a uma única rota ou a um único veículo, mas procura a otimização de um conjunto de rotas para uma frota de veículos, minimizando, portanto, a soma do custo de todas elas. O PRV é normalmente utilizado em operações de entregas de mercadorias de um depósito central para um conjunto de clientes (KHOUI; LAOUI; ADJIH, 2019).

Dantzig e Ramser (1959) foram os primeiros a modelar, utilizando-se de uma frota homogênea de caminhões, o atendimento a vários postos de gasolina a partir de um depósito central, procurando como solução uma rota que minimize a distância total a ser percorrida. Já Clarke e Wright (1964) generalizaram esse problema como um de programação linear, buscando atender a um grupo de clientes, geograficamente dispersos ao redor de um centro de distribuição, usando-se de uma frota de caminhões com capacidade variável.

Atualmente, a aplicação do problema de roteamento de veículos difere cada vez mais das aplicações feitas por Dantzig e Ramser, assim como por Clark e Wright, à medida que as novas técnicas buscam considerar um número cada vez maior de aspectos da realidade. Dessa maneira, os problemas tornam-se cada vez maiores e complexos, de forma que a aplicação de algoritmos exatos, buscando uma solução ótima global, torna-se inviável. Sendo assim, a aplicação de meta-heurísticas e heurísticas costumam ser a maneira mais procurada como forma de resolução computacionalmente viável (BRAEKERS; RAMAËKERS; VAN NIEUWENHUYSE, 2016).

A partir, portanto, da busca pela adaptação do Problema de Roteamento de Veículos a aspectos específicos da realidade, surgem algumas variações do PRV, conforme apresentado por Elshaer e Awad (2020), entre outras variações:

- PRV com janelas de tempo: coloca a restrição de atendimento aos clientes dentro de um intervalo de tempo específico;
- Problema de Coleta e Entrega: as mercadorias precisam ser coletadas em determinados pontos e entregues em outros;
- PRV Heterogêneo: assume que os veículos possuem capacidades de carga diferentes entre si;
- PRV com vários depósitos: presume que os clientes são atendidos a partir de mais de um depósito;

- PRV periódico: situação em que os clientes são atendidos em mais de um dia, com estes podendo inclusive ser atendidos mais de uma vez;
- PRV com entregas fracionadas: as mercadorias podem ser entregues por mais de um veículo, em mais de uma rota.
- PRV estocástico: presume que um ou mais componentes do problema são aleatórios, seguindo determinada distribuição de probabilidade, como a demanda ou quais clientes devem ser atendidos.

2.3 Modelos de Roteirização

O estudo de Qiao *et al.* (2020) busca a otimização da coleta de resíduos sólidos urbanos como um Problema de Roteamento de Veículos Capacitados, buscando minimizar os custos de operação e as emissões de carbono. Os custos com pessoal (C_1) são expressos em (1) e correspondem às despesas com o motorista e a tripulação.

$$C_1 = c_v \sum_{h=1}^H \sum_{i=0}^{N+S} \sum_{j=0}^{N+S} x_{ijh} \quad (1)$$

Por sua vez, os custos com combustível (C_2) são mostrados em (3), à medida que o consumo em si é exibido em (2):

$$F_{fuel} = \sum_{h=1}^H \sum_{i=0}^{N+S} \sum_{j=0}^{N+S} \sum_{m=1}^M \sum_{r=1}^S \left(\eta_0 + \frac{\eta - \eta_0}{Q} Q_{ijh} \right) d_{ij} x_{ijh} z_{mhi}^r \quad (2)$$

$$C_2 = c_f F_{fuel}. \quad (3)$$

De igual maneira, as emissões de carbono do veículo são proporcionais ao consumo de combustível, sendo estas expressas pela Equação (4), de forma que as emissões são consideradas na função objetivo como C_3 , mostrado na Equação (5).

$$E_{co_2} = \lambda F_{fuel} \quad (4)$$

$$C_3 = c_e E_{co_2} \quad (5)$$

Por último, dado que o problema assume que a unidade de processamento de resíduos o qual os veículos destinam as cargas coletadas possui uma capacidade diária de processamento, estabelece-se um custo de penalização caso essa capacidade seja ultrapassada (C_4), sendo este mostrado em (6).

$$C_4 = p \sum_{m=1}^M \sum_{r=1}^S f_m^r \quad (6)$$

Somando-se tais custos, chega-se à função objetivo, mostrada em (7), que busca minimizá-los,

$$\begin{aligned} \text{Min}F = & c_v \sum_{h=1}^H \sum_{i=0}^{N+S} \sum_{j=0}^{N+S} x_{ijh} + \\ & (c_f + c_e \lambda) \sum_{h=1}^H \sum_{i=0}^{N+S} \sum_{j=0}^{N+S} \sum_{m=1}^M \sum_{r=1}^S \left(\eta_0 + \frac{\eta - \eta_0}{Q} Q_{ijh} \right) d_{ij} x_{ijh} z_{mhi}^r + p \sum_{m=1}^M \sum_{r=1}^S f_m^r \end{aligned} \quad (7)$$

estando sujeita às seguintes restrições:

$$\sum_{j=1}^N Q_{0jh} = 0, \forall h \in K \quad (8)$$

$$\sum_{i=1}^N Q_{ijh} \leq Q, \forall h \in K \quad (9)$$

$$\sum_{h=1}^H y_{ih} = 1, \forall i = 1, 2, \dots, N \quad (10)$$

$$\sum_{i=0}^N \sum_{h=1}^H x_{ijh} = 1, \forall j = 1, 2, \dots, N \quad (11)$$

$$\sum_{i=0}^N \sum_{h=1}^H Q_{ijh} - \sum_{i=0}^N \sum_{h=1}^H Q_{jih} = q_j, \forall j = 1, 2, \dots, N \quad (12)$$

$$x_{rjh} \leq z_{mhi}^r, \forall h \in K, r \in R, m \in T, i, j = 0, 1, \dots, N + S \quad (13)$$

$$\sum_{i=0}^N \sum_{h=1}^H x_{i0h} = 1 \quad (14)$$

$$d_{ij} = d_{ji}, \forall i, j = 0, 1, 2, \dots, N + S \quad (15)$$

A restrição mostrada pela Equação (8) garante que os veículos saiam da garagem vazios, enquanto a em (9) restringe-se que os veículos colem uma quantidade maior que sua capacidade de carga. A Equação (10) garante que todos os pontos de coleta serão atendidos, enquanto em (11) garante-se que estes serão atendidos somente uma vez. A fórmula mostrada em (12) não permite que as lixeiras sejam parcialmente coletadas. Já a Equação (13) permite que os veículos somente possam descarregar em uma instalação de descarte enquanto a visitam. Em (14) faz-

se com que os caminhões retornem à base ao fim da jornada. Por último, a restrição mostrada na Equação (15) garante que, independentemente da direção, a distância entre dois pontos é sempre a mesma.

Para a solução do modelo, os autores desenvolveram um algoritmo que combina dois outros algoritmos metaheurísticos, o *Particle Swarm Optimization (PSO)*, ou, Otimização por Enxame de Partículas, e o *Tabu Search (TS)*, em português, Busca Tabu. No estudo, o algoritmo PSO é utilizado primeiramente, obtendo-se uma solução inicial, para, em seguida, ser aplicado o TS, de forma a melhorar essa solução. No decorrer do estudo, o algoritmo PSO-TS proposto demonstrou boa capacidade de melhoria das soluções.

Algoritmo PSO, usado no estudo, foi desenvolvido em 1995 por Eberhart e Kennedy e inicializa com uma população de partículas, que consistem em soluções aleatórias viáveis, de forma que tais partículas modificam sua posição no espaço formado pelas soluções possíveis seguindo a partícula que possui a melhor solução de toda a população naquele momento. A posição das partículas ajusta-se conforme as soluções anteriores obtidas por si mesma e pelas partículas ao redor. A velocidade das partículas (ou seja, o quanto ela modifica sua posição) no espaço depende de sua velocidade atual e do quão distante ela está da melhor solução obtida por ela mesma e da distância em relação a melhor solução obtida por toda a população. (ZEINELDIN; EL-SAADANY; SALAMA, 2006).

Já a Busca Tabu, outro algoritmo usado no estudo de Qiao *et al.*, foi proposta por Glover, em 1989. O algoritmo permite a exploração da vizinhança superando ótimos locais. A metaheurística usa-se de dois conceitos principais. A lista tabu, o primeiro deles, consiste em todas as soluções já exploradas anteriormente pelo algoritmo, evitando que este se mova em círculo pelo espaço das soluções. O segundo conceito é o de critério de aspiração, onde, basicamente, se um movimento leva a uma solução melhor que a atual, o algoritmo parte dessa solução para a obtenção de uma nova, porém, como isso pode levar a busca a estacionar em um ótimo local, uma solução pior que a atual pode ser selecionada caso todas as soluções ao redor também piorem a solução (ou façam parte da lista tabu), de maneira que nesse caso, a melhor destas, ainda que pior que a solução atual, é a escolhida. A melhor solução dentre todas as exploradas desde que o algoritmo foi inicializado é a retornada como resultado (TALBI, 2009).

De forma semelhante, o estudo de Wu, Tao e Yang (2020) compreende a coleta de resíduos sólidos como um problema *Priority Considered Green Vehicle Routing Problem (PCGVRP)*, ou ainda, Problema de Roteamento de Veículos Verde com Prioridade Considerada. Além de considerar a redução das emissões de gases de efeito estufa, o problema determina a coleta de algumas lixeiras com prioridade. Dessa maneira, lixeiras de hospitais e postos de combustíveis são coletadas o mais rápido possível, de forma que é estabelecida uma importância de coleta para cada lixeira dada a influência passiva desses resíduos no meio. Sendo assim, o modelo do estudo possui três funções objetivo. A primeira delas busca a minimização da distância total, sendo esta mostrada em (16).

$$MinT_D = \sum_{i \in (BUD)} \sum_{j \in (BUD)} x_{ij}^k d_{ij} \quad (16)$$

Já a segunda função objetivo procura minimizar a geração de gases do efeito estufa, de maneira que relaciona a distância percorrida, um coeficiente de emissão de gases por litro de combustível consumido e o consumo do veículo a cada unidade de distância, sendo este relacionado a carga suportada pelo caminhão. Tal função é mostrada na Equação (17).

$$MinT_{E_{GHG}} = e \sum_{i \in B} \sum_{j \in B} \left(d_{ij} * r_0 + \left(\frac{r^* - r_0}{C_p} \right) * \left(\sum_{i \in (BUD)} \sum_{j \in (BUD)} x_{ij}^k q_j \right) \right) \quad (17)$$

Por último, a terceira função objetivo minimiza os custos totais e inclui as outras duas, citadas anteriormente, ao incluir, além de custos fixos, também custos variáveis que dependem da distância total percorrida e custos de emissão de carbono. A terceira função é expressa pela Equação (18).

$$\begin{aligned} MinT_C = & \sum_{k \in V} \sum_{j \in (BUD)} x_{0j}^k P_{fixed} \\ & + \sum_{i \in (BUD)} \sum_{j \in (BUD)} x_{ij}^k d_{ij} r_{ij} P_{fuel} \\ & + \epsilon e \sum_{i \in B} \sum_{j \in B} \left(d_{ij} * r_0 + \left(\frac{r^* - r_0}{C_p} \right) \right. \\ & \left. * \left(\sum_{i \in (BUD)} \sum_{j \in (BUD)} x_{ij}^k q_j \right) \right) \end{aligned} \quad (18)$$

O modelo do estudo de Wu, Tao e Yang (2020) sujeita as funções objetivo às seguintes restrições:

$$\sum_{k \in V} \sum_{i \in (B \cup D)} x_{ij}^k = 1, \forall j \in (B \cup D) \quad (19)$$

$$\sum_{k \in V} \sum_{j \in (B \cup D)} x_{ij}^k = 1, \forall i \in (B \cup D) \quad (20)$$

$$\sum_{i \in (B \cup D)} x_{ij}^k = \sum_{j \in (B \cup D)} x_{ij}^k = 1, \forall i \in (B \cup D), k \in V \quad (21)$$

$$\sum_{i \in (B \cup D)} \sum_{j \in (B \cup D)} x_{ij}^k q_j \leq C_p, \forall k \in V \quad (22)$$

$$(\lambda_i - \lambda_j)(t_i^k - t_j^k) \leq 0, \forall i, j \in B, k \in V \quad (23)$$

$$\sum_{i \in (B \cup D)} \sum_{j \in (B \cup D)} x_{ij}^k \leq |S| - 1, S \subseteq \{1, 2, \dots, N\}, S \neq \{\}, \forall k \in V \quad (24)$$

$$x_{ij}^k \in \{0, 1\}, \forall i, j \in (B \cup D), k \in V. \quad (25)$$

A Equação (19) restringe que cada lixeira seja coletada por somente um veículo. As equações (20) e (21) garantem que cada veículo retorne ao centro de descarte após visitar uma lixeira. A inequação (22) assegura que a capacidade é respeitada em cada uma das rotas. Já a restrição mostrada em (23) estabelece o atendimento a lixeiras de alta prioridade. A restrição exibida em (24) evita que se estabeleçam sub-rotas. Por último, a expressão (25) atribui os tipos de cada uma das variáveis.

Assim como no estudo de Qiao *et al.* (2020), o algoritmo PSO é utilizado para a obtenção de uma solução inicial, sendo esta melhorada pela meta-heurística *Simulated Annealing (SA)*, ou Recozimento Simulado, explicada na seção 2.4. No estudo, após diversos experimentos, verificou-se que o modelo proposto é eficiente e aplicável a operações de coleta de resíduos.

Já a pesquisa Mofid-Nakhaee e Barzinpour (2018) faz a aplicação do modelo desenvolvido no estudo em um distrito da cidade de Teerã. O estudo modela o problema como um *Capacitated Arc Routing Problem (CARP)* ou Problema de Roteamento de Arco Capacitado, o que é semelhante ao PRVC, porém, os clientes estão dispostos em arcos ao invés de nós, sendo estes arcos as ruas e avenidas da cidade. Além disso, os autores também tratam o problema como um *Multi-Compartment Capacitated Arc Routing Problem (MCCARP)*, Problema de Roteamento de Arco Capacidade com Multi-Compartimentos, onde leva-se em

consideração, ao invés de um veículo de compartimento único, um veículo com vários compartimentos destinados a separação de resíduos. Dado que o problema também presume a existência de instalações intermediárias para apoio à operação de coleta, o estudo coloca o problema como *MCCARPIF*, sigla para *Multi-Compartment Capacitated Arc Routing Problem with Intermediate Facilities*, ou Problema de Roteamento de Arco Capacidade com veículos Multi-Compartimento com Instalações Intermediárias. Os pesquisadores colocam o problema, portanto, como uma combinação de problema de roteamento em arco com veículo de um único compartimento simultaneamente a veículos com vários compartimentos.

O modelo proposto por Mofid-Nakhaee e Barzinpour (2018) possui a seguinte função objetivo:

$$\text{Minimize } Z = \sum_{(i,j) \in A} \sum_{k=1}^K (c_{ij} + f_k) x_{ijk} + \sum_{w=1}^W \sum_{(i,j) \in A} \sum_{k=1}^K s_{ij} y_{ijkw} \quad (26)$$

A função mostrada em (26) é dividida em duas partes e busca a minimização dos custos totais. A primeira parte expressa os custos fixos e variáveis associadas ao deslocamento ao longo do arco, enquanto a segunda parte diz respeito aos custos de manutenção dos veículos que atendem o arco. Por fim, o modelo possui as seguintes restrições:

$$\sum_{j \in V} x_{ijk} = \sum_{j \in V} x_{jik}, \forall i \in V, \forall (i, j) \in A, \forall k \in K \quad (27)$$

$$\sum_{k=1}^K y_{ijkw} + \sum_{k=1}^K y_{jikw} = 1, \forall (i, j) \in A_R, \forall w \in W \quad (28)$$

$$\sum_{(i,j) \in A_R} q_{ijw} y_{ijkw} \leq C_k, \forall k \in K, \forall w \in W \quad (29)$$

$$y_{ijkw} \leq x_{ijk}, \forall (i, j) \in A, \forall k \in K, \forall w \in W \quad (30)$$

$$\sum_{j \in V} x_{1jk} = \sum_{i \in V} x_{ipk} = 1, \forall k \in K \quad (31)$$

$$\sum_{i,j \in S} x_{ijk} = |S| - 1 + n^2 h_k^S, \forall S \subseteq V, \{1\}; S \neq \emptyset; \forall k \in K \quad (32)$$

$$\sum_{i \in S} \sum_{j \notin S} x_{ijk} \geq 1 - f_k^S, \forall S \subseteq V, \{1\}; S \neq \emptyset; \forall k \in K \quad (33)$$

$$h_k^S + f_k^S \leq 1, \forall S \subseteq V, \{1\}; S \neq \emptyset; \forall k \in K \quad (34)$$

$$h_k^S \in \{0,1\}, f_k^S \in \{0,1\}, \forall S \subseteq V, \{v_1\}; S \neq \emptyset; \forall k \in K \quad (35)$$

$$x_{ijk} \in \{0,1\}, y_{ijkw} \in \{0,1\}, \forall (i,j) \in A, \forall k \in K, \forall w \in W \quad (36)$$

Dentre as restrições, a mostrada em (27) evita o desbalanceamento de fluxo entre os veículos. As restrições (28) e (29) garantem, respectivamente, que os arcos que possuem demanda sejam atendidos e que os veículos respeitem seus limites de capacidade. A restrição em (30) obriga que os arcos sejam atendidos por veículos que passem por estes. Já a restrição (31) possui a função de fazer com que os veículos iniciem suas rotas no ponto 1 e a finalizem no ponto p. Em (32), (32) e (34) restringe-se a formação de sub-rotas e, por último, em (35) e (36) define-se as características das variáveis.

