



CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL

**ABORDAGENS MONO E MULTIOBJETIVO
PARA O PROBLEMA DE ROTEAMENTO DE
VEÍCULOS CAPACITADOS COM
RESTRIÇÕES TRIDIMENSIONAIS DE
CARREGAMENTO**

LUCAS DANIEL PADIA ROCHA

Orientador: Flávio Vinícius Cruzeiro Martins
CEFET-MG

BELO HORIZONTE
FEVEREIRO DE 2020

LUCAS DANIEL PADIA ROCHA

**ABORDAGENS MONO E MULTIOBJETIVO PARA O
PROBLEMA DE ROTEAMENTO DE VEÍCULOS
CAPACITADOS COM RESTRIÇÕES TRIDIMENSIONAIS
DE CARREGAMENTO**

Dissertação apresentada ao Programa de Pós-graduação em Modelagem Matemática e Computacional do Centro Federal de Educação Tecnológica de Minas Gerais, como requisito parcial para a obtenção do título de Mestre em Modelagem Matemática e Computacional.

Área de concentração: Modelagem Matemática e Computacional

Linha de pesquisa: Sistemas Inteligentes

Orientador: Flávio Vinícius Cruzeiro Martins
CEFET-MG

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL
BELO HORIZONTE
FEVEREIRO DE 2020

Rocha, Lucas Daniel Padia
R672a Abordagens mono e multiobjetivo para o problema de roteamento de
veículos capacitados com restrições tridimensionais de carregamento /
Lucas Daniel Padia Rocha. – 2020.
xv, 67 f.

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em
Modelagem Matemática e Computacional.

Orientador: Flávio Vinícius Cruzeiro Martins.

Dissertação (mestrado) – Centro Federal de Educação Tecnológica de
Minas Gerais.

1. Programação heurística – Teses. 2. Otimização combinatória – Teses.
3. Transporte de carga – Teses. 4. Transporte em containers – Teses.
I. Martins, Flávio Vinícius Cruzeiro. II. Centro Federal de Educação
Tecnológica de Minas Gerais. III. Título.

CDD 519.6



SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
COORDENAÇÃO DO PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL

**“ABORDAGENS MONO E MULTI OBJETIVO PARA O PROBLEMA DE
ROTEAMENTO DE VEÍCULOS CAPACITADOS COM RESTRIÇÕES
TRIDIMENSIONAIS DE CARREGAMENTO”**

Dissertação de Mestrado apresentada por **Lucas Daniel Padia Rocha**, em 28 de fevereiro de 2020, ao Programa de Pós-Graduação em Modelagem Matemática e Computacional do CEFET-MG, e aprovada pela banca examinadora constituída pelos professores:

Prof. Dr. Flávio Vinicius Cruzeiro Martins (Orientador)
Centro Federal de Educação Tecnológica de Minas Gerais

Prof. Dr. João Fernando Machry Sarubbi
Centro Federal de Educação Tecnológica de Minas Gerais

Prof. Dr. Elizabeth Fialho Wanner
Centro Federal de Educação Tecnológica de Minas Gerais

Visto e permitida a impressão.

Prof. Dr. Thiago de Souza Rodrigues
Coordenador do Programa de Pós-Graduação em
Modelagem Matemática e Computacional

Agradecimentos

Agradeço primeiramente a Deus, que me deu forças para iniciar, permanecer e finalizar esta caminhada. Aos meus pais, Andréa e Eduardo, que me deram estrutura para que eu pudesse chegar até aqui. Ao meu orientador, professor Dr. Flávio Cruzeiro, por ter me aceito como orientando e por todo suporte e ensinamentos ao longo deste trabalho. Aos meus colegas do programa, que me deram todo apoio quando precisei. À CAPES, pelo auxílio financeiro para participações em eventos. E por fim, agradeço ao CEFET-MG por ter permitido a dedicação parcial da minha jornada de trabalho para realizar as atividades do mestrado.

“A mente que se abre a uma nova ideia jamais voltará ao seu tamanho original.” (Albert Einstein)

Resumo

Com a expansão da internet, o transporte de produtos ocupa uma das funções mais importantes no comércio eletrônico. Um dos parâmetros com elevado impacto nessa importância está nos altos custos envolvidos com o transporte na economia. Diante desse fato, é considerável analisar os problemas reais enfrentados pelos atores desse ramo de atividade. As questões que se referem ao roteamento e carregamento de veículos, em decorrência de suas complexidades, constituem uma das aplicações com maior potencial na área de otimização. Com o foco de facilitar essas questões, este trabalho tem como objetivo solucionar o problema de roteamento de veículos capacitados com restrições tridimensionais de carregamento (3L-CVRP). Este trabalho traz soluções para o problema em duas modelagens distintas, mono-objetivo e multiobjetivo. No mono-objetivo, a finalidade é minimizar a distância total percorrida pelos veículos. Foram implementados os seguintes algoritmos: Algoritmo Genético, *Variable Neighborhood Search* (VNS), *Simulated Annealing* e Busca Tabu. No Algoritmo Genético, utilizou-se as soluções geradas pelas técnicas *Greedy Randomized Adaptive Search Procedure* (GRASP) e soluções resolvidas por programação linear inteira (PLI) utilizando o pacote comercial CPLEX aplicadas ao problema de roteamento de veículos capacitados (CVRP) para construir a população inicial. No restante dos algoritmos implementados, as soluções do PLI obtidas pelo CPLEX foram utilizadas como solução inicial. Os resultados obtidos foram comparados aos da literatura e entre os algoritmos construídos. Na abordagem multiobjetivo, acrescentou-se o objetivo de minimizar a quantidade de veículos utilizados para a entrega dos itens. Foi feita a implementação do algoritmo *Non-Dominated Sorting Genetic Algorithm II* (NSGA-II) aplicado ao CVRP e 3L-CVRP. Para o CVRP, as soluções encontradas foram comparadas às da literatura. O algoritmo alcançou resultados satisfatórios, se aproximando do estado da arte, e em algumas instâncias, superando as soluções obtidas. Para o 3L-CVRP, os resultados alcançados exploraram novas soluções para o problema em comparação com a literatura.

Palavras-chave: Meta-heurísticas. Otimização. Roteamento de Veículos. Carregamento de Contêineres Tridimensional.

Abstract

With internet expansion, products' transportation occupies one of the essential functions in electronic commerce. One of the parameters with a high impact on this importance is the high transportation costs in the economy. Given the fact, it is considered to analyze people's real problems in this industry. The routing and loading of vehicles' problems, due to their complexities, are one of the applications with more significant potential in the area of optimization. The goal of this work is to solve the Three-Dimensional Loading Capacitated Vehicle Routing Problem (3L-CVRP). This work shows solutions to the problem in two different models, mono-objective and multiobjective. In mono-objective, the goal is to minimize the total distance traveled by vehicles. The following algorithms were implemented: Genetic Algorithm, Variable Neighborhood Search (VNS), Simulated Annealing, and Tabu Search. In the Genetic Algorithm, the solutions generated by the Greedy Randomized Adaptive Search Procedure (GRASP) technique and solutions are solved by integer linear programming (PLI) using the CPLEX commercial package applied to the Capacitated Vehicle Routing Problem (CVRP) were used to build the initial population. In the rest of the implemented algorithms, the PLI solutions obtained by CPLEX were used as an initial solution. The results obtained were compared to those in the literature and among the constructed algorithms. In the multiobjective approach, the objective was added to minimize the number of vehicles used to deliver the items. The Non-Dominated Sorting Genetic Algorithm II (NSGA-II) algorithm applied to CVRP and 3L-CVRP was implemented. For CVRP, the solutions found were compared to those in the literature. The algorithm reached satisfactory results, approaching state of the art, and in some instances, overcoming the solutions obtained. For the 3L-CVRP, the results reached explored new solutions to the problem compared with the literature.

Keywords: Metaheuristics. Optimization. Vehicle Routing. Three-Dimensional Containers Loading.

Lista de Figuras

Figura 1 – Rotação nos eixos de comprimento e largura	12
Figura 2 – Rotação nos eixos da altura e comprimento	12
Figura 3 – Rotação nos eixos da altura e largura	13
Figura 4 – Carregamento correto em relação a fragilidade	13
Figura 5 – Carregamento incorreto em relação a fragilidade	13
Figura 6 – Carregamento correto em relação a estabilidade	14
Figura 7 – Carregamento incorreto em relação a estabilidade	14
Figura 8 – Movimentos permitidos na retirada de uma caixa	14
Figura 9 – Exemplo de instância para o 3L-CVRP. Os itens frágeis estão acinzentados	17
Figura 10 – Solução da instância exemplo	17
Figura 11 – EPs definidos em um item (os EPs são os triângulos)	19
Figura 12 – Representação dos <i>Extreme Points</i> em 2D e 3D	19
Figura 13 – Sequência de Carregamento de Itens	19
Figura 14 – Representação da Fronteira Pareto-Ótima	20
Figura 15 – Estrutura básica de um algoritmo genético	23
Figura 16 – Cruzamento em um ponto	25
Figura 17 – Cruzamento em dois pontos	25
Figura 18 – Mutação	26
Figura 19 – Representação dos <i>fronts</i> do algoritmo NSGA-II	26
Figura 20 – Representação da distância de multidão	27
Figura 21 – Esquema do algoritmo NSGA-II	28
Figura 22 – Algoritmo de carregamento factível. Os quadrados de cor verde são clien- tes e itens em que se obteve uma solução factível. Os de cor vermelho são infactíveis. Os de cor branca são clientes e itens no qual não foram feitas tentativas de carregamento	34
Figura 23 – Estratégia de carregamento	35
Figura 24 – Representação de um indivíduo do AG	36
Figura 25 – Decodificação de um indivíduo do AG para o CVRP	37
Figura 26 – PMX - <i>Partially Matched Crossover</i>	40
Figura 27 – Operador de Mutação	40
Figura 28 – Tipos de movimentos que são feitos na geração de novos vizinhos	41
Figura 29 – Comportamento da lista tabu	42
Figura 30 – Representação de um indivíduo no NSGA-II	43
Figura 31 – Permuta a ordem dos itens do cliente	44
Figura 32 – Testa-se o carregamento para os itens do próximo cliente da lista. . . .	45
Figura 33 – Cruzamento na lista de clientes	45

Figura 34 – Cruzamento na lista de itens	46
Figura 35 – Operador de mutação na ordem de carregamento dos itens de um cliente	46
Figura 36 – Solução roteamento da instância 3l_cvrp02	50
Figura 37 – Solução carregamento da instância 3l_cvrp02 - Rota 6-11-5	51
Figura 38 – Solução carregamento da instância 3l_cvrp02 - Rota 1-3-2	51
Figura 39 – Solução carregamento da instância 3l_cvrp02 - Rota 14-13-4	52
Figura 40 – Solução carregamento da instância 3l_cvrp02 - Rota 9-10-15-12	52
Figura 41 – Solução carregamento da instância 3l_cvrp02 - Rota 8-7	53
Figura 42 – <i>Boxplots</i> dos resultados obtidos pelos algoritmos na abordagem mono- objetivo	54
Figura 43 – Fronteira Pareto-ótima da instância 3l_cvrp01. Os pontos em azul repre- sentam os resultados obtidos pelo algoritmo e os pontos em vermelho representam as soluções da literatura	59
Figura 44 – Fronteira Pareto-ótima da instância 3l_cvrp02	59
Figura 45 – Fronteira Pareto-ótima da instância 3l_cvrp03	60
Figura 46 – Fronteira Pareto-ótima da instância 3l_cvrp05	60

Lista de Tabelas

Tabela 1 – Parâmetros do algoritmo genético	47
Tabela 2 – Parâmetros do <i>Simulated Annealing</i>	48
Tabela 3 – Tempos de execução utilizados	48
Tabela 4 – Melhores Resultados Computacionais 3L-CVRP	49
Tabela 5 – Média dos Resultados Computacionais 3L-CVRP	49
Tabela 6 – Desvio Padrão dos Resultados Computacionais 3L-CVRP	49
Tabela 7 – Resultados do teste Shapiro-Wilk	55
Tabela 8 – Resultados do teste Kruskal-Wallis	55
Tabela 9 – Resultados do teste <i>Wilcoxon</i> para instância 3l_cvrp01	56
Tabela 10 – Resultados do teste <i>Wilcoxon</i> para instância 3l_cvrp02	56
Tabela 11 – Resultados do teste <i>Wilcoxon</i> para instância 3l_cvrp03	56
Tabela 12 – Resultados do teste <i>Wilcoxon</i> para instância 3l_cvrp04	56
Tabela 13 – Resultados do teste <i>Wilcoxon</i> para instância 3l_cvrp06	56
Tabela 14 – Resultados do teste <i>Wilcoxon</i> para instância 3l_cvrp07	56
Tabela 15 – Parâmetros utilizados no NSGA-II para o CVRP	57
Tabela 16 – Resultados Computacionais CVRP	58
Tabela 17 – Parâmetros utilizados no NSGA-II para o 3L-CVRP	58

Lista de Quadros

Quadro 1 – Comparação dos Trabalhos Relacionados	5
--	---

Lista de Algoritmos

Algoritmo 1 – Algoritmo Genético	24
Algoritmo 2 – NSGA-II	29
Algoritmo 3 – VNS	29
Algoritmo 4 – <i>Simulated Annealing</i>	30
Algoritmo 5 – Busca Tabu	32

Lista de Abreviaturas e Siglas

AG	Algoritmo Genético
ANOVA	Análise de Variância
BKS	<i>Best Known Solution</i>
BPP	<i>Bin-Packing Problem</i>
3D-BPP	<i>Three-dimensional Bin Packing Problem</i>
BT	Busca Tabu
CLP	<i>Container Loading Problem</i>
CNMAC	Congresso Nacional de Matemática Aplicada e Computacional
CP	<i>Corner Point</i>
CVRP	<i>Capacitated Vehicle Routing Problem</i>
2L-CVRP	<i>Two Dimensional Capacitated Vehicle Routing Problem</i>
3L-CVRP	<i>3D-Loading Capacitated Vehicle Routing Problem</i>
CVRPLIB	<i>Capacitated Vehicle Routing Problem Library</i>
EP	<i>Extreme Point</i>
FFD	<i>First Fit Decreasing</i>
GRASP	<i>Greedy Randomized Adaptive Search Procedure</i>
LIFO	<i>Last In-First Out</i>
3LP	<i>Three Dimensional Problem</i>
NSGA-II	<i>Non-Dominated Sorting Genetic Algorithm II</i>
PMX	<i>Partially Matched Crossover</i>
SA	<i>Simulated Annealing</i>
SBPO	Simpósio Brasileiro de Pesquisa Operacional
VNS	<i>Variable Neighborhood Search</i>
VRP	<i>Vehicle Routing Problem</i>