Para a solução do modelo, o estudo utiliza-se do algoritmo *Adaptive Large Neighborhood Search (ALNS)*, ou Algoritmo Adaptativo de Busca por Grandes Vizinhanças, para chegar a uma solução inicial e, em seguida, é aplicado o *Whale Optimization Algorithm (WOA)*, ou Algoritmo de Otimização de Baleia. Com a aplicação do estudo, obteve-se uma redução 20% no custo do uso de veículos de vários compartimentos, assim como a redução de 17,4% nos custos operacionais de veículos de compartimento único.

O *Adaptive Large Neighborhood Search* foi proposto por Pisinger e Ropke (2007), e consiste em um algoritmo de estrutura de pesquisa local onde, a cada iteração, uma heurística é usada para destruir a solução atual e outra é selecionada para repará-la. Dessa maneira, à medida que o algoritmo é executado, o desempenho de cada uma das heurísticas é registrado com base no quão ela foi capaz de melhorar a solução objetivo: quanto maior a pontuação da heurística, maior a probabilidade de ela ser selecionada nas iterações seguintes. Dessa maneira, vários algoritmos mais simples competem para modificar a solução atual.

Por sua vez, o *Whale Optimization Algorithm* inspira-se no comportamento de caça das baleias jubarte. Tal comportamento consiste na busca aleatória por um cardume de peixes e, à medida que a encontram, o grupo de baleias gera uma rede de bolhas em espiral sobre o cardume, direcionando os peixes para a superfície, momento em que o grupo de predadores ataca. O funcionamento do algoritmo é, basicamente, por meio de três operadores: o de presas circundantes, o de neutralização de bolhas e o de busca pelas presas. De forma análoga à natureza, cada baleia é uma solução candidata, onde a mudança de posição dessa solução dentro da vizinhança é como a busca da baleia pela presa no oceano. O algoritmo inicia com uma população de baleias, ou seja, um grupo de soluções candidatas. Visto

que ainda não se sabe a localização da presa, ou seja, a solução ótima, a melhor solução encontrada na iteração atual é utilizada para a geração de uma nova população através dos operadores, de forma que o processo se repete até que se atinja um critério de parada definido (MIRJALILI *et al.*, 2020).

Por último, o estudo de Tirkolaee, Mahdavi e Esfahani (2018) trata do Problema de Roteamento de Arco Capacitado Periódico Multi-Viagem, onde, além dos clientes estarem dispostos em arcos, a coleta destes é planejada no horizonte de uma semana. O modelo é mostrado a seguir:

$$\text{Minimize } Cost_{total} = \lambda \left(\sum_{(i,j) \in E} \sum_{k \in K} \sum_{o \in O} \sum_{t \in T} dis_{ij} x_{ijkt}^o \right) + \sum_{k \in K} \sum_{t \in T} cv_k u_{kt} \quad (37)$$

$$\sum_{(i,j) \in E} x_{ijkt}^o = \sum_{(j,i) \in E} x_{jikt}^o \quad \forall i \in V, \forall k \in K, \forall t \in T, \forall o \in O \quad (38)$$

$$\sum_{k \in K} (y_{ijkt}^o + y_{jikt}^o) = 1 \quad \forall (i,j) \text{ ou } (j,i) \in E_R, \forall t \in T, \forall o \in O \quad (39)$$

$$\sum_{(i,j) \in E_R} d_{ijt} y_{ijkt}^o + z_{kto} \Gamma_{kto} + \sum_{(i,j) \in E_R} r_{ijkt}^o \leq W_k \quad \forall k \in K, \forall t \in T, \forall o \in O \quad (40)$$

$$z_{kto} + r_{ijkt}^o \geq \hat{d}_{ijt} E_{ijkt}^o \quad \forall k \in K, \forall (i,j) \in E_R, \forall t \in T, \forall o \in O \quad (41)$$

$$-E_{ijkt}^o \leq y_{ijkt}^o \leq E_{ijkt}^o \quad \forall k \in K, \forall (i,j) \in E_R, \forall t \in T, \forall o \in O \quad (42)$$

$$y_{ijkt}^o \leq x_{ijkt}^o \quad \forall (i,j) \in E, \forall k \in K, \forall t \in T, \forall o \in O \quad (43)$$

$$\sum_{(i,j) \in E} \sum_{o \in O} x_{ijkt}^o \leq M u_{kt} \quad \forall k \in K, \forall t \in T \quad (44)$$

$$\sum_{(i,j) \in E} \sum_{o \in O} \frac{dis_{ij}}{Vel_{ijk}} x_{ijkt}^o + \sum_{(i,j) \in E_R} \sum_{o \in O} L w_{ijk} y_{ijkt}^o \leq T_{max} \quad \forall k \in K, \forall t \in T \quad (45)$$

$$\sum_{i,j \in S} x_{ijkt}^o \leq 1 + n h_{kto}^s, \quad \forall S \subseteq V \setminus \{1\}; S \neq \emptyset; \forall k \in K, \forall t \in T, \forall o \in O \quad (46)$$

$$\sum_{i \in S} \sum_{j \neq S} x_{ijkt}^o \geq 1 - f_{kto}^s \quad \forall S \subseteq V \setminus \{1\}; S \neq \emptyset; \forall k \in K, \forall t \in T, \forall o \in O \quad (47)$$

$$h_{kto}^s + f_{kto}^s \leq 1 \quad \forall S \subseteq V \setminus \{1\}; S \neq \emptyset; \forall k \in K, \forall t \in T, \forall o \in O \quad (48)$$

$$h_{kto}^s \in \{0,1\}, f_{kto}^s \in \{0,1\} \quad \forall S \subseteq V \setminus \{1\}; S \neq \emptyset; \forall k \in K, \forall t \in T, \forall o \in O \quad (49)$$

$$x_{ijkt}^o \in \{0,1\}, y_{ijkt}^o \in \{0,1\}, u_{kt} \in \{0,1\} \quad \forall (i,j) \in E, \forall k \in K, \forall t \in T, \forall o \in O \quad (50)$$

$$E_{ijkt}^o, z_{kto}, r_{ijkt}^o \geq 0 \quad \forall (i,j) \in E, \forall k \in K, \forall t \in T, \forall o \in O \quad (51)$$

Em (37) descreve-se a função objetivo do modelo, de maneira que nela busca-se minimizar os custos totais associados a operações dos veículos em cada um dos períodos. Quanto as restrições, em (38) balanceia-se o fluxo de cada veículo em cada um dos períodos. Em (39) é garantido o atendimento em cada arco somente uma vez. As restrições (40), (41) e (42) evitam que os veículos ultrapassem suas capacidades. Em (43) garante-se que cada arco é atendido por um veículo que o atravessa, mas não restringe que um veículo atravesse um arco sem atendê-lo. A restrição (44) garante que o custo de manutenção de cada veículo seja atribuído caso este seja utilizado. Em (45) limita-se o tempo de deslocamento de cada veículo ao tempo disponível. As restrições (46), (47) e (48) eliminam sub-rotas que se formem. Por fim, as restrições (49), (50) e (51) definem os tipos das variáveis.

Para obtenção da solução, os autores desenvolveram um algoritmo heurístico e, à solução obtida por este, é aplicada a meta-heurística *Simulated Annealing*. O estudo também faz uso do solver CPLEX do software GAMS de maneira a obter a solução ótima global do problema e utilizá-la para avaliar a eficiência do algoritmo proposto. Em tal comparação, a desvantagem máxima atingida pelo algoritmo desenvolvido pelos autores em relação a solução exata do problema foi de, no máximo, 2,59%, o que demonstra uma boa eficiência do algoritmo proposto.

2.4 *Simulated Annealing*

O *Simulated Annealing* (SA) foi proposto inicialmente por Kirkpatrick, Gelatt e Vecchi (1983) a partir da aproximação entre conceitos da mecânica estatística e da otimização combinatória. Nesse primeiro trabalho, o SA foi aplicado a problemas de desenho de computadores e ao TSP, como forma de testar a eficiência do algoritmo.

O SA consiste em uma técnica para problemas de otimização global que se baseia no processo de recozimento dos metais, onde a obtenção de uma estrutura cristalina resistente está associada ao aquecimento da substância seguido de um resfriamento lento. A ideia básica do SA consiste no uso de um mecanismo de busca aleatória para percorrer o espaço de soluções, porém, que não aceita só modificações que melhorem a função objetivo (YANG, 2010).

De forma análoga ao recozimento dos metais, o estado do sistema é como a solução, a energia do sistema como a função objetivo, o resfriamento como a busca local e o estado fundamental do sistema como a solução ótima. Gera-se uma solução inicial aleatória a cada iteração. Caso tal solução seja melhor que a atual, esta solução

é aceita. Entretanto, caso a solução seja pior, ela é aceita dada uma distribuição de probabilidade que depende de um parâmetro de controle análogo à temperatura, que começa alto e decai com o tempo. Dessa maneira, no decorrer da execução do algoritmo, o valor do parâmetro de controle cai e, com isso, a probabilidade de aceitar soluções piores é cada vez menor. Ao fim, ao atingir um critério de parada, que pode ser um determinado número de iterações ou uma temperatura mínima, obtém-se a solução ótima (TALBI, 2009).

Porém, como ressalta Yang (2010), a obtenção do resultado ótimo global depende da correta escolha dos parâmetros do *Simulated Annealing*. Dentre tais parâmetros enumera-se a temperatura inicial e de que forma ela decai, a distribuição de probabilidade usada para definir se uma solução pior é aceita, o número de modificações a cada temperatura e como se dá o critério de parada do algoritmo.

A escolha da temperatura inicial pode se dar por meio de diferentes estratégias, sendo uma delas a de um parâmetro alto o suficiente para, de início, aceitar todas as soluções. Por sua vez, outras abordagens comumente usadas baseiam-se, seja no desvio padrão dos valores objetivos obtidos em testes prévios, seja na obtenção de uma determinada taxa de aceitação de soluções. Já o decaimento da temperatura ao longo do algoritmo pode se dar de diferentes formas, seja linear, geométrica, logarítmica, ou, ainda, a escolha de uma dessas maneiras pode ser de forma adaptativa, dependendo de informações obtidas ao longo da busca (TALBI, 2009).

Quanto a distribuição de probabilidade utilizada para definir a aceitação de uma solução que piora a função objetivo, usualmente utiliza-se a distribuição de Boltzman, onde a probabilidade depende do parâmetro de temperatura e de ΔE , que representa a diferença entre o valor objetivo da solução atual e da solução modificada. A equação para a distribuição de probabilidade de Boltzman é mostrada em (52) (TALBI, 2009).

$$P(\Delta E, T) = e^{\frac{-\Delta E}{T}} \quad (52)$$

O número de movimentações efetuado a cada temperatura é comumente definido conforme duas abordagens: adotar um número estático de movimentações ou um valor variável, que se reduz ao longo da execução do algoritmo. Yang (2010) argumenta que um número muito baixo de movimentações a cada temperatura gera o risco do sistema não se estabilizar e, conseqüentemente, não convergir ao ótimo

global. Em contrapartida, um número muito alto de movimentações implica em um alto custo computacional, que tende a aumentar exponencialmente conforme o tamanho do problema.

Por último, quanto ao critério de parada, este pode se dar a uma temperatura mínima, um número máximo de iterações, ou ainda pode se definir um determinado limite de movimentações que não melhora a solução atual, dentre outras abordagens (TALBI, 2009).

3 MÉTODOS E TÉCNICAS DE PESQUISA

Quanto à abordagem, a presente pesquisa é de característica quantitativa, visto que busca, através da aplicação de métodos de roteirização de transporte em uma empresa de coleta de resíduos, avaliar o impacto de tais técnicas nos custos variáveis da operação, como o combustível.

Quanto ao objetivo, a pesquisa é explicativa. De acordo do Gil (2007) *apud* Gerhardt e Silveira (2009), o tipo de pesquisa explicativa procura identificar os fatores determinantes ou que contribuem para a ocorrência de determinado fenômeno. Dessa maneira, o caráter explicativo da pesquisa justifica-se pela sua busca em avaliar os impactos da aplicação de métodos de roteirização e propor melhorias.

Além disso, a pesquisa, que busca a aplicação de modelos matemáticos para avaliar a aplicação de técnicas de roteirização em uma frota de caminhões usando meios computacionais, utiliza-se, portanto, o método da simulação. Nakano (2012) descreve a pesquisa baseada em simulação como a pesquisa onde descrições matemáticas e simbólicas das relações de causa entre as variáveis de controle e desempenho são analisadas, desenvolvidas e testadas.

Quanto à estrutura, a pesquisa será desenvolvida em quatro etapas: na primeira etapa será feito um referencial teórico a respeito do tema roteirização. A segunda etapa, por sua vez, consiste na coleta de dados junto a empresa para análise, enquanto a terceira consiste na análise em si desses dados. Por último, a quarta etapa consiste em fazer propostas de melhoria a partir dos resultados obtidos através da análise.

4 PROPOSTA DE MODELO MATEMÁTICO

A presente seção busca descrever o problema tratado no trabalho e o método para resolvê-lo, consistindo na modelagem matemática do problema e na proposta de um algoritmo para resolução do modelo.

4.1 Descrição do problema

A lei nº 12.305 de 2 de agosto de 2010, ao condicionar o acesso a recursos da união destinados a serviços e iniciativas relacionados a limpeza urbana à elaboração de um plano municipal de gestão integrada de resíduos sólidos, levou-os a estabelecer limites à responsabilidade do poder público quanto a coleta de resíduos. Dessa maneira, Londrina, assim como outras cidades do norte do Paraná, estabeleceram regras limitantes para que um estabelecimento seja atendido pela coleta pública, sendo então tais estabelecimentos caracterizados como “grandes geradores”. Estas entidades, portanto, são levadas a contratar companhias privadas para o gerenciamento de resíduos sólidos, como a empresa tratada no presente trabalho.

Os grandes geradores de resíduos, ao contratar o serviço de coleta privada, estabelecem junto a empresa contratada uma rotina semanal de coleta, assim como um volume de resíduo a ser recolhido em cada atendimento. Dessa maneira, a empresa que realiza a coleta mantém uma lixeira no local de atendimento ao cliente, com o volume correspondente ao contratado, e visita o ponto de coleta de acordo com a periodicidade acordada.

Cabe ressaltar que a periodicidade definida é um ponto importante do serviço prestado. A geração de resíduos tende a ser constante ao longo da semana, o que obriga o serviço de coleta a também respeitar certa constância, seja para que não ocorra o acúmulo de resíduos, seja para evitar que uma frustrada visita ao cliente pela ausência de carga a ser coletada. Sendo assim, é necessário que as coletas ocorram nos mesmos dias da semana em cada ponto e que, caso existam múltiplas coletas, que estas sejam bem distribuídas ao longo dos dias.

Na operação atual da empresa estudada, a cada equipe de coleta é estabelecida uma rotina semanal de atendimentos, de forma que é designado à equipe quais clientes devem ser atendidos em cada dia da semana e, de forma genérica, a sequência em que tais clientes devem ser visitados, onde a exata

sequência em que postos são coletados é definida pelo próprio motorista. Além disso, por conveniência, os pontos de coleta são atendidos sempre pela mesma equipe, ou seja, cada uma é responsável por um grupo de clientes e atende parte deles a cada dia, recolhendo somente rejeito, e não materiais recicláveis ou resíduos de outras classificações. Tais resíduos são destinados em aterro sanitário próprio da empresa, que fica junto à garagem, ponto de partida dos veículos.

As coletas são realizadas por caminhões compactadores, sendo o volume o fator limitante de carga. Entretanto, tais veículos podem descarregar no aterro sanitário mais de uma vez em cada jornada, limitados a jornada de 8 horas diárias de segunda a sexta-feira e de 4 horas aos sábados, com, no máximo, 2 horas extras.

Como pode ser observado, os custos associados ao transporte são de grande importância à operação e são diretamente dependentes de como as rotas de atendimento aos pontos de coleta são planejadas. Dessa maneira, busca-se, portanto, o melhor jeito de atender tais clientes e em qual ordem, de forma a minimizar os custos de deslocamento, pessoal, manutenção, entre outros.

4.2 Modelo matemático

Esta subseção descreve as variáveis de decisão do modelo matemático, parâmetros, variáveis de apoio, descreve a função objetivo e as restrições do modelo.

4.2.1 Variáveis de decisão

As variáveis de decisão tratadas no modelo proposto são as seguintes:

- x_{ijs}^h : 1 se o veículo h desloca-se do ponto i para o destino j no dia s , caso contrário, $x_{ijk_s}^h = 0$.
- y_s^h : 1 se o veículo h é designado para trabalhar no dia da semana s , caso contrário, $y_{ks}^h = 0$.
- u_{is}^h : 1 se o ponto de coleta i é atendido no dia da semana s pelo veículo h , caso contrário, $u_{iks}^h = 0$.
- z_s^h : número de horas extras para cada veículo h em cada dia da semana s , de forma que corresponde a um número entre 0 e 2.

4.2.2 Parâmetros, variáveis auxiliares e conjuntos.

- m_{ij} : distância a ser percorrida para realizar o deslocamento do ponto i para o destino j , em quilômetros.
- t_{ij} : tempo necessário para tráfego da origem i para o destino j em horas.
- c_d : custos de deslocamento, que consiste na soma dos custos de manutenção, lubrificação, pneus e combustível por quilômetro rodado de um caminhão compactador usado na operação de coleta.
- c_u : custos de utilização de um veículo compactador em um turno de coleta, que consiste na soma da depreciação, remuneração de capital, licenciamento e custos de pessoal referentes a 1 turno de trabalho.
- c_h : custo médio de 1 hora extra.
- w_i^f : 1 se o total de coletas semanais no cliente i corresponde a f , caso contrário, $w_i^f = 0$.
- d_i : volume de resíduo a ser recolhido no ponto i a cada coleta em m^3 .
- Q_{ijs}^h : variável auxiliar para eliminação de sub-rotas que descreve o volume de carga transportada pelo veículo h , no dia da semana s , ao trafegar da origem i para o destino j em m^3 .
- $Q_{máx}$: capacidade máxima de carga volumétrica, em m^3 , do veículo.
- M : um número de enorme ordem de grandeza.
- G_i : tempo necessário para atendimento ao i após chegar até este em horas.
- A : tempo de descarrego do veículo no ponto de descarte em horas.
- J_s : tempo máximo da jornada de trabalho no dia da semana s em horas.
- N : conjunto dos pontos de coleta $\{i \vee i = 0, 1, \dots, N\}$.
- S : conjunto composto pelos dias da semana os quais ocorre a operação de coleta, de forma que 2 corresponde a segunda-feira, 3 corresponde a terça-feira e assim sucessivamente: $\{s \vee s = 2, 3, 4, 5, 6, 7\}$.
- H : conjunto dos veículos disponibilizados para operar $\{h \vee h = 1, 2, \dots, H\}$.

4.2.3 Função objetivo e restrições

A função objetivo busca minimizar os custos operacionais do serviço de coleta e é mostrada em (53).

$$MinZ = c_d \sum_{h=1}^H \sum_{s=2}^7 \sum_{j=0}^N \sum_{i=0}^N x_{ijs}^h m_{ij} + c_u \sum_{h=1}^H \sum_{s=2}^7 y_s^h + c_h \sum_{h=1}^H \sum_{s=2}^7 z_s^h \quad (53)$$

A primeira parte da equação expressa os custos que são em função da distância percorrida, sendo eles o consumo de combustível, manutenção, lubrificação e desgaste de pneu. Na segunda parte são expressos os custos de utilização de um caminhão compactador em um turno de trabalho, sendo eles depreciação, remuneração de capital, licenciamento e custos de pessoal. Na terceira parte da expressão são mostrados os custos com horas extras. A função busca, portanto, minimizar o custo total da operação.

Dentre as restrições a função objetivo, a primeira delas é quanto à demanda, ou seja, o conjunto dos veículos que atende os pontos de coleta ao longo da semana precisa realizar, em cada um deles, um número de visitas igual à quantidade de coletas contratada. Isso é representado pela restrição a seguir, em (54):

$$\sum_{h=1}^H \sum_{s=2}^7 u_{is}^h = \sum_{f=1}^F w_i^f \quad \forall i \in N \quad (54)$$

Outro limite à operação é quanto a jornada de trabalho. As equipes compostas por motorista e coletor trabalham sob o regime de 44 horas semanais, sendo 8 horas diárias de segunda-feira a sexta-feira e 4 horas aos sábados. Entretanto, a legislação trabalhista permite, por dia, um limite de 2 horas extras, que, para este problema, considera-se que é remunerada ao trabalhador. Dessa forma, a expressão (55) mostra que os tempos de deslocamento, tempos de descarga no aterro sanitário e tempos de atendimento no ponto de coleta, somados, e subtraída a quantidade de horas extras realizadas nesse turno, não podem exceder a jornada de trabalho neste dia da semana multiplicado pela variável binária que define se o veículo foi ou não designado para trabalhar em tal dia. Tal dispositivo busca evitar que arcos sejam percorridos por veículos não designados.

$$\sum_{j=0}^N \sum_{i=0}^N x_{ijs}^h t_{ij} + A \sum_{i=0}^N x_{i0s}^h + \sum_{i=0}^N u_{is}^h G_i - z_s^h \leq y_s^h J_s \quad \forall s \in S; h \in H \quad (55)$$

Por sua vez, conforme mostrado em (56), o número de horas extras não pode ultrapassar o limite imposto pela legislação trabalhista, ou seja, 2 horas, multiplicado pela variável de designação desse veículo nesse dia da semana, buscando evitar, de igual maneira, a atribuição de horas extras à equipe de coleta de um veículo não designado.

$$z_s^h \leq y_s^h 2 \forall s \in S; h \in H \quad (56)$$

Para o presente problema, todas as rotas se iniciam e terminam no ponto 0. Dessa forma, em (57) e (58) restringe-se que, se um veículo for designado a uma rota, este irá partir e retornar, ao menos uma vez, ao ponto $i = j = 0$.

$$\sum_{i=0}^N x_{i0s}^h \geq y_s^h \forall s \in S; \forall h \in H \quad (57)$$

$$\sum_{j=0}^N x_{0js}^h \geq y_s^h \forall s \in S; \forall h \in H \quad (58)$$

Por sua vez, a expressão (59) garante que um ponto de coleta seja atendido somente por um veículo designado e, que ocorra, no máximo, uma visita em cada dia da semana. Já na expressão (60) é restrito que os arcos só sejam percorridos por um veículo em atividade no dia correspondente.

$$\sum_{h=1}^H \sum_{f=1}^F u_{is}^h \leq y_s^h \forall i \in N \vee i > 0; \forall s \in S \quad (59)$$

$$\sum_{j=0}^N \sum_{i=0}^N x_{ijs}^h \leq M y_s^h \forall s \in S; \forall h \in H \quad (60)$$

A expressão (61) é uma restrição de eliminação de sub-rotas do tipo fluxo de *commodities* de mercadoria única. Esse modelo de restrição foi proposto para o TSP por Gavish e Graves (1978) e foi adaptado ao presente problema. Entretanto, o dispositivo requer uma variável que representa o fluxo através dos arcos, nesse caso, Q_{ijs}^h . De forma resumida, a restrição mostra que, após um veículo passar por um ponto de coleta, a carga transportada por esse veículo crescerá em igual medida ao volume de resíduo a ser coletado nesse ponto, expresso por $d_i u_{is}^h$, onde d_i mostra o volume de coleta contratado e u_{is}^h se tal volume é ou não recolhido nesse dia da semana.

$$\sum_{j=0}^N Q_{jis}^h - \sum_{j=0}^N Q_{ijs}^h = d_i u_{is}^h \forall i \in N \vee i > 0; \forall s \in S; \forall h \in H \quad (61)$$

Por sua vez, a carga transportada pelo veículo de um ponto de coleta para outro é limitado a capacidade do veículo, expressa por $Q_{máx}$ em (62).

$$\sum_{i=0}^N Q_{ijs}^h \leq Q_{máx} \forall j \in N; \forall s \in S; \forall h \in H \quad (62)$$

Além disso, conforme mencionado anteriormente, o ponto $i = j = 0$ corresponde ao ponto de partida dos veículos e ao ponto de descarte. Dessa forma, todo veículo que parte do ponto 0 parte vazio, como mostrado na expressão (63).

$$\sum_{j=0}^N Q_{0js}^h = 0 \forall i \in N; \forall s \in S; \forall h \in H \quad (63)$$

Por último, uma característica importante do serviço de coleta de resíduos é que, dada a constância da geração de rejeitos, as coletas ao longo da semana precisam ser espaçadas. As restrições (64), (65), (66) e (67) fixam a maneira que deve ocorrer a distribuição dessas coletas pelo período de 6 dias para cada frequência de coleta contratada.

$$w_i^2 \sum_{h=1}^H \sum_{s=2}^4 |u_{is}^h - u_{i(s+3)}^h| = 0 \forall i \in N; \forall h \in H \quad (64)$$

$$w_i^3 \sum_{h=1}^H \sum_{s=2}^6 |u_{is}^h - u_{i(s+1)}^h| = 5w_i^3 \forall i \in N; \forall h \in H \quad (65)$$

$$w_i^4 \sum_{h=1}^H \sum_{s=2}^6 |u_{is}^h - u_{i(s+1)}^h| = 4w_i^4 \forall i \in N; \forall h \in H \quad (66)$$

$$w_i^5 \sum_{h=1}^H \sum_{s=2}^6 |u_{is}^h - u_{i(s+1)}^h| = 2w_i^5 \forall i \in N; \forall h \in H \quad (67)$$

Em (64) mostra-se que, se $w_i^2 = 1$, ou seja, a frequência semanal de coleta no ponto i for igual a 2, a segunda coleta precisa ocorrer 3 dias depois da primeira. Já (65) coloca que, se tal ponto for coletado 3 vezes por semana, essas coletas necessariamente devem ocorrer às segundas, quartas e sextas ou às terças, quintas e sábados. A expressão (66) restringe, para um número de coletas de 4 vezes por semana, um ponto de coleta ou é atendido às segundas, quartas, sextas e sábados ou às segundas, terças, quintas e sábados. Por fim, (67) impõe que, para uma coleta de frequência semanal igual a 5, as 5 coletas não ocorram todas em dias seguidos.

Por último, a expressão (68) coloca que x_{ijs}^h , y_s^h e u_{is}^h são variáveis binária, podendo assumir os valores 0 ou 1. Por sua vez, z_s^h e Q_{jis}^h pertencem ao conjunto dos números racionais maiores ou iguais a 0.

$$x_{ijs}^h \in \{0,1\}, y_s^h \in \{0,1\}, u_{is}^h \in \{0,1\}, z_s^h \in Q \vee z_s^h \geq 0, Q_{jis}^h \in Q \vee Q_{jis}^h \geq 0 \quad (68)$$

4.3 Algoritmo proposto

A presente subseção detalha a construção e o funcionamento do algoritmo *Simulated Annealing* proposto para resolução do problema.

4.3.1 Codificação da solução

Para a aplicação do algoritmo proposto, a solução é codificada como uma lista de 3 dimensões. Dessa maneira, cada vetor solução é composto, em uma primeira dimensão, por 6 vetores s , de forma que cada um destes consiste em um dia da semana. Por sua vez, tais vetores possuem outros h vetores, que correspondem aos veículos que designado a operar nesse dia. Por último, cada vetor h possui dentre seus elementos quais pontos serão visitados por esse veículo e na exata ordem em que tais visitas ocorrerão. A **Figura 1** demonstra como o vetor solução é codificado.

Figura 1 - Exemplo de vetor solução.

Dias da semana:	$s = 2$	$s = 3$
Veículos:	$\overbrace{\hspace{10em}}^{h = 1}$ $\overbrace{\hspace{10em}}^{h = 2}$	$\overbrace{\hspace{10em}}^{h = 1}$ $\overbrace{\hspace{10em}}^{h = 2}$
Pontos de visita:	$\overbrace{0 \ 1 \ 2 \ 0}^{h = 1}$ $\overbrace{0 \ 5 \ 7 \ 9 \ 0}^{h = 2}$	$\overbrace{0 \ 25 \ 3 \ 4 \ 8}^{h = 1}$ $\overbrace{0 \ 6 \ 5 \ 0}^{h = 2}$

Fonte: o autor.

4.3.2 Geração da solução inicial

A geração da solução inicial se dá em três etapas. Em um primeiro momento, um algoritmo sorteia qual será a agenda de coleta dos pontos, ou seja, em quais dias da semana cada um deles será visitado, tendo como base a frequência semanal de coleta contratada, um dos dados de entrada do algoritmo.

Uma vez que as visitas aos pontos de coleta devem estar distribuídas ao longo da semana de maneira uniforme, existe um número limitado de dias de coleta possíveis para cada um, conforme mostrado na seção 4.2.3 e nas expressões (58), (59), (60) e (61) presentes nessa mesma seção. Sendo assim, o algoritmo, de

pseudocódigo mostrado na **Figura 2**, sorteia, dentre essas possibilidades, qual a agenda de atendimento a cada um dos pontos de coleta.

Figura 2 - Pseudocódigo para a definição da agenda de coleta da solução inicial.

```

funcao sortear_agenda(frequencia_de_coleta)
  agenda_de_coleta[2 a 7][0 ate o ponto_de_coleta] = 0
  para cada ponto de coleta i em frequencia_de_coleta:
    se frequencia_de_coleta[i] == 3:
      x = sorteia
      se x == 1:
        dias_de_coleta = [2, 4, 6]
      senao:
        dias_de_coleta = [3, 5, 7]
    se frequencia_de_coleta[i] == 1:
      x = um numero inteiro aleatorio de 2 a 6
      dias_de_coleta = [x]
    se frequencia_de_coleta[i] == 2:
      x = um numero inteiro aleatorio de 2 a 3
      dias_de_coleta = [x, x + 3]
    se frequencia_de_coleta[i] == 4:
      x = aleatoriamente 1 ou 2
      se x == 1:
        dias_de_coleta = [2, 3, 5, 7]
      senao:
        dias_de_coleta = [2, 4, 6, 7]
    se frequencia_de_coleta[i] == 5:
      dias_de_coleta = [2, 3, 4, 5, 6, 7]
      x = um numero inteiro aleatorio de 3 a 6
      deletar elemento igual a x no vetor dias_de_coleta
    se frequencia_de_coleta[i] == 6:
      dias_de_coleta = [2, 3, 4, 5, 6, 7]
  para cada elemento dia no vetor dias_de_coleta:
    agenda_de_coleta[dia][ponto_de_coleta] = 1

  retorne agenda_de_coleta

```

Fonte: o autor.

Cabe ressaltar, entretanto que, para o problema tratado, a jornada de trabalho aos sábados é de somente 4 horas, como explicado na seção 4.1. Dessa maneira, o número de visitas agendadas para cada um dos dias da semana não pode ser uniforme, o que implicaria na necessidade por veículos adicionais durante o sábado quando comparado aos demais dias da semana, de forma a não exceder a duração máxima da jornada de trabalho. Sendo assim, o algoritmo mostrado na Figura 1 evita que as visitas aos pontos de coleta de frequência igual a 1 ou 2 sejam agendadas aos sábados, visando, portanto, reduzir o número de atendimentos nesse dia.

Após a definição da agenda, em uma segunda etapa, os pontos de coleta são atribuídos a cada um dos dias da semana conforme o formato mostrado na seção 4.3.1. Entretanto, ao fazer isso, a função distribui tais pontos entre dois veículos, de forma que tal divisão é feita levando em consideração a longitude média dos pontos.

Sendo assim, os pontos de índice maior que 0 e de longitude menor que a média são atribuídos ao primeiro veículo, assim como os demais, também de índice maior que 0, são atribuídos ao segundo veículo. O pseudocódigo do algoritmo que faz essa distribuição é mostrado na **Figura 3**.

Figura 3 - Pseudocódigo para o algoritmo de agrupamento dos pontos.

```

funcao divide_pontos_de_coleta(coordenadas, agenda_de_coleta):
    soma_longitude = 0
    n = 0
    para cada ponto de coleta i em coordenadas:
        soma_longitude = soma_longitude + coordenadas['Longitude'][i]
        n = n + 1

    logitude_media = soma_longitude / n
    programacao_rotas = []
    para s = 0 até 6:
        rota1 = [0]
        rota2 = [0]
        para j = 1 até n:
            se agenda_de_coleta[s + 2][j] == 1:
                se coordenadas['Longitude'][i] < logitude_media:
                    acrescenta i em rota1
                senao:
                    acrescenta i em rota2
        dia = [rota1, rota2]
        acrescenta dia em programacao_rotas
    retorna programacao_rotas

```

Fonte: o autor.

Por fim, a última etapa para a geração da solução inicial utiliza-se da heurística do vizinho mais próximo para construir uma rota que leva em consideração as restrições de jornada de trabalho máxima, carga máxima, origem e destino. O pseudocódigo para a obtenção final da solução inicial é mostrado na **Figura 4**, enquanto na **Figura 5** é mostrada a estrutura do algoritmo como um fluxograma.

Figura 4 – Pseudocódigo para a geração da solução inicial.