Sumário

1 – Introdução	1
1.1 Objetivos: geral e específicos	2
1.2 Trabalhos Relacionados	3
1.3 Organização do trabalho	4
2 – Descrição do Problema	6
2.1 Carregamento de Contêineres	6
2.2 Roteamento de Veículos	9
2.3 Roteamento de Veículos Capacitados com Restrições Tridimensionais de Carregamento	11
3 – Fundamentação Teórica	18
3.1 Métodos de Carregamento	18
3.1.1 <i>Extreme Point-Based Heuristic</i>	18
3.2 Otimização Mono-objetivo e Multiobjetivo	18
3.3 Heurísticas	20
3.3.1 Heurísticas Construtivas	21
3.3.2 Heurísticas de Refinamento	21
3.3.3 Algoritmo Genético	21
3.3.3.1 Definição da População Inicial	22
3.3.3.2 Seleção	23
3.3.3.3 Cruzamento	24
3.3.3.4 Mutação	25
3.3.4 Algoritmo NSGA-II	25
3.3.5 Algoritmo VNS	28
3.3.6 <i>Simulated Annealing</i>	30
3.3.7 Busca Tabu	31
4 – Metodologia	33
4.1 Modelagem Mono-objetivo	33
4.1.1 Algoritmo de Carregamento	33
4.1.2 Algoritmo Genético	36
4.1.2.1 Codificação	36
4.1.2.2 Decodificação e Avaliação do Indivíduo	36
4.1.2.3 Função <i>Fitness</i>	36
4.1.2.4 População Inicial	38

4.1.2.5	Seleção	38
4.1.2.6	Cruzamento	39
4.1.2.7	Mutação	39
4.1.3	VNS, Busca Tabu e <i>Simulated Annealing</i>	39
4.1.3.1	Solução Inicial	39
4.1.3.2	Vizinhança	40
4.1.3.3	Lista Tabu	41
4.2	Modelagem Multiobjetivo	42
4.2.1	Algoritmo NSGA-II	42
4.2.1.1	Codificação	42
4.2.1.2	Decodificação e Avaliação do Indivíduo	43
4.2.1.3	População Inicial	43
4.2.1.4	Operações	44
5	– Experimentos Computacionais	47
5.1	Abordagem Mono-objetivo	47
5.1.0.1	Análise estatística dos resultados	53
5.2	Abordagem Multiobjetivo	56
5.2.1	Resultados CVRP	56
5.2.2	Resultados 3L-CVRP	58
6	– Considerações Finais	61
6.1	Trabalhos Futuros	62
6.2	Publicações	63
	Referências	64

Capítulo 1

Introdução

Com o avanço da globalização e da internet, o transporte de produtos ocupa uma das funções mais importantes dentro da cadeia de suprimentos. Um dos parâmetros que fortalece essa importância está nos elevados custos envolvidos com o transporte na economia (GUIMARÃES, 2012). No ano de 2012, os custos logísticos representaram cerca de 11,5% do produto interno bruto brasileiro, e dessa parcela, aproximadamente 62% referem-se a custos com transporte, além de 7% com custos de armazenagem. Em valores absolutos, os custos com transporte e armazenagem representaram em 2012 aproximadamente R\$ 313,9 bilhões e R\$ 35 bilhões, respectivamente (LIMA, 2014).

Dada a importância do transporte rodoviário para a economia, é importante analisar os problemas reais enfrentados pelos atores desse ramo de atividade. De acordo com Guimarães (2012), questões que se referem ao roteamento e carregamento de veículos, dada a complexidade acerca desses temas, constituem uma das aplicações com mais potencial da área de Pesquisa Operacional, logrando a este ramo da matemática aplicada uma importante posição no âmbito da tomada de decisão em logística.

O problema em foco aborda dois subproblemas conhecidos na literatura: o roteamento de veículos e o empacotamento de cargas. Ambos tratados de forma individual já são difíceis de serem resolvidos, visto que pertencem à classe de problemas NP-Difíceis. Uma forma de resolver tais problemas seria simplesmente enumerar todas as soluções possíveis, o que é inviável quando aplicado em um tamanho de um problema real de logística. Reunidos, eles possuem uma complexidade ainda maior.

Em relação aos problemas em questão, a literatura é extensa no tratamento individual do problema de carregamento de contêiner (CLP - *Container Loading Problem*) e do problema de roteamento de veículos capacitados (CVRP - *Capacitated Vehicle Routing Problem*), porém é escassa na abordagem integrada desses dois problemas de forma tridimensional, chamado 3L-CVRP (*3D-Loading Capacitated Vehicle Routing Problem*). Essa abordagem

com a resolução combinada dos dois problemas foi inicialmente proposta por [Gendreau et al. \(2006\)](#), no qual ele apresentou um algoritmo de Busca Tabu, que é um procedimento auxiliar que guia um algoritmo de busca local na exploração de um espaço de soluções, combinado com outra Busca Tabu interna iterativa para a solução do subproblema de carregamento. Mesmo que o CVRP venha sendo muito abordado nos últimos tempos, ele acaba sendo uma simplificação da situação atual, já que a restrição de carregamento é unidimensional ([GUIMARÃES, 2012](#)). De acordo com [Campos e Mota \(2000\)](#), a abordagem CVRP é requerida quando os produtos a serem distribuídos são de baixa densidade de transporte. Este é o caso de produtos que completam a capacidade física de um veículo pelo seu volume, não alcançando o limite da capacidade de peso que o veículo suporta. Neste caso, restrições espaciais de carregamento são necessárias para uma modelagem mais realista do problema de roteamento, de forma que seja possível minimizar os custos de transporte, relativos à quantidade de veículos, bem como à distância total percorrida.

Este trabalho apresenta duas abordagens para otimizar a solução do problema 3L-CVRP, mono-objetivo e multiobjetivo. O objetivo deste problema é distribuir de maneira eficiente um conjunto de itens (ou cargas) para seus respectivos clientes. Toda carga está concentrada em um depósito, e deve-se utilizar um conjunto de veículos para entregá-las. Existem restrições de fragilidade, superfície de apoio e ordem de retirada das cargas que devem ser respeitadas. Além disso, todos os caminhões são homogêneos e as cargas têm um formato de paralelepípedo. O objetivo é minimizar a distância total percorrida pelos veículos disponíveis. Este problema pertence à classe de NP-Difícil e conseqüentemente não existem na literatura algoritmos capazes de resolvê-lo em tempo polinomial. Devido a isso, métodos heurísticos e exatos são aplicados ao problema.

1.1 Objetivos: geral e específicos

O objetivo geral deste trabalho é utilizar meta-heurísticas para resolução do problema de roteamento de veículos capacitados com restrições tridimensionais de carregamento por meio de duas abordagens distintas, mono e multiobjetivo. Em relação aos objetivos específicos, tem-se:

- Realizar um estudo da literatura acerca das soluções já propostas para o problema;
- Utilizar algoritmos genéticos combinados com heurísticas conhecidas, avaliá-las sobre instâncias clássicas da literatura e compará-las com procedimentos exatos e heurísticos existentes.
- Analisar o desempenho do método completo proposto para a resolução do 3L-CVRP, através de testes computacionais e estatísticos realizados sobre instâncias clássicas da literatura.

1.2 Trabalhos Relacionados

Alguns trabalhos descritos para o problema de roteamento de veículos capacitados com restrições tridimensionais de carregamento podem ser encontrados na literatura. Trabalhos sobre carregamento de contêineres, roteamento de veículos capacitados ou a junção destes dois acrescentando-se restrições bidimensionais podem ser úteis, já que o problema 3L-CVRP é uma abordagem mais completa dos outros três.

Para resolver o 3L-CVRP, [Portal et al. \(2009\)](#) apresentaram um algoritmo de Busca Tabu, baseado fortemente na proposta de [Gendreau et al. \(2006\)](#). Inicialmente fez-se uma busca local responsável pelo empacotamento de um conjunto de itens em um veículo. No caso do roteamento, foi realizada uma Busca Tabu com o objetivo de minimizar o custo total das rotas. A partir dos resultados, os autores concluíram que obter uma solução inicial é bastante importante no processo de alcançar uma melhor solução final para o problema, valendo a pena utilizar mais tempo calculando uma solução inicial de boa qualidade. Para este cálculo, a técnica que gerou melhores resultados foi o Algoritmo de Economia ("Savings") de [Clarke e Wright \(1964\)](#).

Um outro artigo que utilizou Busca Tabu foi [Tarantilis, Zachariadis e Kiranoudis \(2009\)](#). Para resolver ambos os problemas que constituem o 3L-CVRP, os autores propuseram uma metodologia que utiliza uma meta-heurística híbrida que combina estratégias de Busca Tabu com busca local guiada. Para os testes, utilizaram as mesmas instâncias usadas por [Gendreau et al. \(2006\)](#) e instâncias desenvolvidas por eles. As soluções encontradas se mostraram superiores às da literatura.

Em [Fuellerer et al. \(2010\)](#), os autores utilizaram o algoritmo de Colônia de Formigas, utilizando uma heurística para o carregamento dos produtos. O algoritmo combina duas medidas diferentes para a heurística, uma para o roteamento e outra para o carregamento. Em testes numéricos, a implementação alcançou novas soluções para a maioria das instâncias disponíveis na literatura.

Já [Bortfeldt \(2012\)](#) utilizou um algoritmo de Busca Tabu para o roteamento de veículos e um algoritmo de Busca em Árvore para o carregamento de contêineres. As soluções encontradas se mostraram mais eficientes em relação às existentes com um tempo total de computação reduzido.

Com o objetivo de solucionar o 3L-CVRP, [Zhu et al. \(2012\)](#) propuseram melhorias em duas heurísticas de carregamento conhecidas, *Deepest-Bottom-Left-Fill* e *Maximum Touching Area*. A partir dessas duas heurísticas, os autores utilizaram um algoritmo de Busca Tabu para abordar o problema. Novamente, as instâncias utilizadas foram as encontradas em [Gendreau et al. \(2006\)](#) e [Tarantilis, Zachariadis e Kiranoudis \(2009\)](#). De acordo com os

autores, a abordagem apresentada por eles superou as soluções presentes na literatura para 20 de 27 instâncias.

Em [Miao et al. \(2012\)](#) foi utilizado uma abordagem híbrida, que combina Algoritmo Genético com Busca Tabu. O Algoritmo Genético foi desenvolvido para o roteamento de veículos e a Busca Tabu para o carregamento tridimensional. Os dois algoritmos foram combinados para resolver o problema integrado. A solução se mostrou bastante eficaz, encontrando resultados melhores em comparação aos da literatura para algumas instâncias testadas.

Já [Ruan et al. \(2013\)](#) propuseram uma abordagem híbrida, que combina a otimização de Acasalamento de Abelhas e seis heurísticas, uma para o roteamento de veículos e as restantes para o carregamento tridimensional. Como resultados, obtiveram novas melhores soluções para algumas instâncias testadas.

Em [Tao e Wang \(2015\)](#), para resolver o 3L-CVRP, foi utilizado uma heurística chamada de algoritmo de Menor Desperdício para resolver o problema do carregamento tridimensional. Ela tem como objetivo trazer menos desperdício de espaço ao colocar uma caixa no veículo. Para o roteamento de veículos, utilizaram um algoritmo de Busca Tabu também combinado com o Algoritmo de Economia ("*Savings*") de [Clarke e Wright \(1964\)](#). Para os testes, utilizaram dois conjuntos de instâncias, encontradas em [Gendreau et al. \(2006\)](#) e [Tarantilis, Zachariadis e Kiranoudis \(2009\)](#). Os resultados dos testes computacionais mostraram que o algoritmo implementado por eles teve uma boa performance comparada com os algoritmos já existentes em termos de qualidade e eficiência da solução.

Em [Escobar-Falcón et al. \(2016\)](#), os autores propuseram uma abordagem em duas fases. Inicialmente, utilizaram um procedimento de otimização baseado em cortes para o problema de roteamento de veículos. Depois, para as soluções válidas encontradas na fase anterior, aplicou um algoritmo GRASP (*Greedy Randomized Adaptive Search Procedure*) para avaliar as restrições de empacotamento de cada uma das rotas.

Conforme o [Quadro 1](#), um resumo dos trabalhos encontrados na literatura, destacando as técnicas utilizadas pelos autores, pode ser visto. A maior parte dos trabalhos utilizou um algoritmo de busca combinada com heurísticas para resolver o problema. Dos mencionados, apenas um aplicou Algoritmo Genético.

1.3 Organização do trabalho

No [Capítulo 2](#) são detalhados os problemas em questão. O [Capítulo 3](#) apresenta uma breve explicação sobre os principais conceitos em comum com o tema abordado neste trabalho. No [Capítulo 4](#) é explicada a metodologia utilizada. Os resultados computacionais alcançados são relatados no [Capítulo 5](#). O [Capítulo 6](#) conclui, discute trabalhos futuros e

Quadro 1 – Comparação dos Trabalhos Relacionados

Trabalho	Técnicas utilizadas
Portal et al. (2009)	Busca Tabu combinada com busca local.
Tarantilis, Zachariadis e Kiranoudis (2009)	Meta-heurística híbrida combinando estratégias de Busca Tabu com busca local guiada.
Fuellerer et al. (2010)	Algoritmo Colônia de Formigas com heurística.
Bortfeldt (2012)	Algoritmo de Busca Tabu para o roteamento de veículos e um algoritmo de Busca em Árvore para o carregamento de contêineres.
Zhu et al. (2012)	Algoritmo de Busca Tabu combinada com duas heurísticas de carregamento conhecidas, <i>Deepest-Bottom-Left-Fill</i> e <i>Maximum Touching Area</i> .
Miao et al. (2012)	Algoritmo Genético combinado com Busca Tabu.
Ruan et al. (2013)	Algoritmo de Acasalamento de Abelhas combinada com seis heurísticas.
Tao e Wang (2015)	Algoritmo de Menor Desperdício para o problema do carregamento e Busca Tabu para o roteamento de veículos.
Escobar-Falcón et al. (2016)	Procedimento de otimização baseado em cortes para o roteamento de veículos e algoritmo GRASP para as restrições de empacotamento.

apresenta as publicações originadas da pesquisa.

Capítulo 2

Descrição do Problema

Neste capítulo descreve-se com maiores detalhes o problema abordado e suas versões simplificadas. Na [Seção 2.1](#), é apresentado a definição e o modelo matemático para o carregamento de veículos. Na [Seção 2.2](#) define-se o problema de roteamento de veículos e sua formulação matemática. Na [Seção 2.3](#) é apresentado a junção do roteamento e carregamento de veículos juntamente com suas restrições e seu modelo matemático.

2.1 Carregamento de Contêineres

O problema do carregamento de contêineres (CLP - *Container Loading Problem*) é um dos problemas clássicos da otimização combinatória, e pode ser caracterizado como uma abordagem parecida ao problema da mochila ([CHU; BEASLEY, 1998](#)). Esse consiste em determinar a posição de carregamento de caixas, de dimensões variadas, em um espaço com mesma geometria e dimensão maior, denominado contêiner. O objetivo do CLP é maximizar a ocupação do espaço disponível.