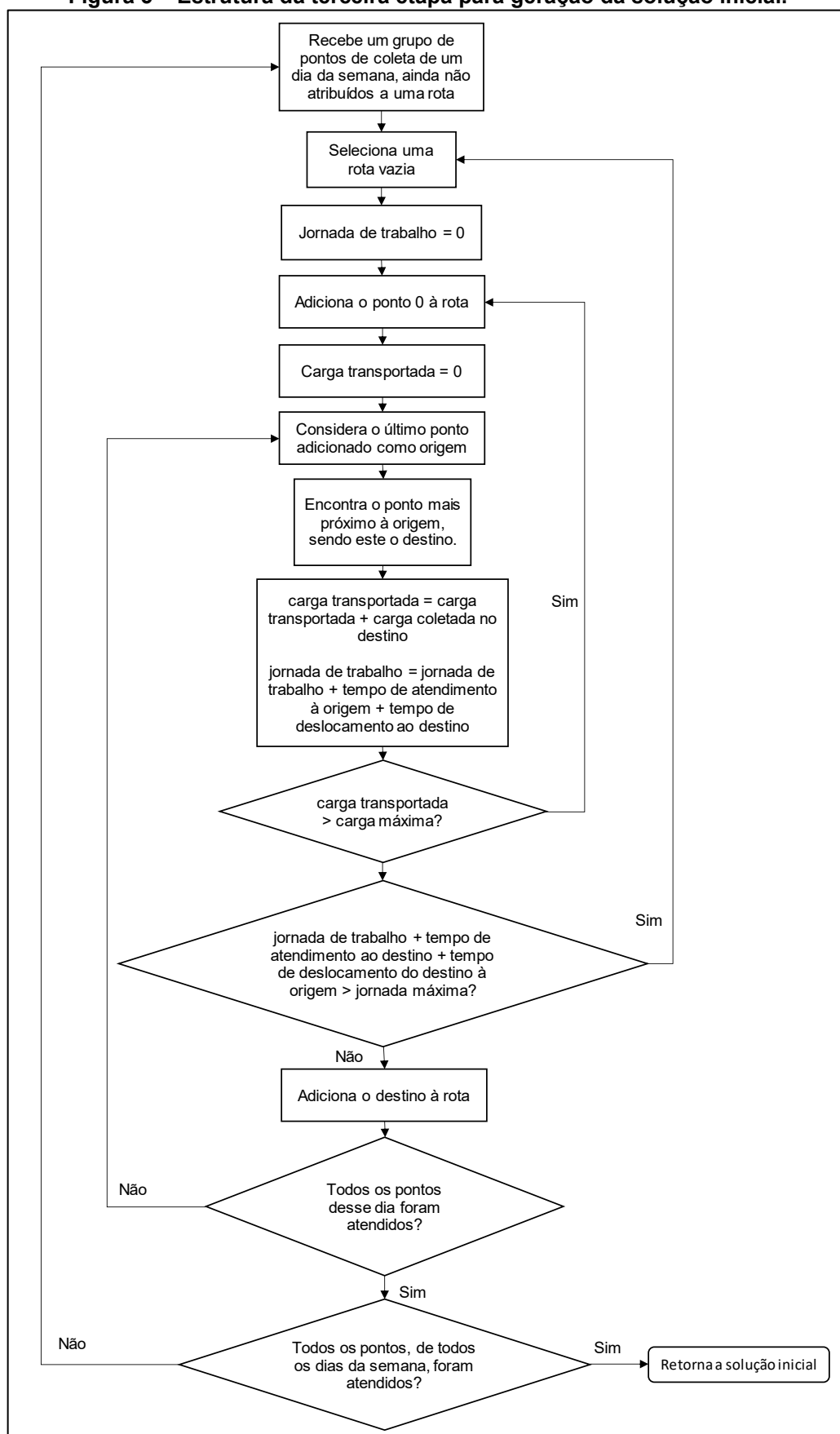
```

funcao solucao_inicial(distancias, tempo, parametros, programacao_rotas):
    solucao_inicial = [[], [], [], [], [], []]
    dia = 0
    para todo dia da semana s em programacao_rotas:
        veiculo = 0
        rotas_do_dia = [[0], [0], [0], [0]]
        para toda rota h do dia da semana s:
            destino = 0
            enquanto nao tamanho do vetor h > 0:
                origem = destino
                se tamanho do vetor(rotas_do_dia[veiculo]) == 1:
                    duracao = 0
                se origem == 0:
                    carga = 0
                destino = vizinho_mais_proximo(origem, h, distancias)
                carga = carga + parametros['carga_no_ponto'][destino]
                insere_o_ponto = Verdadeiro
                se carga >= carga_maxima:
                    carga = 0, destino = 0
                    duracao += duracao_descarga
                    acrescenta o ponto 0 ao vetor rotas_do_dia[veiculo]
                    insere_o_ponto = Falso
                tempo_no_ponto = parametros['Duracao da coleta'][origem]
                duracao += tempos[origem][destino] + tempo_no_ponto
                se duracao + duracao_descarga + tempos[destino][0] + \
                    parametros['Duracao da coleta'][destino] >= jornada_max[dia]:
                    acrescenta o ponto 0 ao vetor rotas_do_dia[veiculo]
                    veiculo += 1, duracao = 0, destino = 0
                    insere_o_ponto = Falso
                se insere_o_ponto = Verdadeiro:
                    acrescenta destino ao vetor rotas_do_dia[veiculo]
                    remove o ponto destino do vetor h
                se último elemento de rotas_do_dia[veiculo] != 0:
                    acrescenta o elemento 0 ao vetor rotas_do_dia[veiculo]
            veiculo += 1
        insere os elementos de rotas_do_dia ao vetor solucao_inicial[dia]
        dia += 1
    solucao_inicial = remove_rotas_vazias(solucao_inicial)
    retorne solucao_inicial

```

Fonte: o autor.

Figura 5 – Estrutura da terceira etapa para geração da solução inicial.



Fonte: o autor.

O algoritmo utilizado na terceira etapa insere o ponto seguinte com base em, dentre os ainda não atendidos, qual está mais próximo do ponto atual analisado, de maneira que o pseudocódigo para o algoritmo de busca do vizinho mais próximo é mostrado na **Figura 6**.

Figura 6 – Pseudocódigo para a obtenção do vizinho mais próximo.

```
funcao vizinho_mais_proximo(origem, possiveis_destino, distancias):
    menor_distancia = Infinito
    para todo ponto i em possiveis_destinos:
        se distancias[origem][destino] <= menor_distancia:
            menor_distancia = distancias[origem][destino]
            destino = possivel_destino
    retorne destino
```

Fonte: o autor.

Caso, após a coleta no vizinho mais próximo, seja atingida a carga máxima, o ponto seguinte a ser inserido é o ponto 0, ou seja, o veículo irá descarregar. Caso o tempo necessário para ir ao próximo ponto, coletá-lo, retornar à origem e descarregar seja maior que a jornada máxima para aquele dia da semana, o veículo retorna à origem, um novo veículo é designado para trabalhar nesse dia e uma nova rota começa a ser preenchida, de forma que a sequência de passos descrita até aqui prossegue com o novo veículo.

Por fim, para a aplicação no algoritmo *Simulated Annealing*, foi usada ainda uma função que gera n determinadas soluções iniciais e retorna a melhor dentre elas. O pseudocódigo para tal função é mostrado **Figura 7**.

Figura 7 – Pseudocódigo para gerar várias soluções iniciais.

```
funcao n_solucoes_iniciais(distancias, tempos, param, \
programacao, N):
    solucao = solucao_inicial(distancias, tempos, param, programacao)
    solucao_custo = objetivo(distancias, tempos, param, solucao)
    para todo n = 0 até N:
        solucao_n = solucao_inicial(distancias, tempos, param, programacao)
        solucao_n_custo = objetivo(distancias, tempos, param, solucao_n)
        se solucao_n_custo < solucao_custo:
            solucao_custo = solucao_n_custo
            solucao = solucao_n
    retorne solucao
```

Fonte: o autor.

4.3.3 Cálculo do custo total das rotas

O pseudocódigo para o cálculo do custo total das rotas é mostrado na **Figura 8**, assim como na **Figura 9** é mostrado o pseudocódigo para obtenção da distância total percorrida e do total de horas extras. Ambos os algoritmos recebem para funcionamento a solução o qual será calculado o custo, uma matriz de distâncias entre

os pontos, uma matriz de tempos de trajeto entre os pontos e, por fim, uma matriz de parâmetros, que contém a frequência semanal de coletas e o volume coletado em cada visita, em cada ponto. As funções percorrem as rotas ponto a ponto e acumulam em variáveis correspondentes a duração da jornada e a distância total percorrida.

Figura 8 – Pseudocódigo para a obtenção do custo total das rotas.

```
funcao objetivo(distancias, tempos, parametros, solucao):
    para todo s em solucao:
        para todo h em s:
            se tamanho(h) > 2:
                n_periodos += 1
    x = calcula_distancia(distancias, tempos, parametros, solucao)
    custo_total = x[0] * custo_por_km + \
    n_periodos * custo_por_período + x[1] * custo_hora_extra
    retorne custo_total
```

Fonte: o autor.

Figura 9 – Pseudocódigo para obtenção das distâncias e horas extras.

```
funcao calcula_distancia(distancias, tempos, parametros, solucao):
    distancia_total = 0
    hora_extra = 0
    jornada = [8, 8, 8, 8, 8, 4]
    para todo s = 0 até 6:
        para todo h = 0 até tamanho(solucao[s]):
            tempototal = 0
            se tamanho(solucao[s][h]) > 1:
                para i = 0 até (tamanho(solucao[s][h]) - 1):
                    j = i + 1
                    origem = solucao[s][h][i]
                    destino = solucao[s][h][j]
                    distancia_total = distancia_total \
                    + distancias[origem][destino]
                    se destino == 0:
                        descarga = tempo_de_descarga
                    senao:
                        descarga = 0
                    tempo = tempos[origem][destino] + \
                    parametros['Duracao da coleta'][origem] + descarga
                    tempototal += tempo
            se tempototal > jornada[s]:
                hora_extra = hora_extra + tempototal - jornada[s]
    retorne distancia_total, hora_extra
```

Fonte: o autor.

4.3.4 Estrutura de vizinhança – realocação entre rotas

A estrutura de vizinhança utilizada baseia-se em algoritmos de realocação entre rotas. Tal tipo de estrutura funciona da seguinte forma: é sorteado um ponto aleatório e tal ponto é removido de sua rota de origem. Em seguida, o algoritmo insere esse ponto em uma das posições possíveis, calcula o custo da solução e desfaz a inserção. Logo após, o ponto é inserido novamente em outra posição, onde é

calculado o custo e assim sucessivamente. Após testar todas as possibilidades, o algoritmo insere o ponto no local onde observou-se o menor custo da solução e retorna tal solução.

Entretanto, a simples aplicação desse método no presente problema geraria muitas soluções ineficazes. Como exemplo, se um ponto que é atendido 3 vezes por semana e está alocado às segundas-feiras, quartas-feiras e sextas-feiras é removido da segunda-feira, a inserção dele em qualquer um dos outros dias da semana geraria uma solução ineficaz, dado que diminuiria o espaçamento entre as coletas.

Como forma de evitar tal cenário, o algoritmo de geração de vizinhança utilizado leva em consideração a frequência semanal de coleta do ponto sorteado e avalia não apenas qual o melhor local para inseri-lo, mas quais os melhores dias da semana para a realização dessas coletas e, nesses dias, qual a melhor posição e rota para inserção do ponto.

Outro ponto a se ressaltar é que a estrutura de vizinhança aqui demonstrada será aplicada em um algoritmo *Simulated Annealing*, de maneira que um dos pontos principais deste tipo de algoritmo é que mesmo uma solução vizinha que piore a função objetivo é aceita dada certa distribuição de probabilidade, isso como uma maneira de ultrapassar ótimos locais e atingir o ótimo global. Dessa forma, para permitir que a estrutura de vizinhança gere também soluções vizinhas piores do que a solução de entrada é estabelecido que as uma ou mais inserções do ponto sorteado serão em locais diferentes dos quais este se encontrava anteriormente. Sem tal restrição, a pior solução que o algoritmo poderia gerar seria uma igual à fornecida como entrada.

O pseudocódigo para o algoritmo de geração de vizinhança é mostrado na Figura 10, assim como os pseudocódigos para obtenção de uma solução vizinha, dado a frequência semanal de coleta do ponto sorteado, são mostrados nas Figuras 11, 12, 13 e 14.

Figura 10 – Pseudocódigo para a estrutura de vizinhança.

```

funcao relocacao(solucao, distancias, tempos, parametros, n):
    custo_melhor_vizinho = Infinito
    para todo i = 0 até n:
        sorteia um ponto aleatório da solucao
        coletas_p_sem = parametros['frequencia'][ponto]
        se coletas_p_sem == 1:
            solucao_vizinha = freq_1(solucao, ponto)
        se coletas_p_sem == 0:
            solucao_vizinha = freq_0(solucao, ponto)
        se coletas_p_sem == 6:
            solucao_vizinha = freq_6(solucao, ponto)
        se coletas_p_sem == 2:
            dias_possiveis = [[0, 3], [1, 4], [2, 5]]
            solucao_vizinha = freq_outras(solucao, ponto, dias_possiveis)
        se coletas_p_sem == 3:
            dias_possiveis = [[0, 2, 4], [1, 3, 5]]
            solucao_vizinha = freq_outras(solucao, ponto, dias_possiveis)
        se coletas_p_sem == 4:
            dias_possiveis = [[0, 2, 4, 5], [0, 1, 3, 5]]
            solucao_vizinha = freq_outras(solucao, ponto, dias_possiveis)
        se coletas_p_sem == 5:
            dias_possiveis = [[0, 2, 3, 4, 5], [0, 1, 3, 4, 5],\
                [0, 1, 2, 4, 5], [0, 1, 2, 3, 5]]
            solucao_vizinha = freq_outras(solucao, ponto, dias_possiveis)
        calcula o custo_da_solucao_vizinha
        se custo_da_solucao_vizinha < custo_melhor_vizinho:
            custo_melhor_vizinho = custo_da_solucao_vizinha
            melhor_vizinho = solucao_vizinha
    retorne melhor_vizinho

```

Fonte: o autor.

Figura 11 – Pseudocódigo para frequência igual a 1.

```

funcao freq_1(solucao, ponto):
    menor_custo = Infinito
    remove o ponto sorteado da solucao
    melhora = Falso
    para todo s = 0 até 6:
        para toda rota r em solucao[s]:
            para todo j = 1 até o tamanho da rota solucao[s][r]:
                se posicao difere da posicao de origem:
                    insere o ponto sorteado\
                    na posição j da rota solucao[s][r]
                    calcula o custo_da_rota
                    se custo_da_rota < menor_custo:
                        se a nova rota atende as restrições:
                            menor_custo = custo_da_rota
                            indices = [s, r, j]
                            melhora = Verdadeiro
                    remove o ponto inserido
    se melhora = Verdadeiro:
        insere o ponto na posição salva no vetor indices
    retorne solucao

```

Fonte: o autor.

Figura 12 – Pseudocódigo para frequência igual a 0.

```

funcao freq_0(solucao, ponto):
  para todo dia = 0 até 6:
    menor_custo = Infinito
    para toda rota em solucao[dia]:
      se o ponto está entre o segundo e o penúltimo ponto da rota:
        remova o ponto
        salva dia, rota e posicao do ponto
      para todo r = 0 até o tamanho do vetor solucao[dia]:
        melhora = Falso
        se dia, rota e posicao j é diferente da de origem
          para todo j = 1 até o tamanho do vetor\
            solucao[dia][r]:
              insere o ponto na posição j da rota solucao[s][r]
              calcula o custo_da_rota
              se custo_da_rota < menor_custo:
                se a nova rota atende as restrições:
                  menor_custo = custo_da_rota
                  indices = [dia, r, j]
                  melhora = Verdadeiro
              remove o ponto inserido
        se melhora = Verdadeiro:
          insere o ponto na posição salva no vetor indices
  retorne solucao

```

Fonte: o autor.

Figura 13 – Pseudocódigo para frequência igual a 6.

```

funcao freq_6(solucao, ponto):
  para todo dia = 0 até 6:
    menor_custo = Infinito
    melhora = Falso
    para toda rota em solucao[dia]:
      se ponto_sorteado está na rota:
        remove o ponto sorteado da rota
        salva dia, rota e posicao de origem
    para todo r = 0 até o tamanho do vetor solucao[dia]:
      para todo j = 1 até o tamanho do vetor solucao[dia][r]:
        se a posicao a inserir difere das de origem:
          insere o ponto sorteado na posição j da\
            rota solucao[s][r]
          calcula o custo_da_rota
          se custo_da_rota < menor_custo:
            se a nova rota atende as restrições:
              menor_custo = custo_da_rota
              indices = [dia, r, j]
              melhora = Verdadeiro
          remove o ponto inserido
    se melhora = Verdadeiro:
      insere o ponto na posição salva no vetor indices
  retorne solucao

```

Fonte: o autor.

Figura 14 – Pseudocódigo para frequências de 2 à 5.

```

funcao freq_outras(solucao, ponto, dias_possiveis):
  para todo dia_de_coleta em solucao:
    para todo rota em dia_de_coleta:
      se ponto está na rota:
        remova o ponto
        salva dia, rota e posicao de origem
  solucao_copia = solucao
  custo_melhor_possibilidade = Infinito
  para todo p em dias_possiveis:
    solucao = solucao_copia
    para todo k em p:
      menor_custo = Infinito
      melhora = Falso
      para todo r = 0 até o tamanho do vetor solucao[k]:
        para todo j = 1 até o tamanho do\
        vetor solucao[k][r]:
          se a posicao a inserir difere das de origem:
            insere o ponto na posição j\
            da rota solucao[k][r]
            calcula o custo_da_rota
            se custo_da_rota < menor_custo:
              se a nova rota atende as restrições:
                menor_custo = custo_da_rota
                indices = [k, r, j]
                melhora = Verdadeiro
            remove o ponto inserido
      se melhora = Verdadeiro:
        insere o ponto na posição salva no vetor indices
        calcula o custo_da_rota
      se custo_da_rota < custo_melhor_possibilidade:
        custo_melhor_possibilidade = custo_da_rota
        melhor_possibilidade = solucao
  retorne melhor_possibilidade

```

Fonte: o autor.

Para uma frequência igual a 1, a solução vizinha é obtida testando a inserção do ponto em qualquer posição, de qualquer rota, em qualquer dia da semana, diferente da de origem. Por sua vez, para uma frequência igual a 6, testa-se em qual posição dentre as rotas de cada um dos dias se minimiza o custo, ou seja, os pontos são mantidos em seus respectivos dias de coleta e, de igual maneira, alocados de forma diferente da original.

Já para frequências de coleta de 2 à 5, existe um número limitado de combinações de dias da semana os quais é possível inserir o ponto. Tais possibilidades de agenda são fornecidas através de um vetor, de maneira que o algoritmo testa cada uma das possibilidades fornecidas e retorna uma solução com o ponto alocado na melhor combinação de dias da semana e na posição, necessariamente diferente da de origem, que minimiza a função objetivo.

Por último, há a situação em que a frequência é igual a 0, que se dá quando o ponto sorteado é a origem. Isso pode acontecer no cenário em que o veículo retorna à origem durante a rota para descarregar e depois volta a atender outros pontos. Quando isso ocorre, o algoritmo remove o ponto e testa a inserção deste em qualquer local, de qualquer rota, em qualquer dia da semana, desde que diferente do local onde o ponto originalmente foi removido.

4.1.1 Estrutura do Simulated Annealing

O algoritmo *Simulated Annealing* utilizado para a resolução do problema recebe, além dos dados do problema, uma temperatura inicial T e uma quantidade inicial de movimentos L a serem realizados. A Figura 15 mostra o pseudocódigo para o algoritmo.

Figura 15 – Pseudocódigo para o *Simulated Annealing*.

```

funcao simulated_annealing(T, L_ini, distancias, tempos, parametros):
    otimo = gera_solucão_inicial(distancias, tempos, parametros, n_sols)
    otimo_fit = objetivo(distancias, tempos, parametros, otimo_local)
    L = L_ini
    enquanto T >= 1:
        i = 0
        maior = -Infinito
        menor = Infinito
        enquanto i <= L:
            solucao_vizinha = relocacao(otimo_local, distancias, \
                tempos, parametros, 3)
            solucao_vizinha_fit = objetivo(distancias, tempos, \
                parametros, solucao_vizinha)
            delta_E = solucao_vizinha_fit - otimo_fit
            se delta_E <= 0:
                otimo_fit = solucao_vizinha_fit
                otimo_local = copy.deepcopy(solucao_vizinha)
            senao:
                p = exponencial(-(delta_E/T))
                rnd = um número aleatório entre 0 e 1
                se rnd <= p:
                    otimo_fit = solucao_vizinha_fit
                    otimo_local = copy.deepcopy(solucao_vizinha)
            i += 1
            se solucao_vizinha_fit > maior:
                maior = solucao_vizinha_fit
            se solucao_vizinha_fit < menor:
                menor = solucao_vizinha_fit
            valor = maior - custos de designação
            F = 1 - math.exp(-1*(maior - menor)/valor)
            L = L_ini + inteiro(L_ini * F)
            T = alfa*T
    retorne otimo

```

Fonte: o autor.

A partir de uma solução inicial, as iterações do algoritmo dão-se da seguinte maneira: é gerada uma solução vizinha a partir da solução atual, de forma que se o custo total dessa solução for menor que o custo da solução atual, tal solução é aceita e torna-se, portanto, a solução atual. Caso isso não ocorra, a solução é aceita com uma probabilidade que depende da temperatura T no momento e de ΔE , que corresponde à diferença entre o custo da solução atual e o custo da solução vizinha analisada. Para o presente trabalho, tal probabilidade segue a distribuição de Boltzmann, mostrada na expressão (52), presente na seção 2.4.

O *Simulated Annealing* baseia-se no processo de recozimento do aço. De forma análoga, duas características do algoritmo são de grande importância: a maneira como se dá o resfriamento, ou seja, de que maneira a temperatura T decai a cada iteração, e como o algoritmo age sob equilíbrio térmico a cada temperatura. Para o presente problema, a temperatura T decai de forma geométrica, conforme a

expressão (69). O algoritmo irá executar iterações até que a temperatura atinja um estado de equilíbrio, definido como $T \leq 1$.

$$T_{n+1} = \alpha T_n \quad (69)$$

Por sua vez, sob uma mesma temperatura, o algoritmo realiza L movimentos, ou seja, gerará L soluções vizinhas e, a cada um desses movimentos, irá avaliar se houve melhora na solução ou se a aceita, ainda que não haja melhora, conforme a distribuição de Boltzmann. Entretanto, para o algoritmo proposto, o valor de L é adaptativo, e dá-se conforme as expressões (70) e (71).

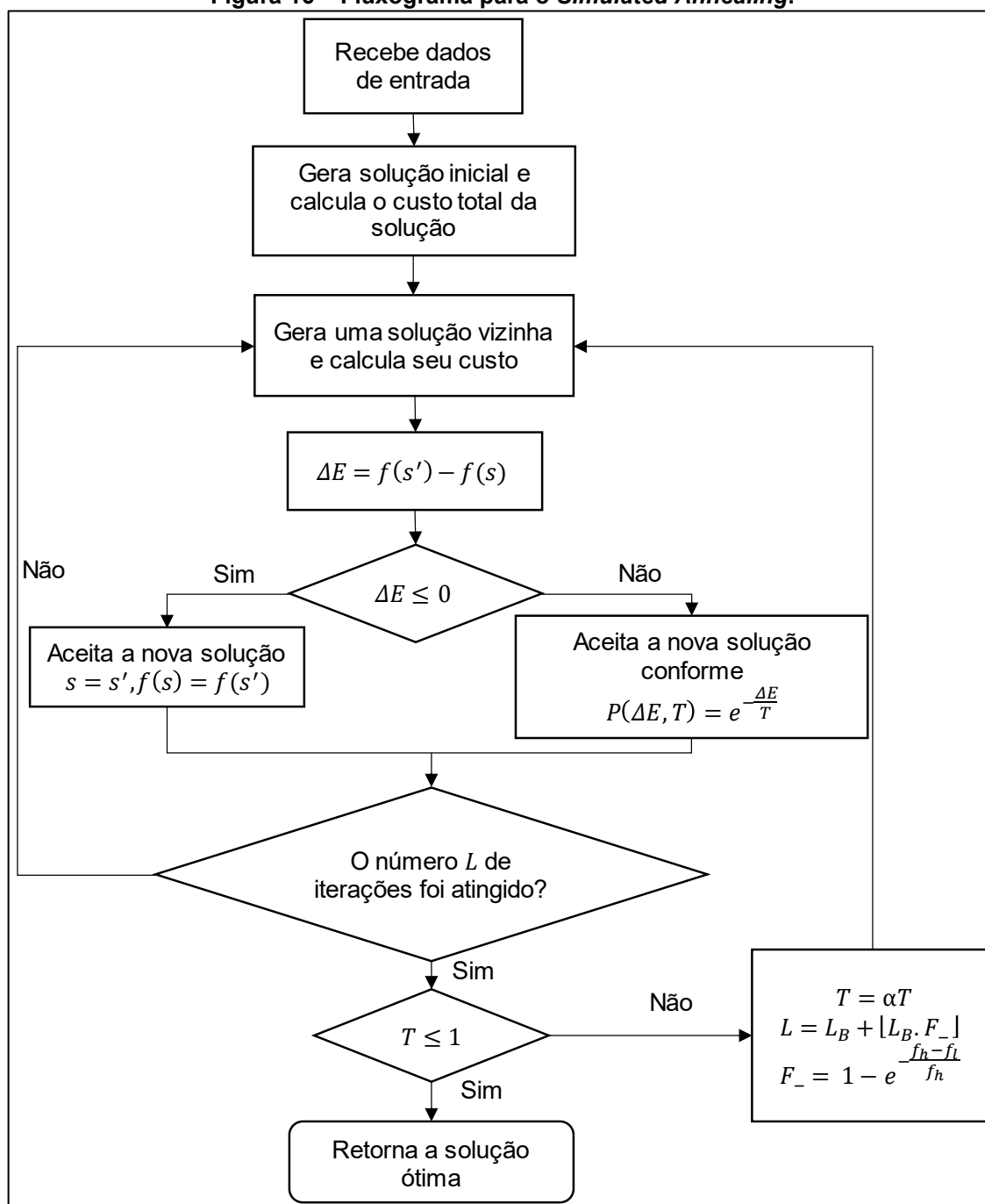
$$L = L_B + \lfloor L_B \cdot F \rfloor \quad (70)$$

$$F = 1 - e^{\frac{-f_h - f_l}{f_h}} \quad (71)$$

Os valores de f_h e f_l correspondem, respectivamente, ao maior e menor valor da função objetivo atingidos na iteração atual. Sendo assim, quanto maior for a diferença entre tais valores, compreende-se que um número maior de iterações é necessário até que se obtenha um estado de equilíbrio, ou seja, os valores objetivo das vizinhanças obtidas variem pouco entre si. Dessa maneira, à medida que a distância entre ambas as grandezas cresce, o número de movimentos realizados na iteração seguinte será maior.

Por fim, o fluxograma mostrado na **Figura 16** resume o funcionamento do *Simulated Annealing*, onde, a cada iteração percorre-se o espaço das soluções simultaneamente ao decair da temperatura T . Ao atingir o equilíbrio térmico, enfim, obtém-se a solução ótima global.

Figura 16 – Fluxograma para o *Simulated Annealing*.



Fonte: o autor.

5 APLICAÇÃO DO MODELO MATEMÁTICO

A presente seção demonstra a aplicação do método proposto. Dessa maneira, são mostrados os dados que serviram de entrada, quais os parâmetros e equipamentos utilizados para execução do algoritmo e os resultados obtidos.

5.1 Dados do problema

Inicialmente foram obtidas junto a empresa informações relacionadas a 356 pontos de coleta, de maneira que estão dispostas no APÊNDICE A -as informações relativas a frequência semanal de coleta, carga recolhida em cada visita e, após a chegada ao ponto, o tempo que o veículo leva para atendê-lo. Além disso, também foram obtidas as coordenadas geográficas dos locais de coleta, de forma que tais informações foram suprimidas por uma questão de confidencialidade.

As coordenadas geográficas fornecidas foram utilizadas para obtenção da distância real da melhor rota entre cada um dos pontos, assim como da duração do trajeto entre estes. Para tal, foi desenvolvido um *script* em *Python* (PYTHON SOFTWARE FOUNDATION, 2022), que se utiliza da *Distance Matrix API*, presente na plataforma para desenvolvedores do Google (2022), de maneira a obter, portanto, duas matrizes, uma contendo as distâncias entre os pontos e a outra os tempos de trajeto, conforme mostrado na **Tabela 1** e na **Tabela 2** para uma amostra de 12 pontos.

Tabela 1 – Distância em quilômetros entre uma amostra de pontos de coleta.

De/para	0	1	2	3	4	5	6	7	8	9	10	11	12
0	999	7,3	6,5	6,5	10,6	7,8	7,7	7,6	8,3	11,8	8,9	11,0	9,0
1	4,7	999	0,9	0,9	4,5	2,6	2,8	2,9	2,6	7,2	4,0	5,6	3,6
2	4,6	0,9	999	0,9	4,9	2,5	2,6	2,8	3,0	7,1	3,8	5,9	4,0
3	4,6	1,0	0,0	999	4,9	2,5	2,7	2,8	3,0	7,1	3,9	6,0	4,0
4	8,1	4,1	3,7	3,7	999	3,3	3,2	3,1	2,7	7,1	4,2	2,9	1,6
5	6,2	2,9	2,1	2,1	3,6	999	0,1	0,3	1,4	4,9	1,5	4,1	2,1
6	6,1	3,3	2,5	2,5	3,5	0,1	999	0,1	1,2	4,7	1,4	3,9	1,9
7	6,0	3,2	2,5	2,4	3,4	0,3	0,1	999	1,2	4,7	1,2	3,9	1,9
8	6,0	3,0	2,5	2,4	3,0	1,2	1,0	1,0	999	4,9	2,0	3,4	1,5
9	5,3	2,5	1,8	1,8	4,4	2,6	2,8	2,9	2,1	999	3,9	5,5	3,5
10	6,9	4,1	3,3	3,3	4,3	1,6	1,4	1,4	2,0	5,5	999	4,7	2,7
11	8,9	5,5	5,1	5,1	2,8	4,0	3,9	3,8	3,3	6,1	4,9	999	2,4
12	6,4	3,1	2,7	2,7	2,2	1,6	1,5	1,4	0,9	5,3	2,5	2,8	999

Fonte: o autor

Tabela 2 – Tempos de deslocamento em horas entre uma amostra de pontos de coleta.

De/para	0	1	2	3	4	5	6	7	8	9	10	11	12
0	999	0,12	0,12	0,12	0,23	0,15	0,14	0,14	0,15	0,10	0,16	0,23	0,16
1	0,17	999	0,04	0,04	0,15	0,12	0,11	0,11	0,12	0,07	0,13	0,19	0,10
2	0,14	0,04	999	0,00	0,14	0,09	0,08	0,08	0,09	0,04	0,10	0,17	0,09
3	0,14	0,04	0,04	999	0,14	0,09	0,08	0,08	0,08	0,04	0,10	0,17	0,09
4	0,26	0,18	0,20	0,20	999	0,15	0,14	0,14	0,13	0,15	0,16	0,10	0,09
5	0,15	0,10	0,10	0,10	0,14	999	0,00	0,01	0,06	0,08	0,05	0,14	0,07
6	0,15	0,11	0,11	0,11	0,14	0,01	999	0,01	0,06	0,09	0,05	0,14	0,07
7	0,15	0,11	0,11	0,11	0,13	0,01	0,00	999	0,05	0,09	0,04	0,13	0,06
8	0,18	0,11	0,12	0,12	0,11	0,07	0,07	0,06	999	0,08	0,08	0,12	0,04
9	0,20	0,16	0,16	0,16	0,19	0,12	0,11	0,11	0,12	999	0,13	0,16	0,13
10	0,18	0,14	0,14	0,14	0,16	0,05	0,05	0,04	0,08	0,12	999	0,16	0,09
11	0,26	0,20	0,22	0,22	0,12	0,15	0,14	0,14	0,13	0,18	0,16	999	0,10
12	0,19	0,14	0,15	0,15	0,08	0,08	0,08	0,08	0,06	0,11	0,09	0,09	999

Fonte: o autor

A *Distance Matrix API* leva em consideração que o trajeto entre os pontos seja percorrido por um carro e em condições normais de trânsito. Entretanto, por meio do sistema de rastreamento da empresa estudada, pôde-se comparar, para uma amostra de trechos, o tempo real de deslocamento dos veículos e tempo obtido via *API*, de forma que não se observou diferença significativa, com o tempo de deslocamento real do caminhão sendo ligeiramente menor.

Quanto à capacidade, os veículos utilizados pela empresa comportam o volume de 19 m³ de resíduo compactado. Após consulta ao manual de operação dos equipamentos, observou-se que estes possuem um fator de compactação de 5:1. Dessa maneira, considerou-se a capacidade de coleta 95 m³ de rejeito.

Por sua vez, quanto à jornada de trabalho, conforme explicado na seção 4.1, de segunda-feira à sexta-feira, a jornada considerada é de 8h, enquanto aos sábados os turnos se estendem por 6h, com até 2 horas extras. Entretanto, no algoritmo para obtenção da solução inicial, adotou-se, considerando já as horas extras, um limite ligeiramente menor, de 9h de jornada de trabalho de segunda a sexta-feira e de 5,5h aos sábados. Tal ajuste possui a intenção de manter uma margem maior para a geração de soluções vizinhas, visto que, se a solução inicial já possuir rotas que utilizam todo o tempo disponível, há uma alta probabilidade da inserção de um novo ponto nessas rotas gerar um trajeto infactível por exceder a jornada máxima.

Por último, obteve-se junto a empresa os seguintes custos operacionais:

- Custo por quilômetro rodado: R\$4,82, que consiste na soma dos custos de manutenção, lubrificação, pneus e combustível para um caminhão compactador 6x2 (3 eixos e tração em 1 deles).
- Custo de utilização por turno de trabalho: R\$501,67, referindo-se a soma dos custos de depreciação, remuneração de capital, licenciamento e encargos salariais referentes a 1 motorista e 1 coletor.
- Custo médio por hora extra: R\$39,84.

5.2 Execução do algoritmo

O algoritmo proposto para solução do problema apresentado foi escrito em linguagem *Python* e executado por meio do interpretador *Python 3.9.2* (PYTHON SOFTWARE FOUNDATION, 2022).

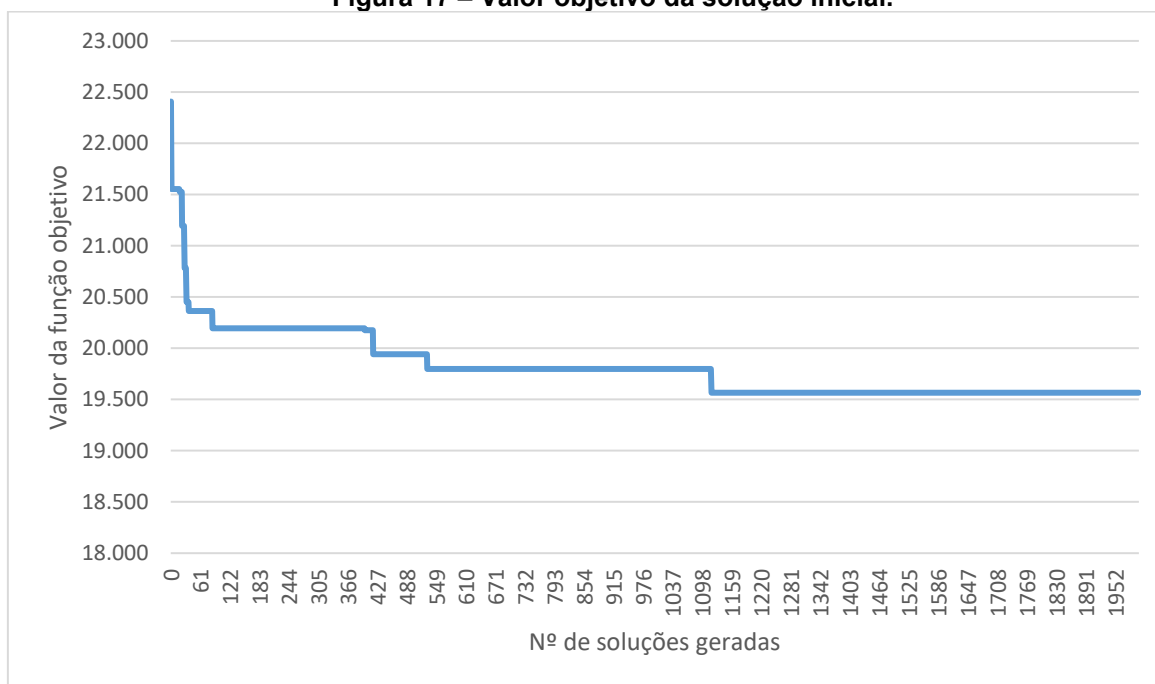
A execução final do programa, assim como os testes para obtenção dos parâmetros do *Simulated Annealing*, foi feita em uma *virtual machine* de um serviço de computação em nuvem. A máquina virtual possui CPU de modelo AMD EPYC 7B13, com 8 núcleos de *clock* igual a 3.050 MHz, 15 Gb de memória RAM e com sistema operacional Debian GNU/Linux 11.