De acordo com [Bischoff e Ratcliff \(1995\)](#), existe uma série de restrições práticas que nem sempre são consideradas, tais quais:

- Orientação: em situações de fragilidade com instruções do tipo "esta face para cima";
- Estabilidade: para amparar o centro de gravidade dos itens, um percentual da área da base de cada caixa deve ser apoiado no piso do contêiner ou na face superior de outras caixas;
- Empilhamento: ao se carregar uma caixa sobre a outra, assume-se que a área de contato da primeira exerce uma pressão direta sobre a face superior da segunda;
- Agrupamento de itens e múltiplos destinos: do ponto de vista operacional, o descarregamento de um conjunto de caixas de determinado cliente não deve requerer a

retirada de outras caixas, pois estas deverão ser carregadas novamente. Então a ordem de descarregamento deve atender a ordem de entrega dos itens;

- Separação de produtos: para o caso de produtos que não podem ser carregados próximos uns dos outros, como alimentos e produtos de limpeza.

Em sua forma genérica, o problema de Carregamento de Contêineres possui muitas variações, em relação as restrições, objetivos e dimensões dos itens e contêineres. Uma variação do problema é conhecida como BPP (*Bin-Packing Problem*), na qual todos os objetos possuem um peso e cada item deve ser inserido dentro de um recipiente. O objetivo do BPP é encontrar uma solução que utilize a menor quantidade de contêineres no carregamento. A versão tridimensional desse problema, o 3D-BPP, faz parte da classe dos problemas NP-difícil, uma vez que é uma generalização do problema BPP (MARTELLO; PISINGER; VIGO, 2000).

O problema de carregamento de contêineres tridimensional (3D-BPP - *Three-dimensional Bin Packing Problem*), de acordo com Martello, Pisinger e Vigo (2000), é definido como: dado um conjunto de itens retangulares $I = 1, \dots, n$, cada um com altura h_i , largura w_i e profundidade l_i , com $i \in I$, e um conjunto infinito de contêineres C , todos com altura H , largura W e profundidade L , deve-se inserir todos os n itens no menor número de contêineres Z .

Em Silva e Soma (2003), os autores criaram o modelo matemático para o 3D-BPP da seguinte forma:

$$\text{Minimizar } \sum_{i=1}^n \eta_i \quad (1)$$

Sujeito à:

$$x_j + w_j \leq x_k + (1 - a_{jk})M, \forall j, k \in J, j < k \quad (2)$$

$$x_k + w_k \leq x_j + (1 - b_{jk})M, \forall j, k \in J, j < k \quad (3)$$

$$y_j + h_j \leq y_k + (1 - e_{jk})M, \forall j, k \in J, j < k \quad (4)$$

$$y_k + h_k \leq y_j + (1 - f_{jk})M, \forall j, k \in J, j < k \quad (5)$$

$$z_j + d_j \leq z_k + (1 - c_{jk})M, \forall j, k \in J, j < k \quad (6)$$

$$z_k + d_k \leq z_j + (1 - g_{jk})M, \forall j, k \in J, j < k \quad (7)$$

$$a_{jk} + b_{jk} + c_{jk} + e_{jk} + f_{jk} + g_{jk} \geq \alpha_{ji} + \alpha_{ki}, \forall j, k \in J, j < k, 1 \leq i \leq m \quad (8)$$

$$\sum_{i=1}^n \alpha_{ji} = 1, \forall j \in J \quad (9)$$

$$\sum_{j=1}^n \alpha_{ji} \leq M\eta_i, \forall 1 \leq i \leq m \quad (10)$$

$$x_j + w_j \leq W + (1 - \alpha_{ji})M, \forall j \in J, 1 \leq i \leq m \quad (11)$$

$$y_j + h_j \leq H + (1 - \alpha_{ji})M, \forall j \in J, 1 \leq i \leq m \quad (12)$$

$$z_j + d_j \leq D + (1 - \alpha_{ji})M, \forall j \in J, 1 \leq i \leq m \quad (13)$$

$$z_j + d_j \leq D + (1 - \alpha_{ji})M, \forall j \in J, 1 \leq i \leq m \quad (14)$$

$$a_{jk}, b_{jk}, c_{jk}, e_{jk}, f_{jk}, g_{jk}, \alpha_{ji}, \eta_i \in \{0, 1\}, \forall j, k \in J, j < k, 1 \leq i \leq m$$

$$x_j, y_j, z_j \geq 0, \forall j \in J$$

onde:

- n é a quantidade de itens;
- m é o número de contêineres disponíveis, onde se supõe, sem perda de generalidade que $m = n$;
- M é um número arbitrário bastante grande;
- α_{ji} é uma variável binária que é igual a 1 se o item j é colocado no contêiner i , caso contrário ela é igual a 0;
- η_i é uma variável binária que é igual a 1 se o contêiner i é usado, caso contrário ela é igual a 0;
- (x_j, y_j, z_j) variáveis contínuas (para localização dos itens) tendo como origem as coordenadas do canto inferior esquerdo do item j ;
- $a_{jk}, b_{jk}, c_{jk}, e_{jk}, f_{jk}, g_{jk}$ são as variáveis binárias definidas para indicar a colocação do item j em relação ao item k .

A variável a_{jk} é igual a 1 se o item j é colocado ao lado esquerdo do item k . Do mesmo modo, as variáveis b_{jk} , c_{jk} , e_{jk} , f_{jk} e g_{jk} representam quando o item j está à direita, atrás, abaixo, acima ou à frente do item k , respectivamente. Estas variáveis são necessárias e definidas somente quando $j > k$. Supõe-se que cada contêiner é colocado com a sua largura ao longo do eixo X , sua altura ao longo do eixo Y e seu comprimento ao longo do eixo Z , do sistema de coordenadas espaciais, com a origem sendo determinada no canto inferior esquerdo mais ao fundo do contêiner.

A função objetivo deste modelo implica em minimizar o número de contêineres utilizados para empacotar todos os n itens do problema. As restrições (2) a (7) garantem que os itens não vão ocupar o mesmo espaço dentro de um mesmo contêiner. A restrição (8) restringe se há sobreposição de itens alocados em um mesmo contêiner. A segurança que cada item será alocado em exatamente um recipiente está descrita na restrição (9). Um contêiner é considerado utilizado quando algum item é a ele alocado, este requisito é garantido pela restrição (10). As restrições (11) à (13) asseguram que todos os itens alocados dentro de um contêiner se encaixam dentro de suas dimensões físicas.

2.2 Roteamento de Veículos

De acordo com [Dantzig e Ramser \(1959\)](#), o problema clássico de roteamento de veículos (VRP) consiste na definição de rotas de entregas para uma determinada frota de veículos, localizadas inicialmente em um depósito, que atendam às demandas de um conjunto de clientes espalhados geograficamente. Desse modo, a restrição mais usual do problema está relacionada às capacidades dos veículos (autonomia, carregamento, depreciação, entre outros).

O problema do roteamento de veículos capacitados (CVRP - *Capacitated Vehicle Routing Problem*), é a junção dos problemas de roteamento e carregamento de forma simplificada, já que a restrição de carregamento é unidimensional. Ele é definido da seguinte forma: seja $G(V, E)$ um grafo não direcionado contendo o conjunto de vértices $V = \{0, 1, \dots, n\}$, onde o vértice 0 representa o depósito, enquanto que todos os outros vértices representam os clientes, de tal forma que cada cliente i possui uma demanda associada d_i . Cada arco $e \in E$ possui um comprimento não negativo $l(e)$. Dado G e dois números positivos e inteiros (K, C) , o CVRP consiste em encontrar um conjunto de rotas para os K veículos. Cada rota inicia e termina no depósito, cada cliente é visitado por um único veículo e a demanda total de todos os clientes de uma sub-rota não pode exceder a capacidade Q do veículo. O problema tem como objetivo minimizar o somatório de todas as sub-rotas ([GUIMARÃES, 2012](#)). O CVRP é NP-difícil, dado que é uma generalização do Problema do Caixeiro Viajante (PCV) ([LAWLER, 1985](#)), precisando, portanto, elevado esforço computacional para obtenção da solução ótima através de busca exaustiva. O modelo matemático do CVRP

adaptado de Lin et al. (2009) foi definido da seguinte forma:

$$\text{Minimizar } \sum_{i=0}^V \sum_{j=0}^V \sum_{k=1}^K C_{ij} X_{ijk} \quad (15)$$

Sujeito à:

$$\sum_{i=0}^V \sum_{j=0}^V X_{ijk} d_i \leq Q, \text{ onde } k \in \{1, \dots, K\} \quad (16)$$

$$\sum_{j=1}^V X_{ijk} = \sum_{j=1}^V X_{jik} \leq 1, \text{ onde } i = 0 \text{ e } k \in \{1, \dots, K\} \quad (17)$$

$$\sum_{k=1}^K \sum_{j=1}^V X_{ijk} \leq K, \text{ onde } i = 0 \quad (18)$$

$$X_{ijk} \in \{0, 1\}, i \neq j; i, j \in (0, 1, \dots, V)$$

onde:

- C_{ij} é o custo para ir do cliente i para o cliente j ;
- X_{ijk} é igual a 1 se o veículo k viaja do cliente i para o cliente j diretamente, 0 caso contrário;
- K é a quantidade de veículos disponível;
- V é o número de clientes;
- Q é a capacidade de carregamento dos veículos;
- d_i é a demanda do cliente i .

A equação (15) é a função objetivo do problema, que minimiza o somatório das distâncias de todas as subrotas. A restrição (16) limita a capacidade máxima de carregamento dos veículos e atendimento das demandas de cada cliente. O requisito que toda rota tem início e fim no depósito é garantida pela restrição (17). A restrição (18) especifica que há no máximo K rotas partindo do depósito.

2.3 Roteamento de Veículos Capacitados com Restrições Tridimensionais de Carregamento

Em [Iori, Salazar-González e Vigo \(2007\)](#), foi apresentado o 2L-CVRP (*Two Dimensional Capacitated Vehicle Routing Problem*), que consiste em uma simplificação do 3L-CVRP. Nesse problema, o veículo possui um espaço para carregamento bidimensional, e a finalidade é otimizar a distribuição de itens que não podem ser carregados de maneira sobreposta, uns sobre os outros.

Já o 3L-CVRP, proposto inicialmente por [Gendreau et al. \(2006\)](#) é uma generalização do 2L-CVRP na medida em que o carregamento permite a sobreposição de caixas. De acordo com os autores, o 3L-CVRP objetiva a definição das rotas de menor custo para uma frota de veículos, para a distribuição de itens demandados por um conjunto de clientes. Todas as rotas devem ter início e fim no depósito. Os itens consistem em caixas regulares com peso e volume definido. Todos os itens requeridos por um cliente devem ser entregues por um mesmo veículo, sendo que esses possuem a mesma capacidade de carregamento. A solução do problema requer a determinação, para cada veículo, do conjunto de itens que devem ser carregados e que não excedam a capacidade de peso do veículo e ao mesmo tempo produzam um carregamento possível no contexto espacial. As caixas devem ser carregadas com suas faces paralelas às faces do veículo, e é permitida a rotação das caixas em 90° no plano horizontal. As restrições citadas anteriormente devem ser consideradas.

Em [Fuellerer et al. \(2010\)](#), é descrito o 3L-CVRP do seguinte modo: dado um grafo completo e não orientado $G = (V, E)$, onde $V = \{0, 1, \dots, n\}$ é um conjunto de $n + 1$ vértices e E o conjunto de arestas conectado a cada par de vértices. O vértice "0" corresponde ao depósito, enquanto que $\{1, \dots, n\}$ é o conjunto de n clientes que devem ser visitados. Cada aresta é denotado por (i, j) e possui um custo de deslocamento c_{ij} com $(i, j = 0, \dots, n)$. É dado também uma frota de v veículos idênticos, com a mesma capacidade de carregamento, sendo D a capacidade de peso e dimensões de largura W , altura H e comprimento L . Os veículos possuem uma abertura na parte traseira para carga e descarga.

A demanda do cliente i é formada por um conjunto de m_i itens, cujo peso total é d_i , com $(i = 1, \dots, n)$. Cada um dos k itens do cliente i , denotado por I_{ik} , consiste em uma caixa, com largura w_{ik} , altura h_{ik} e comprimento l_{ik} ($i = 1, \dots, n : k = 1, \dots, m_i$). O volume total requerido pelo cliente i é denotado pela [Equação 19](#).

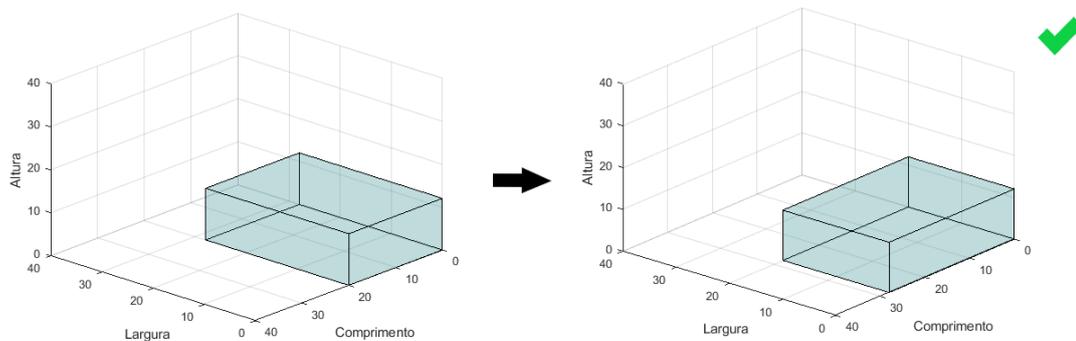
$$vol_i = \sum_{k=1}^{m_i} w_{ik} h_{ik} l_{ik} \quad (19)$$

O 3L-CVRP tem como objetivo encontrar um conjunto de no máximo v sub-rotas, tal que

cada sub-rota inicie e termine no depósito e cada cliente seja visitado uma única vez e por apenas um veículo. O carregamento de cada veículo deve ser factível e o somatório das distâncias de todas as sub-rotas deve ser minimizado. A factibilidade do carregamento é garantida se as caixas carregadas não excederem a capacidade de peso D e se existir um arranjo de caixas que atenda as restrições do 3L-BPP, acrescidas de restrições operacionais, definidas abaixo (GUIMARÃES, 2012):

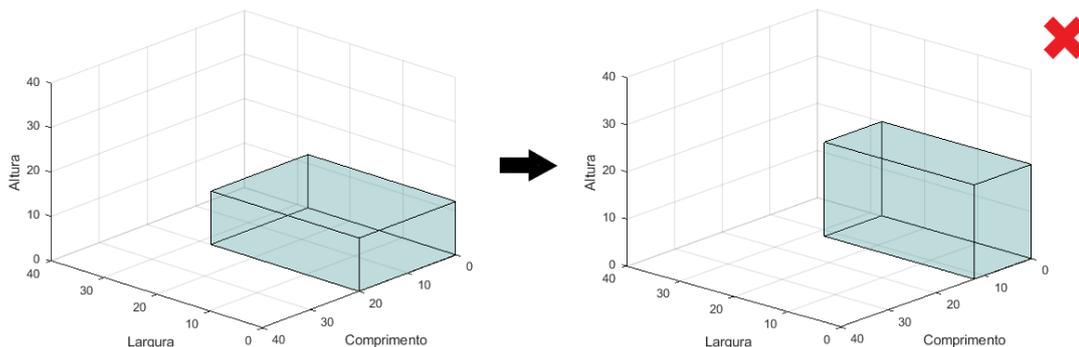
- O carregamento dos itens deve ser feito com seus lados paralelos às faces do veículo (de modo ortogonal). Rotações no plano $w - l$ são permitidas, mas a orientação de altura deve ser fixa, como pode ser visto nas Figuras 1, 2 e 3. Essa restrição garante as características reais de caixas com indicação "este lado para cima".