5.2.1 Parâmetros do *Simulated Annealing*

Para obtenção da solução inicial, foram geradas 500 soluções e selecionada a melhor dentre elas. Tal número mostrou-se adequado após a realização de um teste onde foram geradas 2 mil soluções, de maneira a observar o quanto o custo global das rotas obtidas se reduz à medida mais soluções são geradas.

Como pode ser observado na **Figura 17**, há uma redução substancial no valor global das rotas geradas, de 9,1%, com 37 soluções. Após 2 mil soluções geradas, chega-se até 12,7% de redução no valor objetivo quando comparada a primeira solução gerada. Dessa maneira, dado que, para a geração das 2 mil soluções, levou-se 15,6 min, optou-se pela geração de 500 soluções, de maneira a reduzir o custo computacional do algoritmo e, ainda assim, com uma margem de segurança, manter a redução no custo global das rotas obtido a partir de 37 soluções durante o teste.

Figura 17 – Valor objetivo da solução inicial.



Fonte: o autor.

Para o número L_B inicial de movimentações adotou-se o número de 450 movimentações. Para tal, foi desenvolvida uma estrutura baseada no algoritmo *Hill Climbing* de forma a estimar o número de movimentos necessários até atingir-se um ótimo local, obtendo-se a grandeza de 432 movimentações. Tendo isso em vista, utilizou-se $L_B = 450$ com o objetivo de permitir que o *Simulated Annealing* atinja e ultrapasse o ótimo local.

Por sua vez, quanto a temperatura, utilizou-se $T_{inicial} = 1000$. Tendo em vista que o algoritmo aceita uma solução pior que a anterior seguindo a distribuição de Boltzmann (Equação (52)), sob uma temperatura $T = 1000$ e considerando uma variação pequena de $\Delta E = 10$, a probabilidade de uma solução que piore o valor objetivo ser aceita é de aproximadamente 0,99 na primeira iteração, permitindo a ultrapassagem de ótimos locais nos momentos iniciais de execução do algoritmo.

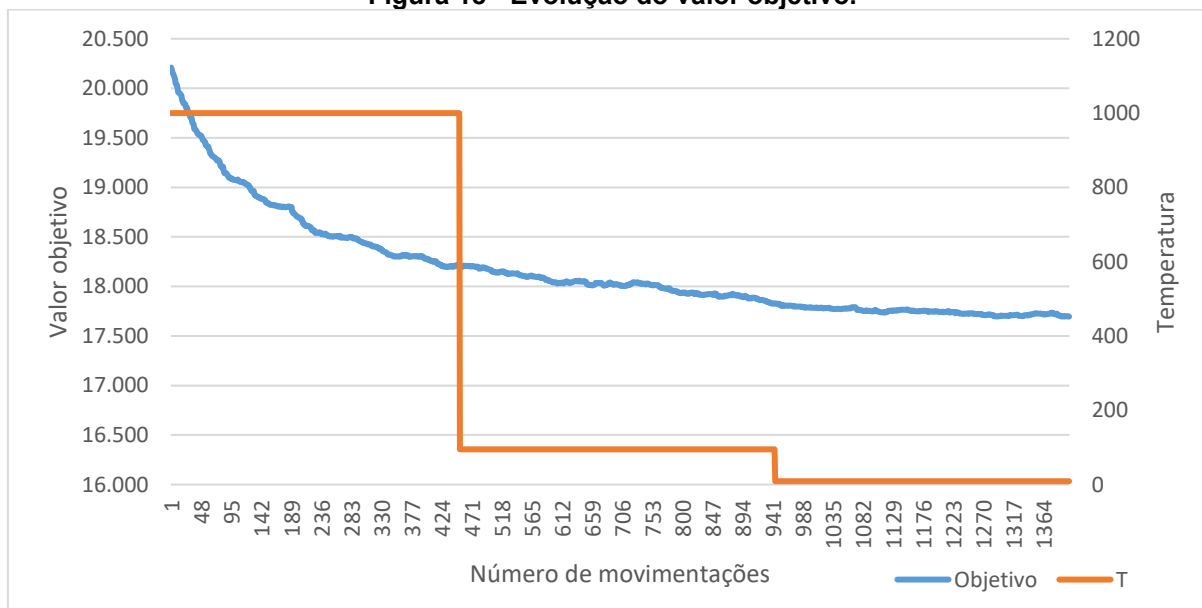
Dessa forma, considerando o critério de parada $T \leq 1$ e que T decai de forma geométrica, como mostrado na Equação (69), o número n de iterações depende de α de acordo com a Equação (72). Sendo assim, o algoritmo foi executado com o valor de $\alpha = 0,095$.

$$n = \left\lceil \frac{\ln\left(\frac{1}{1000}\right)}{\ln(\alpha)} \right\rceil \quad (72)$$

5.3 Resultados

O algoritmo convergiu a um valor objetivo de R\$17.695,86, de maneira que tal valor para a solução inicial utilizada foi de R\$20.209,44, ou seja, uma redução de 12,44%. A evolução da função objetivo a cada movimentação realizada é mostrada na **Figura 18**, sendo esta plotada no eixo esquerdo. Por sua vez, no eixo direito do gráfico, é mostrada a temperatura a cada iteração.

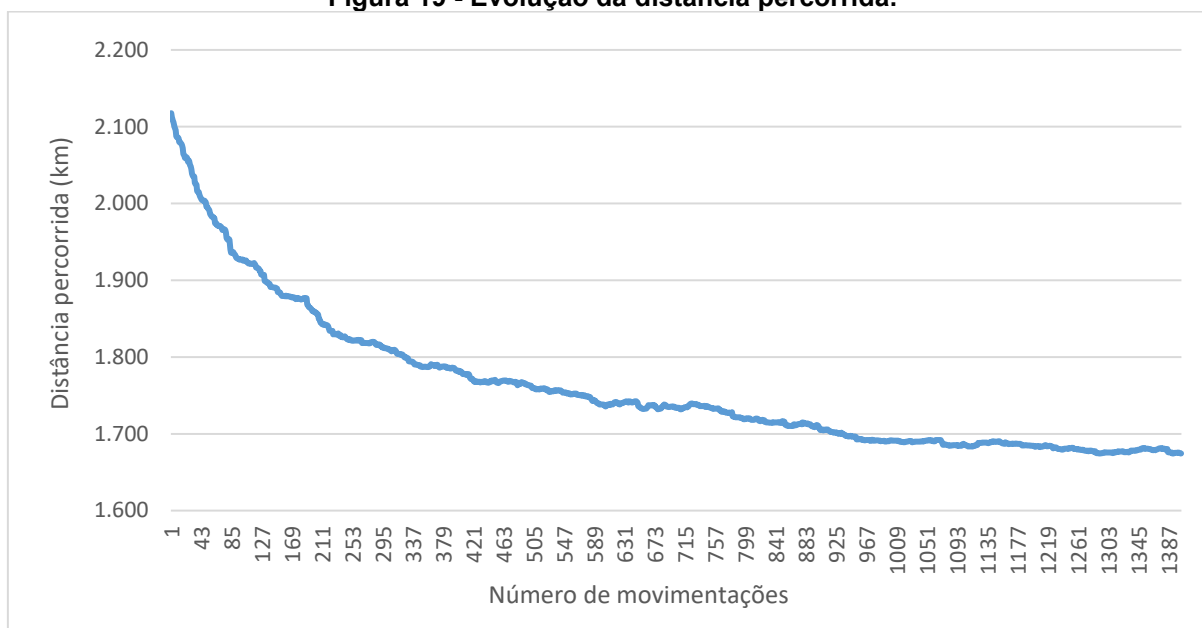
Figura 18 - Evolução do valor objetivo.



Fonte: o autor.

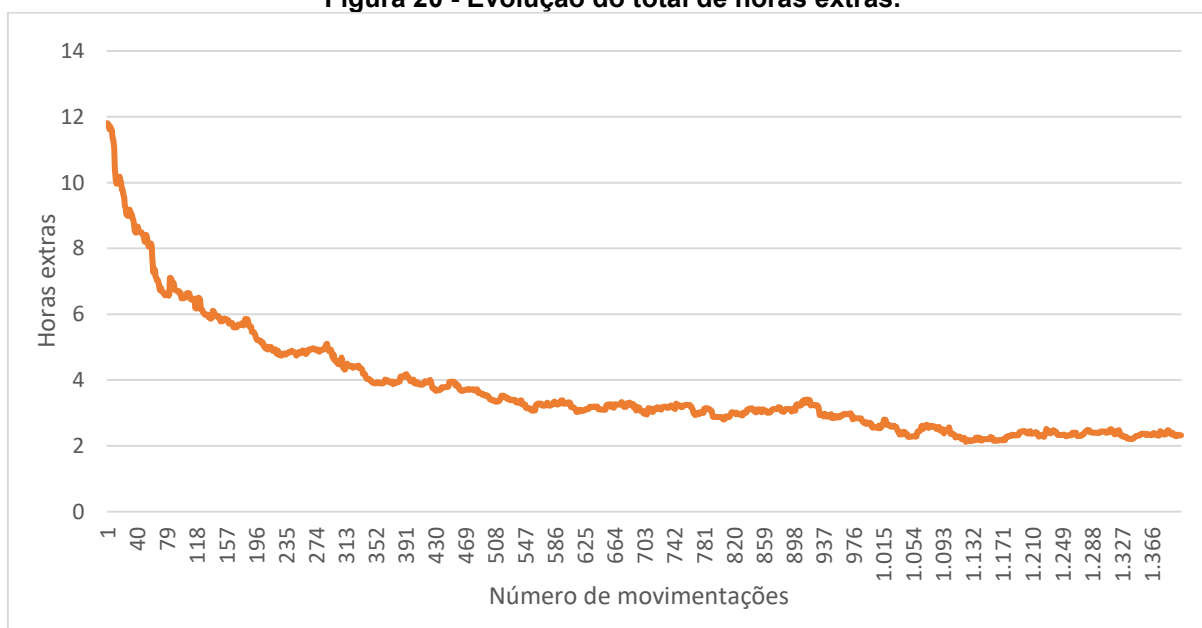
Durante toda a execução do programa, o número de períodos trabalhados foi de 19. Por sua vez, a distância percorrida caiu 20,9% (de 2.117,6 km para 1.674,6 km) e o total de horas extras caiu 80,4% (de 11,8 h para 2,3 h). A evolução da distância ao longo da execução do algoritmo é mostrada na **Figura 19**, em quanto a evolução do total de horas extras é mostrado na **Figura 20**.

Figura 19 - Evolução da distância percorrida.



Fonte: o autor.

Figura 20 - Evolução do total de horas extras.



Fonte: o autor.

Ao total, foram executadas 1.404 movimentações em 3 iterações do *Simulated Annealing*. Tais movimentações levaram, em média, 49,8 segundos, de forma que, ao todo, o algoritmo levou 19,4 horas para ser executado.

As rotas encontradas para cada dia da semana são descritas no APÊNDICE A -.

6 DISCUSSÃO DOS RESULTADOS

A Tabela 3 compara o os custos das rotas encontradas pelo *Simulated Annealing* com as rotas atualmente utilizadas pela empresa, de forma que tais rotas são mostradas no APÊNDICE A -.

Tabela 3 - Resumo dos resultados

Objetivo	Solução Inicial		Operação Real	
	Valor	Variação em relação a solução encontrada	Valor	Variação em relação a solução encontrada
Distâncias	1674,6	-20,90%	1976,3	-15,27%
Horas extras	2,3	-80,40%	5,4	-57,41%
Períodos	19	0%	18	5,56%
Custo global	17695,86	-12,44%	18772,02	-5,73%

Fonte: o autor.

As diferenças entre a solução encontrada pelo *Simulated Annealing* e a operação real ocorrem devido à ausência de planejamento das rotas atuais da empresa. Estas foram feitas levando em consideração a heurística do vizinho mais próximo quando as atividades da empresa se iniciaram e foram sendo adaptadas ao longo do tempo para a adição e remoção de clientes.

A empresa não possui um registro formal, orientação ou documentação que orienta a sequência de visitas aos pontos de coleta que os motoristas devem seguir. Dessa maneira, as rotas atualmente percorridas, mostradas no APÊNDICE A -, foram obtidas por meio do sistema de rastreamento que equipa os veículos, e não mostra o atendimento aos 356 pontos de coleta, porém, são o registro mais próximo possível dentre as informações que a empresa dispõe atualmente.

O não registro do atendimento a um ponto de coleta no sistema de rastreamento utilizado pela empresa pode ocorrer por diferentes fatores, como o cadastro errôneo da área correspondente aos clientes no sistema, o uso eventual de veículos não reestreados para coleta, o efetivo não atendimento a alguns pontos, seja por falta de pagamento, cancelamento, entre outros, ou ainda, é possível que o motorista não atenda ao ponto e não comunique a equipe de logística. Dessa forma, a distância percorrida pela operação real, na prática, é pouco maior do que a mostrada na Tabela 3, porém, o total de horas extras demandado na prática é muito maior que o exibido, e chega a 17,3 horas extras totais.

Além das rotas atualmente utilizadas, ao comparar a solução ótima encontrada pelo algoritmo com a solução inicial fornecida, observa-se que o SA se mostrou eficiente em otimizar a utilização do tempo da jornada de trabalho, assim como das distâncias percorridas. Entretanto, também se observa que o número de períodos trabalhados se manteve igual ao da solução inicial. Com isso o algoritmo mostrou-se incapaz de reduzir a quantidade de períodos trabalhados, o que possivelmente teria um grande impacto na função objetivo. Tal problema deve-se justamente a maneira como o algoritmo explora a vizinhança, dado que, para a redução de um período de trabalho, todos os pontos de um período teriam que ser removidos, o que é improvável dada a maneira como o algoritmo foi construído.

7 CONCLUSÃO

O presente trabalho teve como objetivo a aplicação métodos de roteirização por meio de técnicas computacionais à uma operação de coleta de resíduos sólidos de grandes geradores em uma empresa do Paraná. Para tal, foi necessária uma pesquisa exploratória da literatura em busca das diferentes formas recentemente empregadas para a resolução de problemas análogos ao estudado.

O problema, então, foi modelado, tendo em vista suas especificidades, como a periodicidade no atendimento aos pontos de coleta. Em seguida, obtidas as informações dos locais visitados, como coordenadas geográficas, cargas a serem coletadas e tempos de atendimento, foram calculadas as matrizes de distância e tempos de deslocamento entre os pontos.

Para a obtenção, então, das melhores rotas para atendimento aos pontos de coleta no horizonte de uma semana, foi desenvolvido um algoritmo *Simulated Annealing*, utilizando o método da realocação como estrutura de vizinhança e de um algoritmo baseado na heurística do vizinho mais próximo como forma de obter a solução inicial.

Escrito em *Python* e executado em uma máquina virtual de um serviço de computação em nuvem, o método de resolução empregado mostrou-se eficaz tanto na melhora da solução inicial, reduzindo os custos globais em 20,9%, assim como na obtenção de rotas melhores que as já executadas pela empresa, com custos globais 5,73% menores.

As rotas aqui encontradas podem ser empregadas na coleta de grande gerador da empresa, o que gerará um custo inicial inerente tanto à instrução dos motoristas quanto aos novos trajetos, assim como dos clientes quanto a eventuais modificações na rotina de coleta. Porém, a médio e longo prazo tais custos são compensados, seja pela redução no consumo de combustíveis, redução de custos de horas extras, custos de manutenção, entre outros.

Apesar das melhoras observadas nas rotas encontradas, o algoritmo carece de melhorias em alguns aspectos. O primeiro a ser destacado é o custo computacional empregado, de maneira que mudanças na forma como o algoritmo é construído poderiam torna-lo mais eficiente.

Além disso, ao analisar a evolução do valor da função objetivo ao longo da execução das interações, foi possível observar uma melhora em dois dos três valores

componentes do custo global: distância total e horas extras. Porém, do início ao fim do algoritmo, foram considerados 19 turnos de trabalho. É, portanto, uma sugestão para pesquisas futuras que tratem de problemas semelhantes a este a elaboração de estruturas de exploração de vizinhança ou de um algoritmo em si que seja capaz de convergir para soluções que otimizem o total de turnos trabalhados, podendo assim atingir valores objetivo ainda menores.