Figura 1 – Rotação nos eixos de comprimento e largura



Fonte: Elaborada pelo autor.

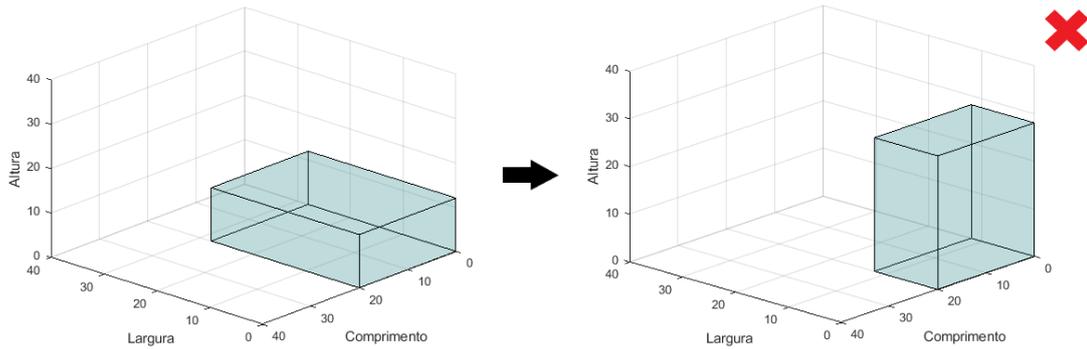
Figura 2 – Rotação nos eixos da altura e comprimento



Fonte: Elaborada pelo autor.

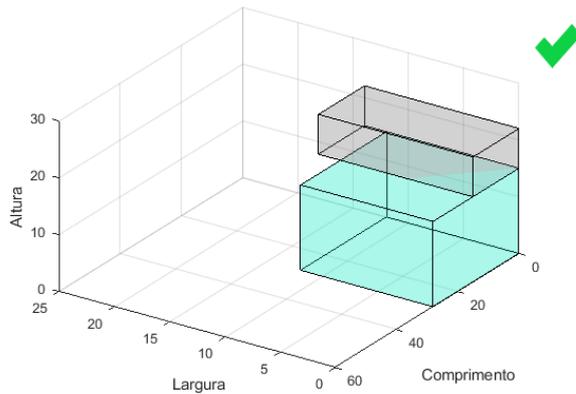
- Um parâmetro f_{ik} é utilizado para definir a fragilidade dos itens. Se $f_{ik} = 1$ informa que o item é frágil e $f_{ik} = 0$, caso contrário. Itens frágeis não podem sustentar outros itens sobre eles. Demais combinações são permitidas. As Figuras 4 e 5 detalham essa restrição. Nelas, os itens frágeis estão representados na cor cinza.

Figura 3 – Rotação nos eixos da altura e largura



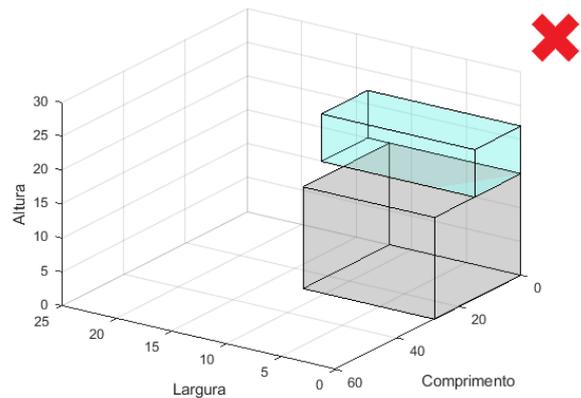
Fonte: Elaborada pelo autor.

Figura 4 – Carregamento correto em relação a fragilidade



Fonte: Elaborada pelo autor.

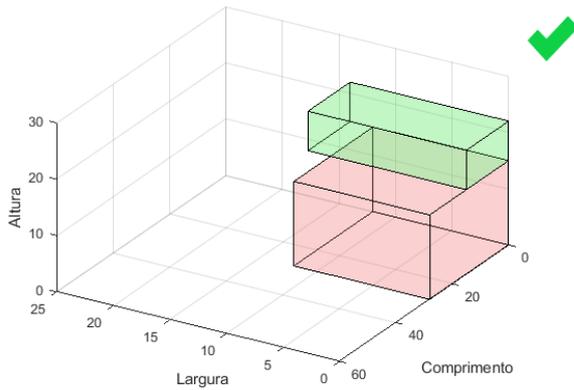
Figura 5 – Carregamento incorreto em relação a fragilidade



Fonte: Elaborada pelo autor.

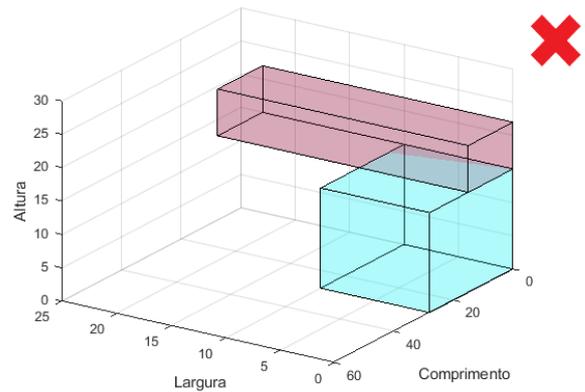
- Para garantia da estabilidade, quando um item I_{ik} é carregado, sua base deve ter uma mínima parcela suportada $\bar{A} \geq aw_{ik}l_{ik}$ (pelo topo de um outro item ou pelo piso do veículo), onde $0 < a \leq 1$ é um parâmetro que determina o mínimo da área da base que deve ser suportada. É evidente que quando um item é carregado diretamente sobre o piso do contêiner, a estabilidade do carregamento é sempre satisfeita. Um carregamento correto e incorreto em relação a estabilidade estão ilustrados pelas Figuras 6 e 7, respectivamente.
- Quando um cliente i é visitado, a retirada de todos os itens I_{ik} demandados por ele deve atender a movimentos retos e paralelos à face L (comprimento) do veículo (Figura 8) e nenhum outro item pertencente à um outro cliente poderá ser movimentado, ou seja, deve-se atender a política "o último que entra, é o primeiro que sai" (LIFO - Last In-First Out).

Figura 6 – Carregamento correto em relação a estabilidade



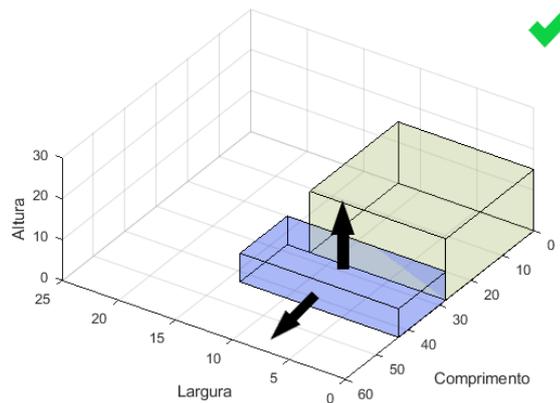
Fonte: Elaborada pelo autor.

Figura 7 – Carregamento incorreto em relação a estabilidade



Fonte: Elaborada pelo autor.

Figura 8 – Movimentos permitidos na retirada de uma caixa



Fonte: Elaborada pelo autor.

Em Ruan et al. (2013), foi definido o modelo matemático do problema. Para uma maior facilidade de entendimento, eles dividiram o modelo em dois submodelos, um para o CVRP e outro para o 3LP (*Three Dimensional Problem*). O modelo adaptado do CVRP de Ruan et al. (2013) é definido da seguinte forma:

$$\text{Minimizar } \sum_{p=1}^v \sum_{j=0}^n \sum_{i=0}^n c_{ij} y_{ij}^p \quad (20)$$

Sujeito à:

$$\sum_{p=1}^v \sum_{j=0}^n y_{ij}^p = 1, \quad i = 0, \dots, n; \quad i \neq j \quad (21)$$

$$\sum_{j=1}^n y_{0j}^p = x^p, p = 1, \dots, v \quad (22)$$

$$\sum_{j=1}^n y_{j0}^p = x^p, p = 1, \dots, v \quad (23)$$

$$\sum_{i=0}^n y_{il}^p - \sum_{j=0}^n y_{lj}^p = 0, l = 1, \dots, n; p = 1, \dots, v; l \neq i; i \neq j \quad (24)$$

$$\sum_{i=1}^n z_i^p d_i \leq D, p = 1, \dots, v \quad (25)$$

$$\sum_{j=0}^n y_{ij}^p = z_i^p, i = 0, 1, \dots, n; p = 1, \dots, v; i \neq j \quad (26)$$

$$\sum_{i,j \in U} y_{ij}^p \leq |U| - 1, p = 1, \dots, v, U \subset C \quad (27)$$

onde:

- x^p é igual a 1 se o veículo p é usado, 0 caso contrário;
- c_{ij} é o custo para ir do cliente i para o cliente j ;
- y_{ij}^p é igual a 1 se a rota (i, j) é feita pelo veículo p , 0 caso contrário;
- z_i^p é igual a 1 se os itens do cliente i são distribuídos pelo veículo p , 0 caso contrário;
- D é o peso máximo que pode ser transportado nos veículos;
- d_i é a soma dos pesos dos itens requeridos pelo cliente i ;

A função objetivo (20) é minimizar o custo da distância total percorrida. A restrição (21) garante que todos os clientes são visitados apenas uma vez. A garantia que o depósito é o primeiro e último destino dos veículos é definida nas restrições (22) e (23). A restrição (24) assegura que o cliente que o veículo chega é o mesmo que ele sai. A restrição (25) certifica que a capacidade de peso dos veículos não é excedida. A restrição (26) garante que os itens do cliente foram entregues por apenas um veículo. A restrição (27) elimina sub-rotas.

Antes da modelagem do modelo do 3LP, é necessário definir um sistema de coordenadas com sua origem no contêiner *front-left-bottom*, ou seja, mais ao fundo e mais à esquerda. Dado que (x, y, z) sejam as coordenadas possíveis onde o canto *front-left-bottom* de um item pode ser colocado. Estas posições possíveis ao longo do eixo L, W e H do contêiner pertencem ao conjunto: $X = \{0, 1, 2, \dots, L - \min_i(l_i)\}$, $Y = \{0, 1, 2, \dots, W - \min_i(w_i)\}$,

$Z = \{0, 1, 2, \dots, H - \min_i(h_i)\}$. Mesmo que o volume total de itens a serem carregados no veículo seja menor do que o volume do veículo, existe a possibilidade que alguns itens não serem carregados, devido suas diferentes formas. O objetivo do 3LP é maximizar o número de itens a serem carregados no veículo:

$$\text{Maximizar } \sum_{i=1}^{n_p} \sum_{x \in X} \sum_{y \in Y} \sum_{z \in Z} s_i a_i^{xyz} \quad (28)$$

Sujeito à:

$$\sum_{i=1}^{n_p} \sum_{x \in X} \sum_{y \in Y} \sum_{z \in Z} L_{ij} W_{ij} a_i^{xyz} \geq \rho l_j w_j a_j^{x'y'z'}, \quad j = 1, \dots, n; \quad p = 1, \dots, v \quad (29)$$

$$\sum_{i=1}^{n_p} \sum_{x \in X} \sum_{y \in Y} \sum_{z \in Z} \sigma_i a_i^{xyz} \geq \sum_{j=1}^{n_p} \sum_{x' \in X} \sum_{y' \in Y} \sum_{z' \in Z} \left(\frac{d_j}{l_j w_j} \right) a_j^{x'y'z'}, \quad p = 1, \dots, v \quad (30)$$

$$a_i^{xyz} = \begin{cases} 1 & i = 1, \dots, n_p, \quad x \in X, \quad y \in Y, \quad z \in Z \\ 0 & \text{caso contrário} \end{cases} \quad (31)$$

$$L_{ij} = \min(x + l_i, x' + l_j) - \max(x, x'), \quad j = 1, \dots, n; \quad x \in X; \quad x' \in X \quad (32)$$

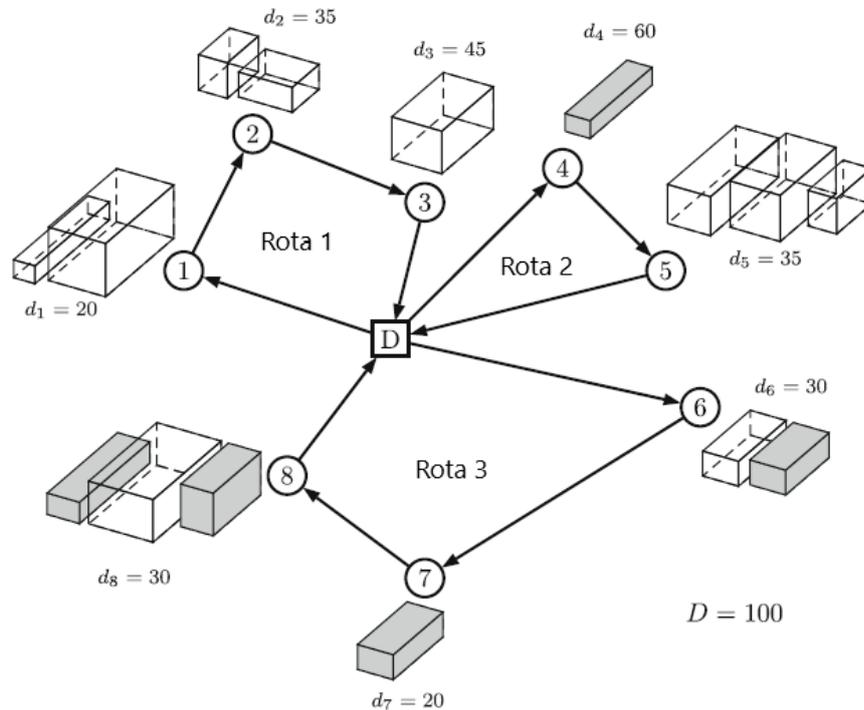
$$W_{ij} = \min(y + w_i, y' + w_j) - \max(y, y'), \quad j = 1, \dots, n; \quad y \in Y; \quad y' \in Y \quad (33)$$

onde:

- n_p é o número de itens no veículo p ;
- (x', y', z') uma possível coordenada do canto *front-left-bottom* de outro item;
- σ_i é a pressão máxima admissível que um item i pode suportar em qualquer ponto (x', y') da sua face superior.
- ρ define a proporção da área de um item que deve ser colocada sobre a área de outro item para ser considerada estável.

A função objetivo (28) maximiza o número de itens a serem carregados em cada veículo. A restrição (29) garante que todos os itens sejam apoiados por uma área de suporte sem existir sobreposição. A restrição (30) assegura a máxima pressão admissível de itens frágeis e não frágeis. $\sigma_i = 0$ define o item i como frágil, ou seja, nenhum outro item pode ser colocado em cima dele. A variável de decisão que indica se o item i está carregado é exemplificada na restrição (31). As restrições (32) e (33) definem os cálculos de comprimento e largura

Figura 9 – Exemplo de instância para o 3L-CVRP. Os itens frágeis estão acinzentados

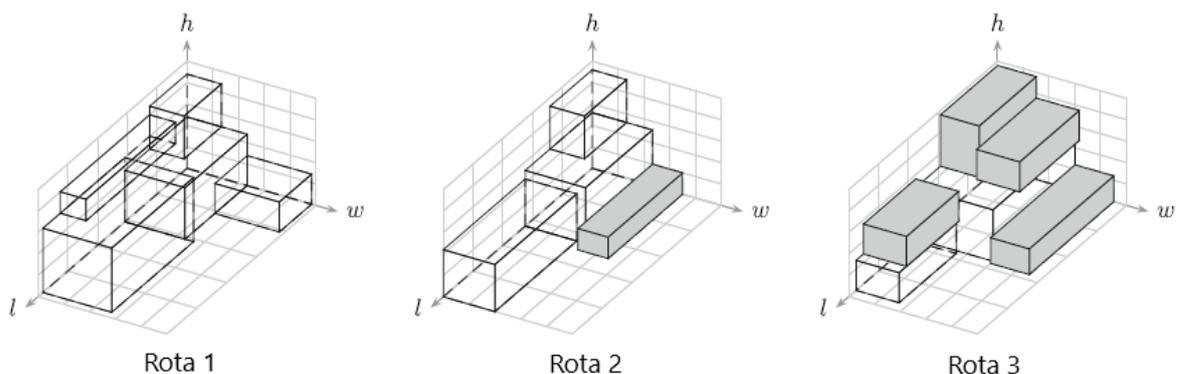


Fonte: (FUELLERER et al., 2010) (Adaptado)

para a área de apoio entre os itens i e j . A Figura 9 apresenta uma instância exemplo para o 3L-CVRP.

A quantidade de itens requeridos por cada cliente é denotado pela quantidade de caixas e o peso total demandado pelo cliente i é indicado por d_i . Os itens frágeis estão acinzentados. A capacidade dos veículos é $D = 100$. Uma solução factível de carregamento para a instância exemplo é representada pela Figura 10. Nela, não há itens carregados sobre item frágeis e a restrição de estabilidade é atendida.

Figura 10 – Solução da instância exemplo



Fonte: (FUELLERER et al., 2010) (Adaptado)

Capítulo 3

Fundamentação Teórica

Para entender melhor o problema abordado neste trabalho, é preciso conhecer alguns conceitos importantes, essencialmente os das áreas de otimização, algoritmos genéticos e heurísticas. Neste capítulo é apresentada uma breve explicação sobre eles.

3.1 Métodos de Carregamento

Na literatura, existem métodos de carregamento de itens em um veículo. Um dos mais conhecidos é o *Extreme Point-Based Heuristic* introduzido por [Crainic, Perboli e Tadei \(2008\)](#).

3.1.1 *Extreme Point-Based Heuristic*

A partir do conceito de *corner points* (CPs) ([MARTELLO; PISINGER; VIGO, 2000](#)), que são pontos onde os itens podem ser carregados sem que haja sobreposição, [Crainic, Perboli e Tadei \(2008\)](#) apresentaram o conceito de *extreme points* (EPs). Nessa abordagem, toda vez que um item é carregado no veículo, pode-se gerar até seis novos pontos (EPs) a partir da projeção do item recém inserido nos elementos já carregados no veículo. As Figuras 11 e 12 ilustram o conceito.

Em [Crainic, Perboli e Tadei \(2008\)](#) também foi apresentada uma estratégia utilizando o conceito de EPs: o *First Fit Decreasing* (FFD), no qual o item é carregado mais abaixo, ao fundo e à esquerda possível. Um exemplo de carregamento pode ser visto na [Figura 13](#).

3.2 Otimização Mono-objetivo e Multiobjetivo

Otimização é a escolha de um melhor elemento, em relação a algum critério, a partir de um conjunto de possibilidades. Um problema de otimização consiste em maximizar ou minimizar uma função, escolhendo de forma correta os valores do domínio e calculando o

Figura 11 – EPs definidos em um item
(os EPs são os triângulos)

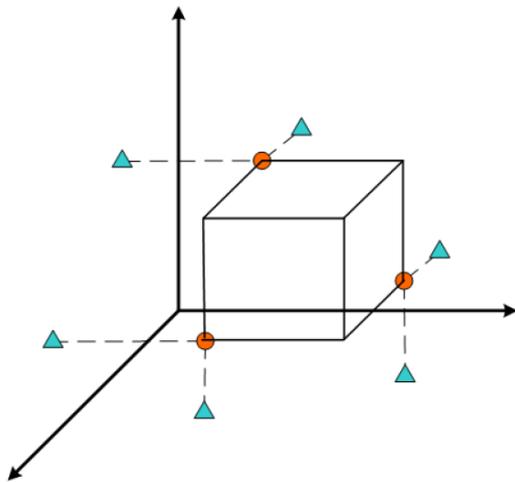
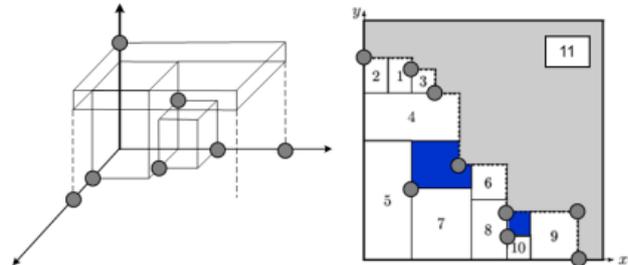


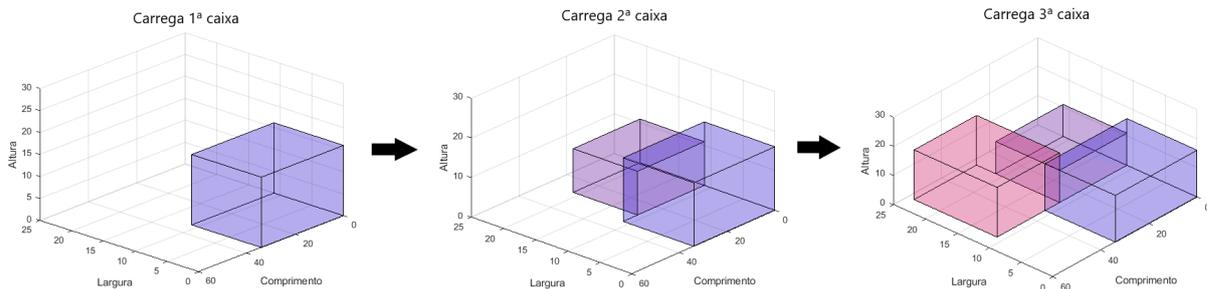
Figura 12 – Representação dos *Extreme Points* em 2D e 3D



Fonte: (CRAINIC; PERBOLI; TADEI, 2008)

Fonte: (CRAINIC; PERBOLI; TADEI, 2008)

Figura 13 – Sequência de Carregamento de Itens

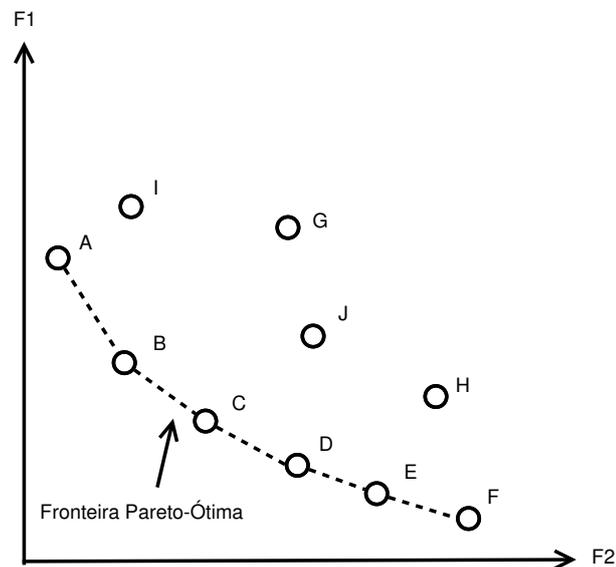


Fonte: Elaborada pelo autor.

valor da função. Na otimização mono-objetivo, tem-se apenas um objetivo a ser minimizado ou maximizado. Já em um problema de otimização multiobjetivo, há dois ou mais objetivos que devem ser otimizados simultaneamente. Porém, os objetivos são conflitantes entre si (COELLO, 2006) e podem estar sujeitos a restrições. Na otimização multiobjetivo não existem soluções ótimas no sentido de minimizarem (ou maximizarem) individualmente todos os objetivos. A característica principal é a existência de um conjunto de soluções aceitáveis que são mais eficientes que às demais. Estas soluções aceitáveis são conhecidas como soluções Pareto-ótimas ou eficientes. Para que um conjunto seja denominado Pareto-ótimo, é preciso que para cada solução, não exista nenhuma outra solução factível capaz de melhorar o valor de uma das funções objetivo do problema sem que simultaneamente cause uma piora em pelo menos um dos demais objetivos (ARROYO, 2002; AZUMA, 2011). O conjunto de soluções Pareto-ótimas é denominado fronteira Pareto-ótima, exemplificada

pela Figura 14.

Figura 14 – Representação da Fronteira Pareto-Ótima



Fonte: Elaborada pelo autor.

3.3 Heurísticas

Uma heurística é definida como sendo um algoritmo que busca uma solução factível baseada em observações do espaço de soluções, não sendo obrigatoriamente o melhor resultado, para um problema específico em um tempo computacional aceitável (DÍAZ et al., 1996).

De um modo geral, essas abordagens utilizam uma combinação de escolhas aleatórias e informações históricas dos resultados já alcançados anteriormente para se nortear e realizarem buscas pelo espaço de soluções do problema, de modo a evitar paradas precoces em ótimos locais (TEIXEIRA, 2018).

Devido ao fato das heurísticas apresentarem um bom desempenho, espera-se que elas sejam utilizadas para resolver problemas da classe NP-difícil. Porém, nenhum método heurístico garante solução ótima.

Já as meta-herísticas são heurísticas que possuem estratégias para escapar de mínimos locais. *Simulated Annealing* (KIRKPATRICK; GELATT; VECCHI, 1983), Algoritmos Genéticos (HOLLAND, 1992) e VNS (*Variable Neighborhood Search*) (MLADENOVIĆ; HANSEN, 1997) são exemplos de meta-herísticas. As heurísticas são classificadas em construtivas e de refinamento (PEARL, 1984).

3.3.1 Heurísticas Construtivas

As heurísticas construtivas são aquelas que constroem uma solução passo a passo, introduzindo um elemento de cada vez até encontrar uma solução que satisfaça o problema. Geralmente, esse tipo de heurística se comporta de forma gulosa, ou seja, o elemento escolhido para ser inserido na solução é o melhor elemento naquele momento. São usadas normalmente como solução inicial para uma meta-heurística.

3.3.2 Heurísticas de Refinamento

Heurísticas de refinamento, por sua vez, partem de uma solução inicial qualquer com o objetivo de melhorá-la ao máximo.

Esse tipo de heurística utiliza o conceito de vizinhança. Ela caminha, a cada iteração, de vizinho a vizinho de acordo com o critério de vizinhança escolhido tentando melhorar a solução construída. Essa exploração do espaço busca é feita através do conceito de movimento, isto é, uma modificação que transforma uma solução em solução na vizinhança.

As heurísticas de refinamento podem ser divididas basicamente em três tipos de estratégias de acordo o modo de exploração do espaço de buscas:

- *Best Improvement Method*: a cada passo analisa todos os possíveis vizinhos de uma solução e retorna o melhor entre todos eles. Esse método tem como ponto positivo de sempre retornar um ótimo local. Porém, como ele avalia todas as soluções vizinhas, o método pode perder desempenho caso a quantidade de vizinhos seja grande;
- *First Improvement Method*: parecido com o método anterior, no entanto, interrompe a busca pela vizinhança quando um vizinho melhor é encontrado. Este método tem como ponto positivo evitar a exploração de toda vizinhança, fato que acontece apenas no pior caso;
- *Random Descent*: consiste em escolher um vizinho aleatoriamente. Se ele apresentar uma melhora em comparação com a solução corrente, ele é aceito e se torna a nova solução. Caso contrário, um outro vizinho é gerado. Se ao final de um número fixo de iterações não houver melhora no valor da solução, o método é interrompido e retorna a melhor solução encontrada durante a busca, por esse motivo não há garantia de retornar o ótimo local.

3.3.3 Algoritmo Genético

Um dos ramos da Inteligência Artificial é a Computação Evolucionária, que propõe um modelo de solução de problemas inspirado na Seleção Natural proposta por [Darwin](#) e

[Wallace \(1858\)](#).

Esse modelo compreende um conjunto de técnicas de busca e otimização inspiradas na evolução natural das espécies. Inicialmente, cria-se uma população inicial de indivíduos, que vão reproduzir e competir pela sobrevivência entre si. Os melhores sobrevivem e transferem suas características genéticas a novas gerações. Entre outras técnicas, uma das mais utilizadas são os Algoritmos Genéticos (AGs) ([HOLLAND, 1992](#)).

Os AGs empregam uma terminologia originada da teoria da evolução natural e da genética. Um indivíduo da população é representado por um único cromossomo, o qual contém a codificação (genótipo) de uma possível solução do problema (fenótipo). Cromossomos são usualmente implementados na forma de listas de atributos ou vetores, onde cada atributo é conhecido como gene. Os possíveis valores que um determinado gene pode assumir são denominados alelos ([ZUBEN, 2000](#)).