REFERÊNCIAS

- ARTHUR, Willian de Azeredo; TROTTA, Fernanda. O Problema de Roteirização de Veículos – Conceito, Estratégias e Métodos de Solução: um exemplo de aplicação de um software comercial no sistema de coleta dos correios. In: CONGRESSO NACIONAL DE EXCELÊNCIA EM GESTÃO, 2., 2004, Rio de Janeiro. **Anais [...]**. Rio de Janeiro: UFF, 2004. p. 112-133. Disponível em: <https://www.inovarse.org/artigos-por-edicoes/II-CNEG-2004/112.pdf>. Acesso em: 12 dez. 2021.
- BAUTISTA, Joaquín; FERNÁNDEZ, Elena; PEREIRA, Jordi. Solving an urban waste collection problem using ants heuristics. **Computers & Operations Research**, [S.L.], v. 35, n. 9, p. 3020-3033, set. 2008. Elsevier BV. <http://dx.doi.org/10.1016/j.cor.2007.01.029>.
- BELFIORE, Patrícia; FÁVERO, Luiz Paulo. **Pesquisa Operacional Para Cursos de Engenharia**. Rio de Janeiro: Elsevier Editora Ltda., 2013. Disponível em: [https://integrada.minhabiblioteca.com.br/reader/books/9788595155626/epubcfi/6/10\[%3Bvnd.vst.idref%3DCopyright\]/4/2/16/3:27\[Ltd%2Ca.\]](https://integrada.minhabiblioteca.com.br/reader/books/9788595155626/epubcfi/6/10[%3Bvnd.vst.idref%3DCopyright]/4/2/16/3:27[Ltd%2Ca.]). Acesso em: 12 dez. 2021.
- BRASIL. **Lei nº 12.305, de 2 de agosto de 2010**. Disponível em: http://www.planalto.gov.br/ccivil_03/_ato2007-2010/2010/lei/l12305.htm. Acesso em: 10 jun. 2022.
- GAVISH, Bezalel; GRAVES, Stephen C.. The Travelling Salesman Problem and Related Problems. **Operations Research Center**, Cambridge, jul. 1978. Disponível em: <http://hdl.handle.net/1721.1/5363>. Acesso em: 03 nov. 2022.
- GERHARDT, Tatiana Engel; SILVEIRA, Denise Tolfo. Métodos de pesquisa. Porto Alegre: Editora da Ufrgs, 2009. 120 p.
- GHIANI, Gianpaolo *et al.* **Real-time vehicle routing: solution concepts, algorithms and parallel computing strategies**. European Journal Of Operational Research, Londres, v. 155, n. 1, p. 1-11, 16 nov. 2003. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0377221702009153>. Acesso em: 12 dez. 2021.
- GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization and Machine Learning**. Addison-Wesley: New York, 1989.
- GOOGLE (Mountain View, Califórnia, Estados Unidos). **Distance Matrix API**. V1.21.1. [S. l.], 11 nov. 2022. Disponível em: <https://developers.google.com/maps/documentation/distance-matrix>. Acesso em: 12 nov. 2022.
- KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P.. Optimization by Simulated Annealing. **Science**, [S.L.], v. 220, n. 4598, p. 671-680, 13 maio 1983. American Association for the Advancement of Science (AAAS). <http://dx.doi.org/10.1126/science.220.4598.671>.
- MIRJALILI, Seyedehzahra *et al.* Whale Optimization Algorithm: Theory, Literature Review, and Application in Designing Photonic Crystal Filters. In: MIRJALILI,

Seyedehzahra *et al.* **Nature-Inspired Optimizers**. Berlin: Springer, 2020. p. 219-238.

MOFID-NAKHAEE, Elham; BARZINPOUR, Farnaz. A multi-compartment capacitated arc routing problem with intermediate facilities for solid waste collection using hybrid adaptive large neighborhood search and whale algorithm. **Waste Management & Research**, [S.L.], v. 37, n. 1, p. 38-47, 15 out. 2018. SAGE Publications. <http://dx.doi.org/10.1177/0734242x18801186>.

NAKANO, D. **Métodos de pesquisa adotados na Engenharia de Produção e Gestão de Operações**. In: CAUCHICK, P. M. (Coord.). Metodologia de pesquisa em Engenharia de Produção e Gestão de Operações. Rio de Janeiro: Elsevier, 2012.

PISINGER, David; ROPKE, Stefan. A general heuristic for vehicle routing problems. **Computers & Operations Research**, [S.L.], v. 34, n. 8, p. 2403-2435, ago. 2007. Elsevier BV. <http://dx.doi.org/10.1016/j.cor.2005.09.012>.

POLETTTO, Luíza Denardin. **Metodologia de definição de grandes geradores de resíduos sólidos urbanos para o município de Florianópolis**. 2018. 114 f. TCC (Graduação) - Curso de Engenharia Sanitária e Ambiental, Universidade Federal de Santa Catarina, Florianópolis, 2018. Disponível em: <https://repositorio.ufsc.br/bitstream/handle/123456789/195656/TCC%20II%20Lu%c3%adza%20Denardin%20Poletto.pdf?sequence=1&isAllowed=y>. Acesso em: 10 jun. 2022.

PYTHON SOFTWARE FOUNDATION (Beaverton, OR). **Python 3**. 3.11.0. Estados Unidos, 24 out. 2022. Disponível em: <https://www.python.org/>. Acesso em: 12 nov. 2022.

QIAO, Qingqing *et al.* Optimization of a Capacitated Vehicle Routing Problem for Sustainable Municipal Solid Waste Collection Management Using the PSO-TS Algorithm. **International Journal Of Environmental Research And Public Health**, [S.L.], v. 17, n. 6, p. 2163, 24 mar. 2020. MDPI AG. <http://dx.doi.org/10.3390/ijerph17062163>.

SOCIEDADE BRASILEIRA DE PESQUISA OPERACIONAL. **O que é Pesquisa Operacional?** 2021. Disponível em: <https://www.sobrapo.org.br/o-que-e-pesquisa-operacional>. Acesso em: 12 dez. 2021.

TALBI, El-Ghazali. Single-Solution Based Metaheuristics. In: TALBI, El-Ghazali. **Metaheuristics: From Design to Implementation**. Hoboken: John Wiley And Sons, 2009. p. 87-189.

TIRKOLAEI, Erfan Babaei *et al.* Developing an applied algorithm for multi-trip vehicle routing problem with time windows in urban waste collection: a case study. **Waste Management & Research: The Journal for a Sustainable Circular Economy**, [S.L.], v. 37, n. 1, p. 4-13, jan. 2019. SAGE Publications. <http://dx.doi.org/10.1177/0734242x18807001>.

TIRKOLAEI, Erfan Babaei; MAHDAVI, Iraj; ESFAHANI, Mir Mehdi Seyyed. A robust periodic capacitated arc routing problem for urban waste collection considering drivers and crew's working time. **Waste Management**, [S.L.], v. 76, p. 138-146, jun. 2018. Elsevier BV. <http://dx.doi.org/10.1016/j.wasman.2018.03.015>.

WU, Hailin; TAO, Fengming; YANG, Bo. Optimization of Vehicle Routing for Waste Collection and Transportation. **International Journal Of Environmental Research And Public Health**, [S.L.], v. 17, n. 14, p. 4963, 9 jul. 2020. MDPI AG. <http://dx.doi.org/10.3390/ijerph17144963>.

YANG, Xin-She. Simulated Annealing. In: YANG, Xin-She. **Engineering Optimization: An Introduction with Metaheuristic Applications**. Cambridge: John Wiley And Sons, 2010. p. 181-188. Disponível em: <https://onlinelibrary.wiley.com/doi/10.1002/9780470640425.ch12>. Acesso em: 28 nov. 2022.

ZEINELDIN, H.H.; EL-SAADANY, E.F.; SALAMA, M.M.A.. Optimal coordination of overcurrent relays using a modified particle swarm optimization. **Electric Power Systems Research**, [S.L.], v. 76, n. 11, p. 988-995, jul. 2006. Elsevier BV. <http://dx.doi.org/10.1016/j.epsr.2005.12.001>

APÊNDICE A - Dados dos pontos de coleta

Tabela A-1 – Dados do problema.

Ponto	Frequência semanal de coleta (coletas/semana)	Carga recolhida por coleta (m ³)	Tempo de atendimento no ponto de coleta (h)
1	3	0,240	0,02373
2	3	0,480	0,04233
3	3	0,480	0,05100
4	3	3,600	0,05405
5	3	0,240	0,05081
6	2	2,400	0,05272
7	3	0,480	0,04210
8	1	1,200	0,04210
9	6	1,200	0,07348
10	3	1,200	0,06144
11	3	0,480	0,02092
12	2	0,240	0,03838
13	3	1,200	0,07864
14	6	1,200	0,04494
15	3	0,480	0,03806
16	3	0,480	0,04688
17	4	2,400	0,07597
18	3	0,480	0,02955
19	3	0,480	0,03608
20	2	1,200	0,09068
21	2	0,480	0,01675
22	3	2,400	0,24430
23	2	1,200	0,09068
24	3	0,240	0,02876
25	1	0,480	0,08086
26	3	0,240	0,03063
27	3	1,200	0,04176
28	3	0,480	0,04246
29	3	0,480	0,09068
30	4	0,480	0,01653
31	1	0,240	0,03449
32	1	0,240	0,09068
33	1	0,480	0,04147
34	2	0,240	0,07590
35	2	1,200	0,09068
36	3	0,480	0,09068
37	2	1,200	0,09447
38	6	1,200	0,10082
39	3	0,480	0,04303
40	4	1,200	0,10341
41	3	0,480	0,04889
42	3	0,240	0,01042

Ponto	Frequência semanal de coleta (coletas/semana)	Carga recolhida por coleta (m³)	Tempo de atendimento no ponto de coleta (h)
43	3	1,200	0,04699
44	3	1,200	0,09517
45	2	0,240	0,05518
46	3	1,200	0,10667
47	2	2,400	0,09935
48	3	1,200	0,06732
49	1	0,480	0,09068
50	1	0,240	0,00861
51	2	2,400	0,09068
52	3	7,000	0,16370
53	2	1,200	0,05026
54	3	1,200	0,04891
55	3	0,480	0,09068
56	2	0,480	0,01125
57	3	1,200	0,05653
58	3	0,480	0,04788
59	3	0,960	0,11048
60	1	0,480	0,09068
61	3	0,240	0,02125
62	3	1,200	0,09068
63	3	0,240	0,09068
64	1	1,200	0,05278
65	2	0,480	0,05525
66	6	0,240	0,04417
67	2	1,200	0,01632
68	3	0,480	0,09068
69	6	0,480	0,04303
70	2	0,240	0,01111
71	2	1,200	0,06316
72	3	1,200	0,09068
73	3	0,480	0,09068
74	3	0,480	0,06028
75	1	7,000	0,00910
76	3	1,200	0,15239
77	2	1,200	0,04403
78	3	1,200	0,09068
79	1	1,200	0,07143
80	2	1,200	0,03799
81	3	0,240	0,02056
82	3	1,200	0,03458
83	5	0,960	0,00917
84	6	0,240	0,00917
85	6	0,240	0,00917

Ponto	Frequência semanal de coleta (coletas/semana)	Carga recolhida por coleta (m³)	Tempo de atendimento no ponto de coleta (h)
86	3	0,480	0,03747
87	3	1,200	0,04747
88	6	0,480	0,01963
89	3	1,200	0,11757
90	5	0,480	0,04764
91	3	0,480	0,03002
92	3	0,720	0,08413
93	2	1,200	0,05823
94	3	0,480	0,02403
95	6	1,200	0,06188
96	2	1,200	0,03056
97	3	1,200	0,06250
98	6	4,800	0,13418
99	3	0,240	0,01910
100	3	0,480	0,02264
101	1	0,240	0,09068
102	6	0,240	0,03241
103	3	0,720	0,04024
104	2	1,200	0,03813
105	6	0,240	0,01583
106	3	0,480	0,02250
107	1	1,200	0,09068
108	3	1,200	0,07130
109	3	0,480	0,09068
110	3	0,960	0,07729
111	2	0,240	0,03153
112	3	0,480	0,14104
113	2	0,240	0,02014
114	6	0,240	0,09068
115	1	0,240	0,09068
116	3	0,480	0,02565
117	4	0,480	0,02948
118	3	0,480	0,16111
119	1	0,240	0,11840
120	4	0,720	0,09068
121	2	0,480	0,08528
122	3	0,480	0,00694
123	6	0,240	0,00975
124	2	1,200	0,03806
125	2	1,200	0,06594
126	2	0,240	0,09068
127	3	0,240	0,02264
128	6	2,400	0,09318

Ponto	Frequência semanal de coleta (coletas/semana)	Carga recolhida por coleta (m³)	Tempo de atendimento no ponto de coleta (h)
129	6	0,240	0,01491
130	3	0,960	0,08758
131	3	0,480	0,10611
132	3	0,240	0,09068
133	2	1,200	0,01190
134	3	0,720	0,05722
135	3	1,200	0,09068
136	3	0,240	0,09068
137	3	0,240	0,04101
138	6	1,200	0,21125
139	6	1,200	0,07368
140	4	1,200	0,00556
141	4	1,200	0,06539
142	2	1,200	0,04840
143	3	0,240	0,04172
144	6	0,960	0,13508
145	3	0,480	0,09068
146	3	1,200	0,09068
147	3	0,240	0,09068
148	3	0,720	0,04075
149	6	1,200	0,03694
150	3	0,240	0,08689
151	3	0,720	0,09068
152	3	0,960	0,04681
153	6	0,240	0,01306
154	1	0,240	0,09068
155	3	0,240	0,03828
156	2	1,200	0,09068
157	3	0,480	0,09556
158	3	2,400	0,06439
159	3	0,720	0,04576
160	3	1,920	0,02993
161	3	0,960	0,08153
162	3	0,240	0,06636
163	3	0,240	0,03375
164	3	2,400	0,09841
165	3	0,960	0,05478
166	5	0,480	0,00944
167	2	0,960	0,58319
168	5	1,000	0,05894
169	6	0,960	0,03663
170	1	0,480	0,03069
171	6	0,480	0,08015

Ponto	Frequência semanal de coleta (coletas/semana)	Carga recolhida por coleta (m³)	Tempo de atendimento no ponto de coleta (h)
172	2	0,480	0,04440
173	3	0,240	0,09068
174	3	0,240	0,09068
175	1	1,200	0,07278
176	3	0,240	0,09068
177	3	1,200	0,10822
178	3	1,200	0,09368
179	3	0,240	0,02898
180	3	1,200	0,01940
181	3	2,400	0,26394
182	6	2,400	0,04913
183	6	1,200	0,03150
184	3	2,400	0,09184
185	2	0,240	0,00993
186	3	0,960	0,13133
187	3	1,200	0,07252
188	3	0,480	0,09068
189	5	0,720	0,02397
190	3	2,400	0,09068
191	2	1,200	0,09068
192	1	1,200	0,09068
193	3	0,960	0,08285
194	2	1,200	0,09068
195	2	0,480	0,09068
196	4	0,720	0,10080
197	5	0,240	0,01761
198	3	0,240	0,01076
199	2	0,240	0,07057
200	3	1,200	0,00926
201	3	1,200	0,09068
202	3	1,200	0,09213
203	3	0,960	0,05977
204	1	2,400	0,03952
205	3	0,240	0,02627
206	2	1,200	0,05191
207	1	6,000	0,61539
208	3	0,720	0,04317
209	1	0,720	0,09068
210	2	0,480	0,09068
211	1	1,200	0,02889
212	3	1,200	0,09068
213	3	2,400	0,07528
214	1	0,480	0,04340

Ponto	Frequência semanal de coleta (coletas/semana)	Carga recolhida por coleta (m³)	Tempo de atendimento no ponto de coleta (h)
215	4	0,240	0,02233
216	2	0,240	0,05258
217	3	0,240	0,07590
218	3	1,200	0,12270
219	3	0,480	0,02935
220	3	0,480	0,01129
221	3	0,480	0,02948
222	3	0,480	0,04804
223	3	0,480	0,09068
224	2	0,240	0,09068
225	6	1,200	0,02571
226	3	0,240	0,02125
227	3	0,240	0,04656
228	3	0,240	0,02635
229	3	2,400	0,06231
230	3	0,240	0,08417
231	3	1,200	0,02669
232	2	0,480	0,09068
233	2	0,240	0,01792
234	2	0,240	0,09068
235	3	0,480	0,09068
236	1	2,400	0,10704
237	6	1,200	0,05996
238	3	0,240	0,03847
239	1	1,200	0,09068
240	2	0,240	0,03109
241	3	0,480	0,04340
242	3	0,240	0,06253
243	3	1,200	0,04944
244	3	0,480	0,09068
245	3	0,240	0,02633
246	1	1,200	0,37028
247	2	0,480	0,05899
248	2	0,240	0,09144
249	1	0,240	0,09068
250	2	1,200	0,07769
251	2	1,200	0,02500
252	3	0,240	0,09068
253	6	1,200	0,09068
254	1	0,240	0,01111
255	1	1,200	0,09068
256	5	1,200	0,09068
257	6	1,200	0,09467