A ideia de funcionamento dos algoritmos genéticos é olhar as possíveis soluções do problema como indivíduos de uma população, a qual passará por um processo evolutivo a cada iteração ou geração. A execução do algoritmo acontece da seguinte forma:

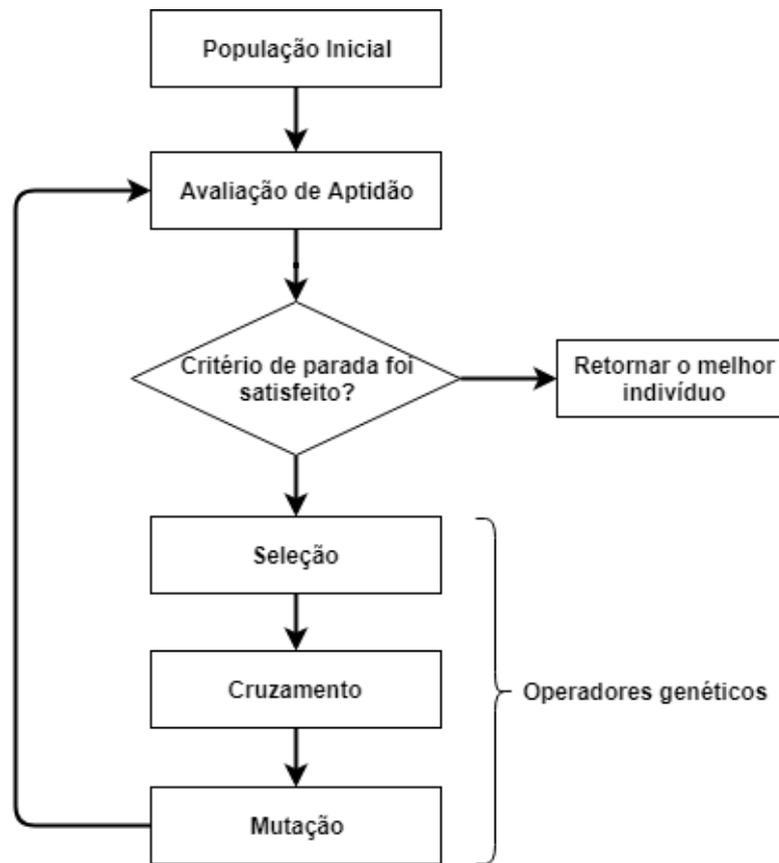
- Primeiramente gera-se uma população inicial, normalmente de forma aleatória;
- Avalia-se toda a população de indivíduos através de uma função que mede a qualidade do indivíduo (função de aptidão ou *fitness*);
- Em seguida, através do operador de seleção, escolhem-se os melhores indivíduos (a partir da função aptidão) como base para criação de um novo conjunto de soluções;
- Este novo conjunto é obtido aplicando-se sobre os indivíduos selecionados operações que misturem suas características com o objetivo de diversificar a população. Esse processo é feito através dos operadores de cruzamento (*crossover*) e mutação;
- Estes passos são repetidos até que um critério de parada seja alcançado. Esse critério pode ser a solução aceitável encontrada, um número predeterminado de passos ou quando o algoritmo não consiga mais melhorar a solução já encontrada.

Uma melhor visualização do funcionamento do método pode ser visto no [Algoritmo 1](#) e na [Figura 15](#).

3.3.3.1 Definição da População Inicial

A maneira mais comum utilizada na criação da população é a inicialização aleatória dos indivíduos. Se algum conhecimento inicial a respeito do problema for sabido, ele pode ser

Figura 15 – Estrutura básica de um algoritmo genético



Fonte: Elaborada pelo autor.

utilizado na inicializa o da popula o. Por exemplo, no caso de uma codifica o bin ria, se tem a informa o que a solu o final vai apresentar mais 0's do que 1's, ent o ela pode ser utilizada com o objetivo de gerar um maior n mero de indiv duos iniciais que tenham uma maior probabilidade de representarem uma solu o final, mesmo que n o se saiba exatamente a propor o (KRUSE et al., 2016).

3.3.3.2 Sele o

Dada uma popula o em que cada indiv duo foi atribu do um valor de aptid o,   preciso selecionar os indiv duos sobre os quais ser o aplicados os operadores gen ticos. Existem muitas formas de sele o, entre eles os mais utilizados s o os m todos de sele o por roleta e o m todo de sele o por torneio (HOLLAND, 1992).

No m todo de sele o por roleta, cada indiv duo da popula o   representado na roleta proporcionalmente ao seu  ndice de aptid o. Assim, para indiv duos com elevada aptid o   dada uma por o maior da roleta, enquanto aos indiv duos de aptid o mais baixa,   dada uma por o relativamente menor. Ap s a distribui o dos indiv duos, gira-se a roleta um determinado n mero de vezes, e s o escolhidos, como indiv duos que participar o da

Algoritmo 1: Algoritmo Genético

Entrada: Variáveis do problema**Saída:** Uma solução s^* otimizada

```
1 início
2   Gera uma população inicial  $P$ 
3   Avalia indivíduos de  $P$ 
4   enquanto condição de parada não for satisfeita faça
5      $P' \leftarrow$  seleção( $P$ )
6      $P'' \leftarrow$  cruzamento( $P'$ )
7      $P''' \leftarrow$  mutação( $P''$ )
8     Substitui  $P$  por  $P'''$ 
9     Avalia indivíduos de  $P$ 
10  fim
11   $s^* \leftarrow$  melhor indivíduo de  $P$ 
12 fim
13 retorna  $s^*$ 
```

próxima geração, aqueles sorteados na roleta (HOLLAND, 1992).

O outro método é a seleção por torneio. A ideia deste método é promover um torneio entre um grupo de n indivíduos escolhidos aleatoriamente. Então, o indivíduo que vencer este torneio (indivíduo com melhor valor de aptidão), é selecionado para ser aplicado os operadores genéticos (HOLLAND, 1992).

3.3.3.3 Cruzamento

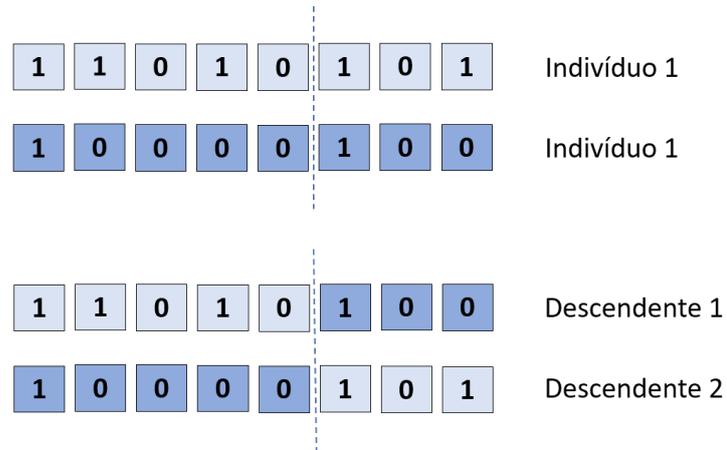
O cruzamento ou *crossover* (HOLLAND, 1992) cria novos indivíduos através da combinação de dois ou mais indivíduos. A ideia por trás do *crossover* é a troca de informação entre diferentes soluções candidatas. O resultado desta operação é um indivíduo que potencialmente combine as melhores características dos indivíduos usados como base.

Os tipos de cruzamento mais utilizados são o cruzamento em um ponto e o cruzamento em dois pontos, mostrados nas Figuras 16 e 17, respectivamente.

Quando se tem um ponto de cruzamento, seleciona-se aleatoriamente um ponto de corte do cromossomo. Um dos filhos recebe as características do início até o ponto de corte de um dos pais e do ponto de corte até o final do cromossomo do outro pai. Para o outro filho, repete-se o procedimento, porém invertendo os pais (Figura 16).

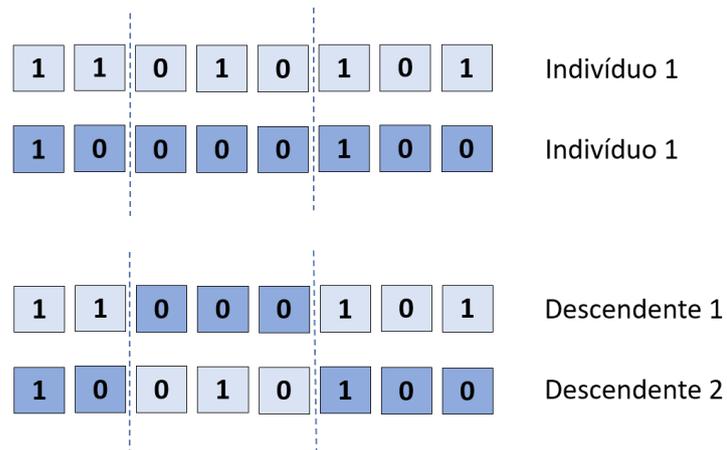
Com dois pontos de cruzamento, delimita-se uma parte específica do cromossomo. Selecionado os dois pais, essa parte delimitada é trocada entre eles, gerando assim, dois novos descendentes. No caso da Figura 17, a parte central do cromossomo dos indivíduos selecionados é trocada entre eles.

Figura 16 – Cruzamento em um ponto



Fonte: Elaborada pelo autor.

Figura 17 – Cruzamento em dois pontos



Fonte: Elaborada pelo autor.

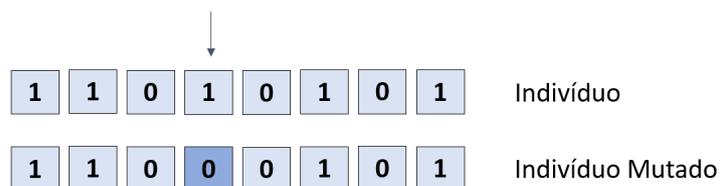
3.3.3.4 Mutação

Esta operação modifica aleatoriamente alguma característica do indivíduo sobre qual é aplicada. Esse operador é importante, pois cria novos valores de características que não existiam ou apareciam em pequena quantidade na população, contribuindo para a introdução e manutenção da diversidade da população. Desta forma, a mutação assegura a possibilidade de todo o espaço de busca ser explorado. Este operador é aplicado aos indivíduos através de uma taxa de mutação geralmente pequena (HOLLAND, 1992). Na Figura 18, um indivíduo sofre mutação alterando-se o gene sorteado de 1 para 0.

3.3.4 Algoritmo NSGA-II

O algoritmo NSGA-II (*Non-Dominated Sorting Genetic Algorithm II*) (Deb et al., 2002) é um algoritmo multiobjetivo baseado em Algoritmos Genéticos e que utiliza o conceito de

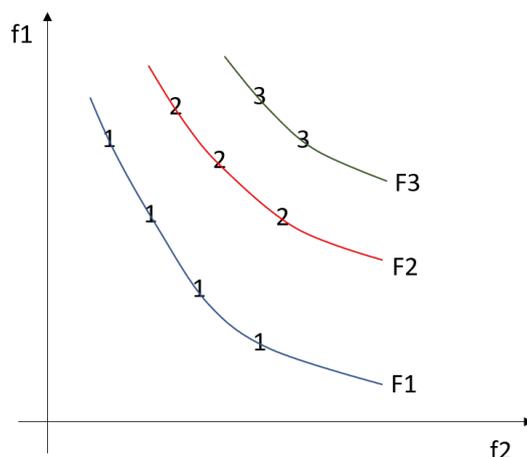
Figura 18 – Mutaç o



Fonte: Elaborada pelo autor.

domin ncia, ou seja, classifica a popula o total em *fronts* de acordo com o grau de domin ncia, como pode ser visto na Figura 19. Solu es n o-dominadas s o caracterizadas por n o existir uma outra solu o admiss vel que melhore simultaneamente todos os objetivos. Com isso, o *front* 1   composto pelas solu es n o-dominadas dentro da popula o. O *front* 2, por sua vez, integram os indiv duos que s o dominados apenas pelos indiv duos do *front* 1 e que dominam todas as solu es dos *fronts* consecutivos. Os demais indiv duos pertencentes aos *fronts* restantes s o organizados seguindo a mesma l gica dita acima.

Figura 19 – Representa o dos *fronts* do algoritmo NSGA-II



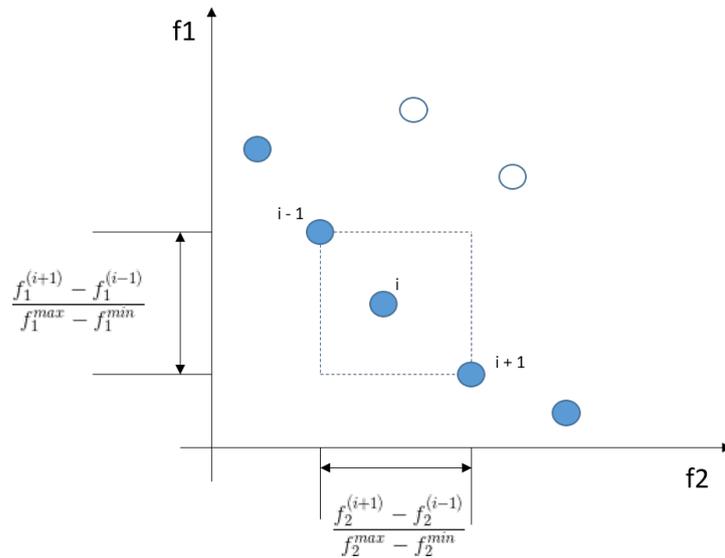
Fonte: Elaborada pelo autor.

Juntamente a conceito de domin ncia, o algoritmo utiliza o c lculo de dist ncia de multid o para ordenar os indiv duos pertencentes a um mesmo *front*. Esse c lculo refere-se a m dia da dist ncia das solu es adjacentes a um dado indiv duo para todos os objetivos considerados. O c lculo da dist ncia de multid o para cada indiv duo i   dado na Equa o 34, onde m   o n mero de objetivos, $f_k^{(i+1)}$ e $f_k^{(i-1)}$ correspondem respectivamente aos valores da fun o objetivo k do sucessor e predecessor do indiv duo i . f_k^{max} e f_k^{min} s o os valores m ximos e m nimos da fun o objetivo k (Deb et al., 2002).

$$c^i = \sum_{k=1}^m \frac{f_k^{(i+1)} - f_k^{(i-1)}}{f_k^{max} - f_k^{min}} \quad (34)$$

Uma melhor visualização dessa equação pode ser vista na [Figura 20](#). Esse cálculo tem como vantagem em comparação com outros métodos que o antecederam pelo fato de não ser necessário a determinação de outros parâmetros além dos valores dos objetivos para cada indivíduo ([KATAOKA et al., 2017](#)).

Figura 20 – Representação da distância de multidão

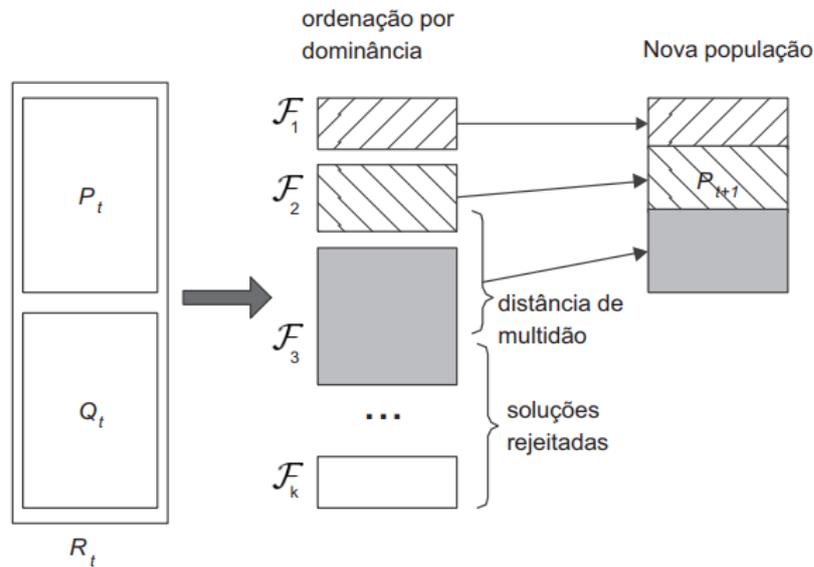


Fonte: ([Deb et al., 2002](#)) (Adaptado)

O algoritmo trabalha com duas populações, chamadas P e Q cada uma de tamanho N . Na primeira geração, os indivíduos da população P_1 são gerados aleatoriamente, e as soluções em Q_1 são geradas a partir de P_1 utilizando os operadores genéticos clássicos: seleção (torneio), cruzamento e mutação. Em seguida é formada a população combinada $R_1 = P_1 \cup Q_1$. Então a população R_1 é ordenada por não-dominância e classificada em fronteiras. Para preencher as N vagas da próxima geração, nesse caso P_2 , um processo competitivo é feito com a população de R_1 . As soluções não-dominadas pertencentes a primeira fronteira da população combinada, ou seja, de F_1 , são as melhores entre todas as soluções em R_1 . Se o tamanho de F_1 for maior que N , realiza-se o cálculo da distância de multidão para decidir aqueles que passarão para a próxima geração. Caso contrário, todos os indivíduos de F_1 integrarão a nova população.

Procede-se do mesmo modo com as fronteiras subsequentes até uma dada fronteira F_x , cujos indivíduos, se tomados totalmente, ultrapassarão o total de P_2 . Então, realiza-se o cálculo da distância de multidão, selecionando os indivíduos com os maiores resultados nesse quesito para completar P_2 . Esse mecanismo favorece as fronteiras iniciais, destacando o caráter elitista do NSGA-II ([KATAOKA et al., 2017](#)). O procedimento completo em uma geração pode ser visto na [Figura 21](#) e no [Algoritmo 2](#).

Figura 21 – Esquema do algoritmo NSGA-II



Fonte: (TICONA; DELBÉM, 2008)

3.3.5 Algoritmo VNS

O VNS (*Variable Neighborhood Search*) (MLADENOVIĆ; HANSEN, 1997), também chamado como Busca em Vizinhança Variável (BVV), é um método de busca local que consiste em explorar o espaço de soluções através de trocas sistêmicas de estruturas de vizinhança. Se opondido a outras meta-heurísticas baseadas em métodos de busca local, o algoritmo VNS não percorre um caminho, mas sim explora vizinhanças cada vez mais complexas e “distantes” da solução corrente, focando a busca em torno de uma nova solução se, e somente se, um movimento de melhora é realizado (CUNHA et al., 2008).

A vizinhança de uma solução são todas as alterações que podem ser realizadas na solução corrente, de tal forma a gerar uma nova solução distinta. Uma estrutura de vizinhança é uma forma pré-estabelecida de se percorrer o espaço de soluções vizinhas à solução atual.

O Algoritmo 3 apresenta o pseudocódigo da meta-heurística VNS. Nele, parte-se de uma solução inicial qualquer e, a cada iteração, seleciona-se aleatoriamente um vizinho y dentro da vizinhança $N_k(x^*)$ obtida através da estrutura de vizinhança de ordem k ($1 \leq k \leq k_{max}$) a partir da solução x^* corrente. Esse vizinho y é então submetido a um procedimento de busca local. Se a nova solução obtida y' , resultante dessa busca local no entorno de y , for melhor que a solução corrente x^* , a busca prossegue a partir da nova solução encontrada y' , reiniciando a partir da primeira estrutura de vizinhança $N_1(x^*)$. Caso contrário, a busca continua a partir da próxima estrutura de vizinhança $N_{k+1}(x^*)$. Este procedimento termina quando uma condição de parada for atingida, que pode ser tempo de processamento ou número máximo de iterações, por exemplo. Para aplicação do VNS, é necessária a geração

Algoritmo 2: NSGA-II

Entrada: Conjunto de parâmetros necessários para o algoritmo

Saída: Conjunto de soluções eficientes para o problema

```
1 início
2   Criar uma população inicial  $P_1$  de  $N_{ind}$  indivíduos
3   Ordenar  $P_1$  pelo critério de dominância
4   Gerar uma nova população  $Q_1$  de tamanho  $N_{ind}$  aplicando operadores genéticos em
    $P_1$ 
5   para cada geração  $t = 1, \dots, N_{iter}$  faça
6     Ordenar por dominância  $R_t = P_t \cup Q_t$ 
7      $k = 1$ 
8     enquanto  $|P_{t+1} + \mathcal{F}_k| \leq N_{iter}$  faça
9       Calcular distância de multidão para  $\mathcal{F}_k$ 
10       $P_{t+1} = P_{t+1} \cup \mathcal{F}_k$ 
11       $k = k + 1$ 
12     fim
13     Calcular distância de multidão para  $\mathcal{F}_k$ 
14     Classificar a  $\mathcal{F}_k$  pelo ranking e a distância de multidão
15     Copiar as primeiras  $N_{ind} - |P_{t+1}|$  posições de  $\mathcal{F}_k$  para  $P_{t+1}$ 
16     Aplicar operadores genéticos em  $P_{t+1}$  para gerar uma nova população  $Q_{t+1}$ 
17   fim
18    $R_{final} = P_{t+1} \cup Q_{t+1}$ 
19 fim
```

Algoritmo 3: VNS

Entrada: Conjunto N_k de estruturas de vizinhança, $k = 1, \dots, k_{max}$; solução inicial $x \in S$

Saída: Uma solução x^* otimizada do problema

```
1 início
2    $x^* \leftarrow x$ 
3    $f^* \leftarrow f(x)$ 
4   enquanto condição de parada não for satisfeita faça
5      $k \leftarrow 1$ 
6     enquanto  $k \leq k_{max}$  faça
7       Gere um vizinho qualquer  $y \in N_k(x^*)$ 
8       Aplique um método de busca local em  $y$  para obter um ótimo local  $y'$ 
       // para problemas de minimização
9       se  $f(y') < f^*$  então
10         $x^* \leftarrow y'$ 
11         $f^* \leftarrow f(y')$ 
12         $k \leftarrow 1$ 
13      senão
14         $k \leftarrow k + 1$ 
15      fim
16    fim
17  fim
18 fim
19 retorna  $x^*$ 
```

de uma solução inicial.

3.3.6 Simulated Annealing

O algoritmo *Simulated Annealing* (Recozimento Simulado) (KIRKPATRICK; GELATT; VECCHI, 1983) é uma meta-heurística que consiste em uma busca local probabilística fundamentada numa analogia com a termodinâmica. Este algoritmo é uma metáfora de um processo térmico, chamado recozimento, utilizado em metalurgia para obtenção de estados de baixa energia num sólido. Uma característica importante deste método é a possibilidade de realizar movimentos que não são de melhora de acordo com a temperatura corrente, permitindo uma maior exploração do espaço de buscas.

Algoritmo 4: Simulated Annealing

Entrada: Conjunto N de estruturas de vizinhança, solução inicial s , $SAmax$, α , T_0

Saída: Uma solução s^* otimizada do problema

```
1 início
2    $s^* \leftarrow s$ 
3    $IterT \leftarrow 0$ 
4    $T \leftarrow T_0$ 
5   enquanto  $T > 0.0001$  faça
6     enquanto  $IterT < SAmax$  faça
7        $IterT \leftarrow IterT + 1$ 
8       Gerar um vizinho ( $s'$ ) aleatoriamente na vizinhança  $N(s)$ 
9        $\Delta = f(s') - f(s)$ 
10      se  $\Delta < 0$  então
11         $s \leftarrow s'$ 
12        // para problemas de minimização
13        se  $f(s') < f(s^*)$  então
14           $s^* \leftarrow s'$ 
15        fim
16      senão
17        Tome  $x \in [0, 1[$ 
18        se  $x < e^{-\Delta/T}$  então
19           $s = s'$ 
20        fim
21      fim
22     $T = T * \alpha$ 
23     $IterT = 0$ 
24  fim
25 fim
26 retorna  $s^*$ 
```

No Algoritmo 4 é apresentado o pseudocódigo do *Simulated Annealing*. Ele recebe um conjunto N_k de estrutura de vizinhanças, uma solução inicial s , uma quantidade máxima

de iterações para cada temperatura SA_{max} , a taxa de redução da temperatura α e a temperatura inicial T_0 . Inicialmente atribui a solução a ser retornada como a melhor solução e a temperatura corrente como a inicial. Enquanto a temperatura corrente não atingir um valor mínimo, geralmente 0.0001, ele executa SA_{max} iterações, onde a cada iteração, é sorteado uma solução s' da vizinhança N . Calcula-se Δ a partir do valor das soluções, s' e s . Para problemas de minimização, se Δ for menor que zero, ou seja, $f(s')$ for menor que $f(s)$, $f(s')$ passa a ser a solução corrente, e se for menor que s^* , é atribuída como a melhor solução encontrada até então. Se Δ for maior ou igual a zero, sorteia-se um número entre zero e um, e verifica se $e^{-\Delta/T}$ é menor que o sorteado. Se sim, s' é atribuída a s . Após as SA_{max} iterações, a temperatura é atualizada a partir de α , e então, recomeça o ciclo com a nova temperatura.

No início do processo, a temperatura é elevada aumentando a probabilidade de se aceitar soluções piores. Essa característica tem como objetivo escapar de ótimos locais. No decorrer do algoritmo, a temperatura diminui, reduzindo assim, movimentos de piora. Ao final do processo, como a temperatura é muito pequena, praticamente não se aceita perturbações que piore a solução.

3.3.7 Busca Tabu

A Busca Tabu (GLOVER, 1986; HANSEN, 1986) é uma meta-heurística de construção de vizinhanças com a capacidade de fugir de ótimos locais. Ela consiste em explorar o espaço de buscas, passando de uma solução para seu melhor vizinho, mesmo que isso resulte uma piora no valor da função objetivo.

Para evitar o ciclismo, as soluções já exploradas são proibidas para uma certa quantidade de iterações. Um movimento feito a partir de uma solução é chamada tabu, onde a duração do tabu é escolhida de forma aleatória em um intervalo pré-definido. Assim, a lista tabu é uma fila que contém movimentos proibidos. Sempre que um movimento é feito, ele é inserido no final da lista tabu, e então, ela é atualizada. Um movimento de melhora não será aceito se estiver na lista tabu, a menos que satisfaça o critério de aspiração. Este é um parâmetro que tem a finalidade de retirar a regra de proibição (o *status* tabu de um movimento) sob certas condições, por exemplo, caso for gerada uma solução melhor que a melhor solução encontrada até então (BARBAROSOGLU; OZGUR, 1999).

O Algoritmo 5 apresenta o pseudocódigo da meta-heurística Busca Tabu. O algoritmo inicia-se a partir de uma solução do problema previamente definida. Enquanto a solução não melhorar por um número determinado de iterações, estabelecida por BT_{max} , busca o melhor elemento pertencente a vizinhança de s , chamado s' , de modo que o movimento que originou esta solução não seja tabu ou que atenda o critério de aspiração. Definida s' , atualiza a lista tabu com o movimento feito e remove da lista os movimentos "vencidos". Se

Algoritmo 5: Busca Tabu

Entrada: Conjunto N de estruturas de vizinhança, solução inicial s , $BTmax$ **Saída:** Uma solução s^* otimizada do problema

```
1 início
2    $s^* \leftarrow s$ 
3    $Iter \leftarrow 0$ 
4    $MelhorIter \leftarrow 0$ 
5    $T \leftarrow \emptyset$ 
6   Inicialize a função de aspiração  $A$ 
7   enquanto  $Iter - MelhorIter \leq BTmax$  faça
8      $Iter \leftarrow Iter + 1$ 
9     Seja  $s' \leftarrow s \oplus m$  o melhor elemento de  $V \subseteq N(s)$  tal que o movimento  $m$  não
      seja tabu ( $m \notin T$ ) ou  $s'$  atenda a condição de aspiração ( $f(s') < A(f(s))$ )
10    Atualize a lista tabu  $T$ 
11     $s \leftarrow s'$ 
12    se  $f(s) < f(s^*)$  então
13       $s^* \leftarrow s$ 
14       $MelhorIter \leftarrow Iter$ 
15    fim
16    Atualize a função de aspiração  $A$ 
17  fim
18 fim
19 retorna  $s^*$ 
```

s' for melhor do que a melhor solução encontrada até então, ela passa a ser s^* , a melhor iteração é atualizada para corrente, e por fim, a função de aspiração é modificada.

Capítulo 4

Metodologia

Para resolver o problema 3L-CVRP utilizou-se duas abordagens distintas, mono-objetivo e multiobjetivo. Fez-se a implementação do NSGA-II, Algoritmo Genético, VNS, Busca Tabu e *Simulated Annealing* utilizando a linguagem de programação JAVA, e nos três últimos algoritmos, aplicando diferentes soluções iniciais. Ao longo do capítulo são apresentados os detalhes da implementação.

4.1 Modelagem Mono-objetivo

Nessa abordagem soluciona-se o 3L-CVRP considerando o objetivo de minimizar a distância total percorrida pelos veículos. Para isso, foram implementados os algoritmos: Algoritmo Genético, VNS, *Simulated Annealing* e Busca Tabu. Os detalhes de implementação são detalhados ao longo da seção.

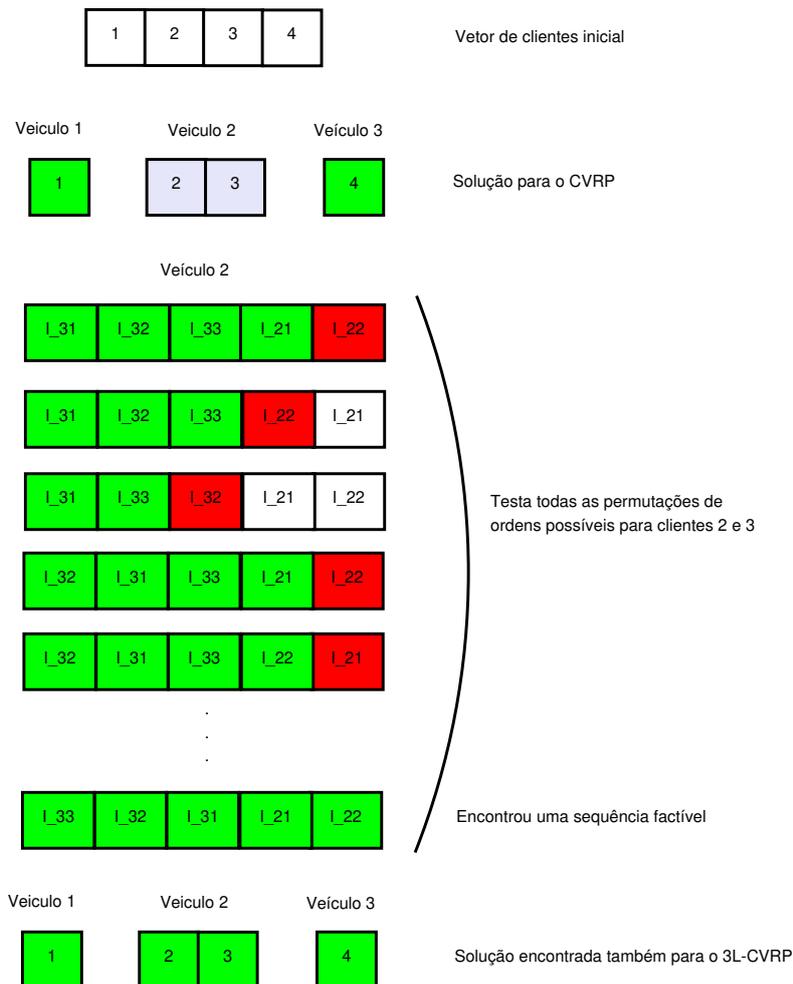
4.1.1 Algoritmo de Carregamento

O carregamento de itens dentro de um veículo deve respeitar várias restrições, as quais foram exemplificadas na [Seção 2.3](#). Dado os itens que devem ser colocados em um veículo, é preciso verificar se há um carregamento factível. Para isso, ordena-se as caixas de um cliente em ordem lexicográfica decrescente (volume, 1 - fragilidade). Usa-se essa estratégia pois itens maiores e não-frágeis mais próximos ao chão permitem um maior aproveitamento de espaço na direção do eixo da altura do veículo.