Ponto	Frequência semanal de coleta (coletas/semana)	Carga recolhida por coleta (m³)	Tempo de atendimento no ponto de coleta (h)
258	2	1,200	0,09406
259	1	0,480	0,09068
260	3	0,240	0,06736
261	3	0,960	0,05116
262	3	1,200	0,09068
263	1	0,480	0,05549
264	4	2,400	0,11016
265	2	0,480	0,07937
266	2	1,200	0,06104
267	1	0,960	0,23311
268	3	0,240	0,09068
269	3	1,200	0,12832
270	2	0,960	0,09068
271	3	3,000	0,09068
272	6	2,400	0,08467
273	3	2,400	0,05597
274	2	1,440	0,11417
275	3	0,240	0,09068
276	6	1,200	0,11980
277	3	0,720	0,09068
278	3	2,400	0,06577
279	3	1,200	0,05091
280	2	1,200	0,01931
281	3	0,480	0,03403
282	3	0,240	0,00889
283	3	1,200	0,04917
284	3	2,160	0,08577
285	1	0,240	0,09068
286	2	1,200	0,02942
287	1	1,200	0,09208
288	2	0,480	0,07676
289	5	0,240	0,09068
290	2	0,240	0,02852
291	1	0,480	0,04089
292	1	0,480	0,01889
293	1	1,200	0,06118
294	3	0,480	0,04190
295	2	2,400	0,06075
296	3	1,200	0,01964
297	3	0,480	0,01964
298	1	0,240	0,03690
299	2	1,200	0,03538
300	3	0,720	0,05323

Ponto	Frequência semanal de coleta (coletas/semana)	Carga recolhida por coleta (m³)	Tempo de atendimento no ponto de coleta (h)
301	1	0,480	0,05549
302	2	1,200	0,03774
303	2	0,240	0,09068
304	2	0,480	0,09796
305	2	3,000	0,06856
306	1	0,480	0,02028
307	3	1,200	0,05361
308	2	2,400	0,08119
309	3	0,240	0,09068
310	1	1,200	0,03216
311	3	0,240	0,00611
312	3	1,200	0,07520
313	3	0,480	0,01740
314	1	0,480	0,02263
315	3	1,200	0,09068
316	3	0,240	0,02712
317	3	1,200	0,02092
318	3	0,240	0,06620
319	1	0,240	0,09068
320	3	0,240	0,01352
321	3	0,480	0,02948
322	3	2,400	0,24464
323	3	0,240	0,09068
324	3	0,960	0,37963
325	2	1,200	0,06192
326	3	2,400	0,04786
327	3	0,240	0,02556
328	3	0,240	0,09068
329	3	0,240	0,09068
330	1	0,480	0,09549
331	3	0,240	0,01081
332	2	1,200	0,05851
333	2	0,480	0,05894
334	3	0,240	0,09068
335	3	1,200	0,09068
336	3	2,400	0,09068
337	3	0,720	0,09068
338	4	0,480	0,08585
339	1	0,240	0,04375
340	3	3,600	0,09068
341	2	0,240	0,09068
342	2	2,000	0,16785
343	1	1,200	0,04870

Ponto	Frequência semanal de coleta (coletas/semana)	Carga recolhida por coleta (m³)	Tempo de atendimento no ponto de coleta (h)
344	2	1,200	0,08606
345	1	0,240	0,06383
346	3	1,200	0,04078
347	3	0,960	0,03316
348	3	0,240	0,58319
349	3	1,200	0,04075
350	3	1,200	0,02382
351	2	1,200	0,08006
352	2	3,480	0,09068
353	3	0,240	0,11389
354	2	0,720	0,09310
355	1	1,200	0,15412
356	3	0,240	0,03551

Fonte: o autor.

APÊNDICE B - Rotas obtidas pelo SA

Tabela B-1 – Rotas de segunda-feira a quarta-feira

Dia da semana	2	2	2	3	3	3	4	4	4
Ordem\Rota	1	2	3	1	2	3	1	2	3
1	0	0	0	0	0	0	0	0	0
2	21	9	1	8	9	24	70	9	13
3	36	144	2	12	324	56	106	40	4
4	40	149	3	35	320	67	136	36	11
5	50	158	6	117	273	39	149	61	16
6	70	159	13	149	252	42	88	73	29
7	103	160	4	147	253	41	92	83	22
8	113	134	11	116	232	43	112	100	28
9	106	148	16	111	248	52	130	115	30
10	136	141	29	137	199	55	158	82	32
11	112	129	30	135	177	54	159	76	34
12	90	85	31	138	168	62	160	114	79
13	87	66	71	139	144	68	171	127	81
14	86	69	81	156	171	38	188	128	91
15	138	84	91	162	163	51	236	148	110
16	139	109	104	182	202	193	228	169	181
17	143	127	93	195	183	72	219	153	231
18	105	128	110	172	258	60	222	140	229
19	132	126	181	169	309	65	216	129	242
20	95	173	167	153	316	58	212	141	205
21	74	165	205	122	353	224	203	138	220
22	76	215	220	118	346	271	204	139	208
23	82	222	242	129	348	332	211	143	230
24	100	217	229	97	354	352	221	85	234
25	146	219	231	85	355	342	225	84	238
26	194	168	250	78	263	344	264	109	254
27	201	171	208	84	272	333	269	105	279
28	203	121	230	105	290	280	284	132	312
29	189	80	238	107	295	275	268	150	315
30	197	88	251	95	301	246	256	151	321
31	185	228	270	108	302	223	262	123	323
32	182	227	279	114	311	45	253	102	326
33	196	261	274	128	307	178	257	131	339
34	169	299	315	152	289	125	237	146	336
35	153	305	321	157	286	96	235	166	335
36	140	313	323	161	245	98	282	184	334
37	114	318	326	174	225	102	277	244	331
38	124	330	336	166	237	123	276	272	340
39	131	349	335	206	218	120	289	213	356
40	150	350	334	257	196	69	318	200	343
41	166	198	331	256	186	53	313	176	347
42	212	183	340	243	164	59	322	201	327
43	247	187	356	210	99	66	337	189	329

Dia da semana Ordem\Rota	2	2	2	3	3	3	4	4	4
	1	2	3	1	2	3	1	2	3
44	253	190	347	215	94	48	338	182	328
45	235	188	327	226	89	19	349	197	317
46	225	180	328	265	88	15	350	173	300
47	221	130	329	278	63	14	325	165	281
48	237	117	317	276	37	10	155	154	259
49	257	92	312	288	0	7	261	134	241
50	256	142	300	294	-	5	227	90	240
51	262	145	281	283	-	0	217	87	57
52	282	184	241	303	-	-	145	86	44
53	277	176	57	304	-	-	98	103	38
54	276	200	44	297	-	-	95	144	46
55	289	213	38	296	-	-	74	190	27
56	268	244	46	287	-	-	66	198	20
57	264	284	27	260	-	-	69	183	47
58	269	308	18	209	-	-	26	187	18
59	266	322	22	179	-	-	17	0	1
60	272	337	28	133	-	-	14	-	2
61	233	338	14	119	-	-	0	-	3
62	191	341	0	0	-	-	-	-	0
63	151	351	-	-	-	-	-	-	-
64	123	155	-	-	-	-	-	-	-
65	120	0	-	-	-	-	-	-	-
66	98	-	-	-	-	-	-	-	-
67	102	-	-	-	-	-	-	-	-
68	83	-	-	-	-	-	-	-	-
69	77	-	-	-	-	-	-	-	-
70	73	-	-	-	-	-	-	-	-
71	61	-	-	-	-	-	-	-	-
72	26	-	-	-	-	-	-	-	-
73	23	-	-	-	-	-	-	-	-
74	17	-	-	-	-	-	-	-	-
75	0	-	-	-	-	-	-	-	-

Fonte: o autor.

Tabela B-2 – Rotas de quinta-feira à sábado.

Dia da semana	5		6		6		7		7			
	Ordem	Rota	1	2	1	2	1	2	3	4		
1			0	0	0	0	0	0	0	0		
2			121	9	43	1	9	13	10	17	9	30
3			113	180	52	2	103	4	7	48	260	34
4			75	144	62	3	88	11	5	69	285	43
5			63	149	68	12	92	16	40	66	316	52
6			80	215	25	14	134	29	171	59	309	62
7			175	226	38	17	138	28	163	108	353	68
8			171	247	223	40	129	30	179	105	346	47
9			163	245	251	35	148	22	202	95	348	20
10			179	225	271	70	153	18	183	83	320	38
11			202	196	270	106	139	27	144	85	325	234
12			183	186	275	136	143	38	149	78	324	271
13			309	177	274	149	114	46	147	84	338	275
14			316	168	298	111	132	44	116	97	319	240
15			296	117	293	112	131	57	94	186	311	223
16			297	147	207	130	150	33	99	215	307	193
17			294	116	214	158	151	110	117	226	289	72
18			283	99	255	159	123	231	135	252	283	58
19			276	94	250	160	102	229	141	278	273	178
20			278	90	239	168	95	242	138	276	253	98
21			273	89	249	144	82	205	139	292	257	102
22			267	88	167	198	67	220	152	287	256	55
23			243	85	193	183	76	181	157	297	243	39
24			252	78	72	187	165	208	161	296	216	42
25			257	66	58	258	173	238	162	294	218	41
26			253	59	178	155	169	279	182	272	164	14
27			237	48	104	248	145	312	197	264	177	0
28			218	69	93	228	105	315	196	245	168	-
29			182	84	71	219	100	321	169	237	0	-
30			197	97	39	192	83	317	153	225	-	-
31			185	164	42	199	74	327	140	166	-	-
32			169	260	41	222	73	328	122	189	-	-
33			153	287	23	212	85	329	129	174	-	-
34			139	299	19	203	84	331	114	123	-	-
35			152	305	15	98	109	332	128	120	-	-
36			157	308	14	125	127	340	118	54	-	-
37			161	310	0	176	128	352	137	24	-	-
38			162	341	-	200	146	356	90	19	-	-
39			189	353	-	213	184	342	89	15	-	-
40			170	351	-	264	244	347	88	0	-	-
41			174	346	-	269	272	344	63	-	-	-
42			101	345	-	284	290	333	0	-	-	-

Dia da semana Ordem\Rota	5			6			7		7	
	1	2	3	1	2	3	1	2	3	4
43	95	348	-	268	291	336	-	-	-	-
44	83	324	-	256	295	335	-	-	-	-
45	77	320	-	253	302	334	-	-	-	-
46	108	307	-	235	318	323	-	-	-	-
47	128	311	-	257	314	326	-	-	-	-
48	126	306	-	265	313	300	-	-	-	-
49	118	266	-	276	322	281	-	-	-	-
50	142	272	-	288	337	280	-	-	-	-
51	137	233	-	289	338	241	-	-	-	-
52	135	191	-	282	349	230	-	-	-	-
53	138	194	-	277	350	224	-	-	-	-
54	122	166	-	262	354	60	-	-	-	-
55	129	55	-	217	303	65	-	-	-	-
56	114	54	-	210	304	45	-	-	-	-
57	105	10	-	227	286	51	-	-	-	-
58	124	7	-	232	237	91	-	-	-	-
59	123	6	-	261	225	81	-	-	-	-
60	120	5	-	190	221	96	-	-	-	-
61	98	0	-	188	206	61	-	-	-	-
62	102	-	-	171	201	56	-	-	-	-
63	24	-	-	133	189	69	-	-	-	-
64	21	-	-	37	195	53	-	-	-	-
65	64	-	-	0	156	66	-	-	-	-
66	0	-	-	-	182	49	-	-	-	-
67	-	-	-	-	197	36	-	-	-	-
68	-	-	-	-	172	26	-	-	-	-
69	-	-	-	-	140	0	-	-	-	-
70	-	-	-	-	141	-	-	-	-	-
71	-	-	-	-	90	-	-	-	-	-
72	-	-	-	-	87	-	-	-	-	-
73	-	-	-	-	86	-	-	-	-	-
74	-	-	-	-	0	-	-	-	-	-

Fonte: o autor.

APÊNDICE C - Rotas usadas pela empresa

Tabela C-1 – Rotas de segunda-feira a quarta-feira

Dia da semana	2		3		3		4		4	
	1	2	3	1	2	3	1	2	3	
1	0	0	0	0	0	0	0	0	0	
2	2	10	190	14	97	37	2	10	2	
3	103	14	82	28	137	180	103	8	103	
4	113	42	169	193	141	144	92	12	113	
5	111	43	141	65	148	80	160	14	111	
6	92	91	182	181	169	87	199	42	92	
7	149	93	196	207	185	149	177	43	149	
8	159	110	150	220	196	175	236	91	159	
9	160	58	203	263	197	209	183	51	160	
10	177	193	206	355	182	202	187	110	177	
11	258	341	237	295	204	183	133	58	258	
12	187	346	225	302	237	198	48	193	187	
13	40	167	264	301	245	35	69	181	40	
14	48	181	284	290	225	40	66	208	48	
15	59	214	289	291	273	59	59	214	59	
16	67	229	282	272	278	66	128	229	67	
17	76	263	273	266	276	84	108	263	76	
18	128	295	294	269	288	114	124	295	128	
19	124	272	297	284	299	128	146	272	124	
20	146	234	296	268	303	95	131	234	146	
21	131	238	299	178	252	108	95	238	131	
22	95	223	104	110	243	98	98	223	95	
23	53	251	164	47	232	151	123	251	53	
24	74	254	226	62	227	184	151	254	74	
25	66	279	212	52	142	286	96	270	66	
26	98	300	139	38	289	222	68	279	98	
27	123	317	157	46	307	215	54	280	123	
28	151	326	161	33	306	216	76	281	151	
29	68	332	257	20	314	172	162	274	68	
30	54	339	262	25	313	118	170	312	54	
31	172	281	308	16	341	138	174	241	172	
32	162	312	318	11	346	219	216	240	162	
33	218	246	322	0	353	112	121	326	218	
34	216	240	338	-	348	119	144	344	216	
35	144	356	324	-	316	0	171	342	144	
36	171	347	155	-	155	-	119	352	171	
37	19	340	228	-	6	-	0	356	19	
38	0	298	9	-	9	-	-	271	0	
39	-	207	0	-	0	-	-	347	-	
40	-	208	-	-	-	-	-	340	-	
41	-	57	-	-	-	-	-	57	-	
42	-	44	-	-	-	-	-	44	-	
43	-	38	-	-	-	-	-	38	-	

Dia da semana	2	2	2	3	3	3	4	4	4
Ordem\Rota	1	2	3	1	2	3	1	2	3
44	-	25	-	-	-	-	-	25	-
45	-	29	-	-	-	-	-	29	-
46	-	22	-	-	-	-	-	22	-
47	-	18	-	-	-	-	-	18	-
48	-	71	-	-	-	-	-	13	-
49	-	13	-	-	-	-	-	0	-
50	-	0	-	-	-	-	-	-	-

Fonte: o autor.

Tabela C-2 – Rotas de quinta-feira à sábado.

Dia da semana	5		6		6		7		7	
	1	2	3	1	2	3	1	2	3	4
1	0	0	0	0	0	0	0	0	0	0
2	190	144	14	92	2	190	10	190	144	14
3	82	86	28	137	37	82	14	82	86	28
4	169	87	30	141	103	169	43	169	87	30
5	153	130	20	169	89	161	79	153	130	20
6	165	258	52	185	80	150	91	165	258	52
7	182	202	269	196	92	146	93	182	202	269
8	150	183	284	197	177	203	51	150	183	284
9	203	198	268	182	168	237	110	203	198	268
10	237	133	272	204	183	225	58	237	133	272
11	225	40	301	225	187	200	72	225	40	301
12	264	59	295	237	133	264	193	264	59	295
13	284	53	263	245	59	284	272	284	53	263
14	289	66	239	273	66	289	295	289	66	239
15	282	108	220	278	128	282	263	282	108	220
16	273	105	181	288	101	273	231	273	105	181
17	294	107	285	287	95	276	229	294	107	285
18	297	95	346	289	98	294	214	297	95	346
19	296	98	207	243	123	297	181	296	98	207
20	232	151	293	232	151	296	207	232	151	293
21	164	125	193	206	96	232	208	164	125	193
22	226	286	65	276	68	164	238	226	286	65
23	247	222	45	307	54	247	223	247	222	45
24	212	215	60	306	76	142	279	212	215	60
25	139	216	33	314	141	139	281	139	216	33
26	157	128	62	313	138	143	274	157	128	62
27	161	118	79	310	172	157	312	161	118	79
28	257	219	38	305	162	257	259	257	219	38
29	262	112	46	341	216	262	240	262	112	46
30	304	159	25	346	199	304	317	304	159	25
31	308	0	16	351	144	318	326	308	0	16
32	318	-	24	348	171	325	331	318	-	24
33	325	-	4	316	163	324	339	325	-	4
34	324	-	0	155	0	320	332	324	-	0
35	322	-	-	285	-	322	340	322	-	-
36	338	-	-	260	-	338	343	338	-	-
37	349	-	-	248	-	330	344	349	-	-
38	261	-	-	6	-	349	342	261	-	-
39	228	-	-	9	-	354	352	228	-	-
40	7	-	-	0	-	350	356	7	-	-
41	6	-	-	-	-	260	347	6	-	-
42	5	-	-	-	-	261	57	5	-	-

Dia da semana	5	5	5	6	6	6	7	7	7	7
Ordem\Rota	1	2	3	1	2	3	1	2	3	4
43	9	-	-	-	-	228	44	9	-	-
44	0	-	-	-	-	7	38	0	-	-
45	-	-	-	-	-	5	25	-	-	-
46	-	-	-	-	-	9	29	-	-	-
47	-	-	-	-	-	0	22	-	-	-
48	-	-	-	-	-	-	18	-	-	-
49	-	-	-	-	-	-	71	-	-	-
50	-	-	-	-	-	-	98	-	-	-
51	-	-	-	-	-	-	13	-	-	-
52	-	-	-	-	-	-	0	-	-	-

Fonte: o autor.