Posteriormente, tenta-se carregar todos os itens de acordo com essa ordem, utilizando o conceito de *Extreme Points* e a estratégia *First Fit Decreasing*. Para as instâncias utilizadas, a quantidade de itens por cliente geralmente é pequena, assim como a quantidade de veículos disponíveis. Por isso, caso o carregamento seja infactível, tenta-se as $n!$ possíveis permutações de ordem, sendo n o número de itens do primeiro cliente a ser visitado. Caso essa nova ordem de carregamento não seja factível, repete-se o procedimento para o

segundo cliente visitado, juntamente com as permutações do primeiro cliente, como pode ser visto na [Figura 22](#). Nessa abordagem gera-se $(n! * m)$ permutações de carregamento de itens, sendo m o número de clientes atendidos pelo veículo. Essa estratégia é computacionalmente aceitável para as instâncias testadas, pois os maiores valores que n e m podem assumir são três e seis, respectivamente.

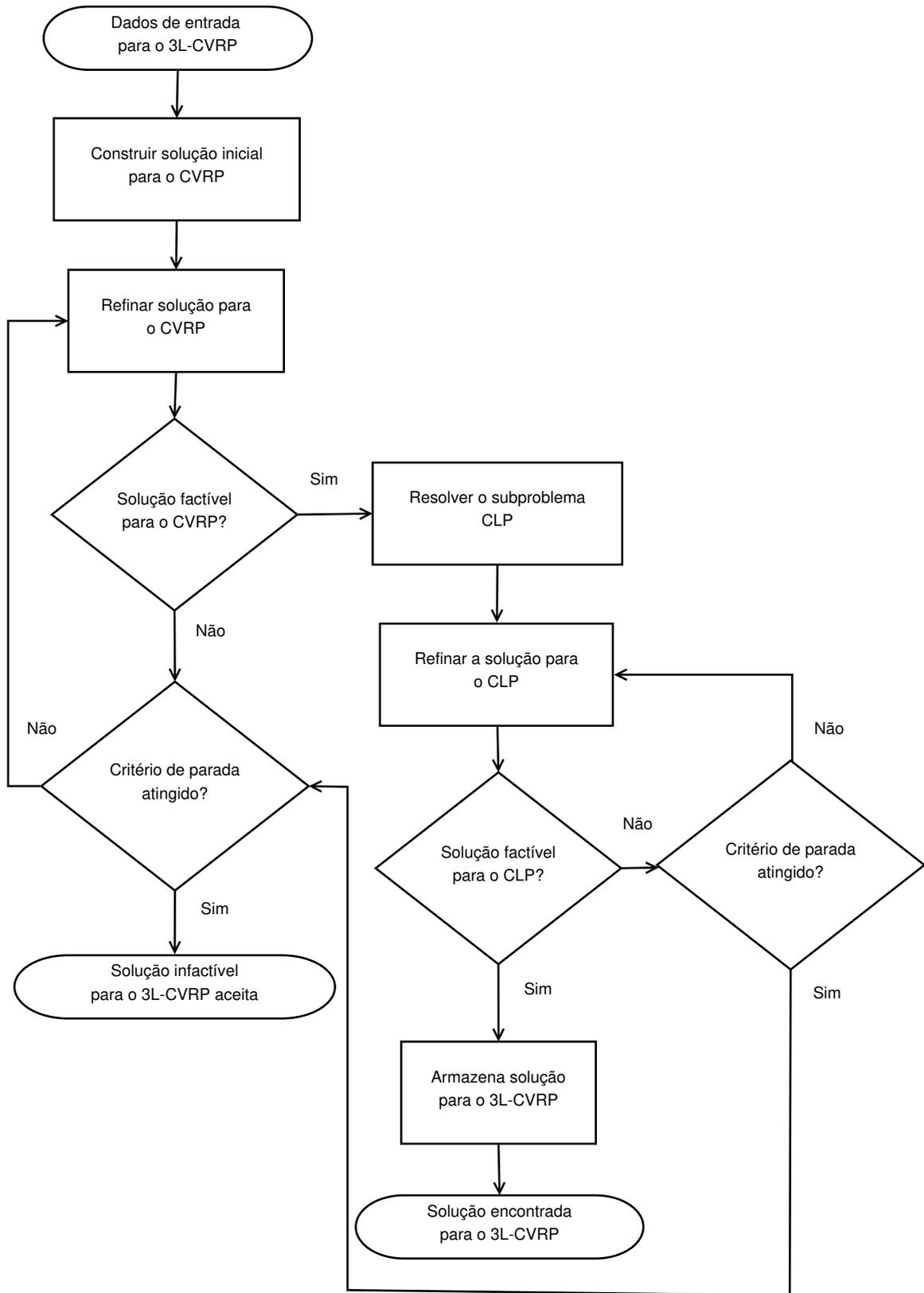
Figura 22 – Algoritmo de carregamento factível. Os quadrados de cor verde são clientes e itens em que se obteve uma solução factível. Os de cor vermelho são infactíveis. Os de cor branca são clientes e itens no qual não foram feitas tentativas de carregamento



Fonte: Elaborada pelo autor.

Se por ventura, ainda não for possível encontrar uma solução factível, inverte-se a rota, já que a distância total percorrida é a mesma, e verifica o carregamento novamente. Se uma solução factível não for encontrada, permuta-se a ordem de clientes visitados para sair da infactibilidade, mesmo que piore o valor da função objetivo. A [Figura 23](#) exemplifica a estratégia.

Figura 23 – Estratégia de carregamento



Fonte: Elaborada pelo autor.

4.1.2 Algoritmo Genético

Na implementação de um algoritmo genético é necessário definir a população inicial, a codificação e decodificação do indivíduo, o cálculo da função objetivo e os tipos de operadores. Sendo assim, as próximas seções detalham melhor a definição desses parâmetros.

4.1.2.1 Codificação

O problema 3L-CVRP foi codificado no AG em indivíduos como uma lista de clientes. Essa lista representa a ordem em cada cliente é visitado supondo que a rota é feita por um único veículo. A [Figura 24](#) mostra um exemplo de codificação de um cromossomo, no qual o primeiro cliente visitado é o cliente 1, seguido pelo 2 e assim em diante, até concluir a rota com o cliente 9. O cliente 0 não é representado pois ele é o depósito.

Figura 24 – Representação de um indivíduo do AG

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

Fonte: Elaborada pelo autor.

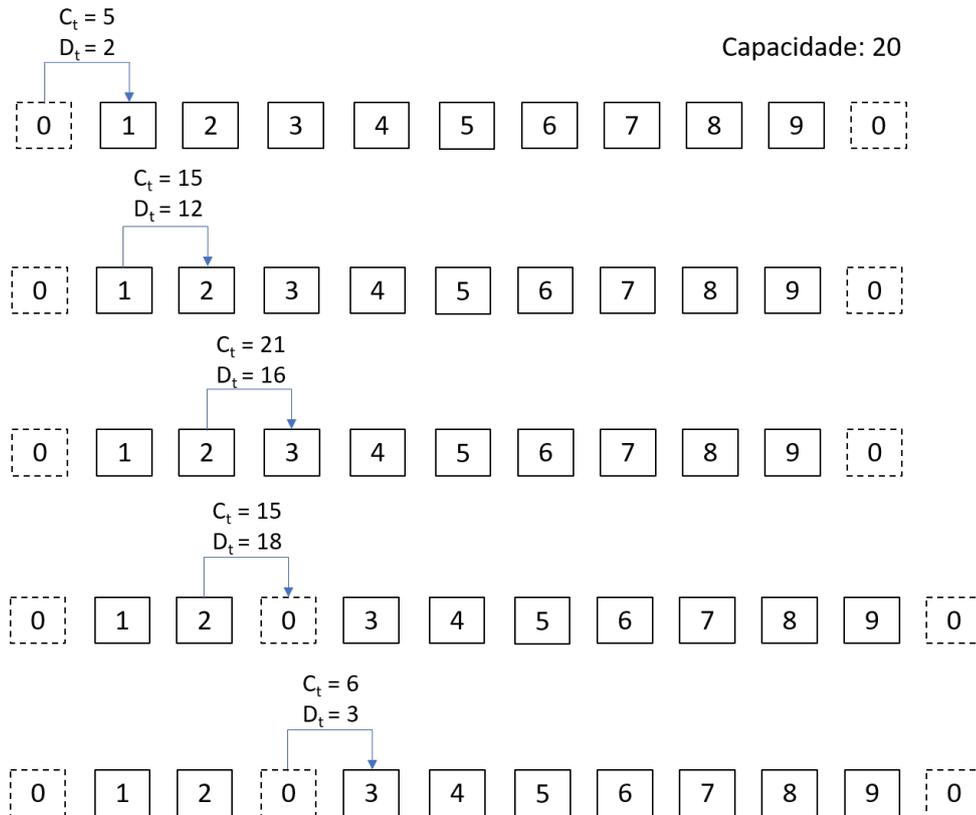
4.1.2.2 Decodificação e Avaliação do Indivíduo

Definida a codificação do indivíduo, a sequência de clientes é decodificada utilizando um algoritmo que percorre toda a lista. Dessa forma, o algoritmo verifica se existe um carregamento factível para aquela sequência de clientes respeitando a quantidade de veículos disponíveis. Inicialmente, é feito o carregamento para o problema do CVRP, considerando apenas a demanda de cada cliente e a capacidade dos veículos. A [Figura 25](#) ilustra esse carregamento, no qual inicialmente alocam-se os clientes 1 e 2 no primeiro veículo. Porém quando tenta-se alocar o cliente 3, ultrapassa a capacidade C do veículo 1. Em decorrência disso, é necessário usar um novo veículo. Feito o carregamento, verifica-se as restrições tridimensionais da solução do CVRP de cada veículo, utilizando a estratégia descrita na [Subseção 4.1.1](#), para confirmar a factibilidade também para o 3L-CVRP. Se ao final, não for possível encontrar uma solução válida, ela é mantida na população, porém penalizada de acordo com a quantidade de veículos com carregamento infactível e o excedente de veículos utilizados. O indivíduo é avaliado de acordo com a distância total D percorrida pelos veículos. Ou seja, quanto menor a distância, melhor é o indivíduo.

4.1.2.3 Função *Fitness*

A função *fitness*, também chamada de função de aptidão, avalia o quanto uma solução é boa. No caso de problemas de minimização, quanto menor o *fitness*, melhor é o indivíduo. A função objetivo é uma equação do problema que avalia a qualidade da solução. Em alguns casos, a função de aptidão é igual a função objetivo. Neste trabalho, como trabalhamos com

Figura 25 – Decodificação de um indivíduo do AG para o CVRP



Fonte: Elaborada pelo autor.

soluções infactíveis, o *fitness* é igual a função objetivo mais uma penalização caso exista veículos com carregamento infactível ou o uso de um número maior de veículos do que o disponível, de acordo com a [Equação 35](#).

$$f = \sum_{i=1}^V c_i + 1000x + 10y \quad (35)$$

onde:

- V é o número total de veículos utilizados na solução;
- c_i é a distância total percorrida pelo veículo i ;
- x é a quantidade de veículos com carregamento infactível;
- y é a quantidade de veículos utilizados além do disponível.

4.1.2.4 População Inicial

No carregamento de itens com restrições tridimensionais às vezes não é possível ocupar todo o volume disponível no veículo, devido as várias restrições que o problema deve atender. Sendo assim, a população inicial do AG é realizada de duas formas: utilizando as soluções resolvidas por programação linear inteira (PLI) utilizando o pacote comercial CPLEX para resolver o CVRP e utilizando a fase construtiva do GRASP (*Greedy Randomized Adaptive Search Procedure*) (FEO; RESENDE, 1995).

Na primeira estratégia, utiliza-se o CPLEX para resolver o modelo exato do CVRP adicionando uma nova restrição para ele limitar a capacidade ocupada dos veículos, já que no carregamento não é possível ocupar 100% do volume devido as características dimensionais das caixas. Desta forma, é necessário definir uma taxa máxima de ocupação para definir a nova restrição do modelo. Neste caso, foram definidas as seguintes porcentagens de volume: 95%, 90%, 85%, 80% e 75%.

Na segunda estratégia, para construção do indivíduo, inicialmente calcula-se a distância de todos os clientes para o depósito (índice 0). A partir disso, constrói-se a RCL (*Restrict Candidate List*) utilizando a Equação 36 para definir se o cliente i deve ser inserido na lista, considerando a maior e menor distância para o depósito dentre os clientes não inseridos no indivíduo e um parâmetro α que define o nível de gulosidade do algoritmo. Construída a RCL, escolhe aleatoriamente um cliente para ser adicionado ao indivíduo. Repete-se o procedimento a partir do último cliente inserido até que o indivíduo seja construído.

$$dist_para_cliente_i \leq (distMin + \alpha(distMax - distMin)) \quad (36)$$

Na Equação 36, o α quanto mais próximo de 1, maior é a aleatoriedade da seleção e quanto mais próximo de 0, mais gulosa ela se torna. Ao substituir α por 1, resulta em $dist_para_cliente_i \leq distMax$, ou seja, todos os clientes serão incluídos na RCL e qualquer cliente pode ser sorteado para ser inserido no indivíduo. Ao substituir o valor de α por 0, a Equação 36 transforma em $dist_para_cliente_i \leq distMin$, ou seja, apenas o cliente mais próximo será selecionado, sendo o escolhido para ser inserido no indivíduo.

Para uma maior diversidade da população inicial, usa-se o α com os valores 1, 0.5, 0.3 e 0 para a mesma proporção de quantidade de indivíduos.

4.1.2.5 Seleção

Para a definição dos indivíduos que serão submetidos ao cruzamento, um operador de seleção do tipo torneio foi implementado, com o objetivo de selecionar dois pais. Foi feito a

seleção por torneio duas vezes, uma para cada pai, no qual cada indivíduo foi comparado com outros três, com o objetivo de selecionar o melhor deles.

4.1.2.6 Cruzamento

Definido os dois pais pelo método de seleção, o cruzamento entre eles foi feito utilizando o operador PMX (*Partially Matched Crossover*) (GOLDBERG; LINGLE et al., 1985). O PMX opera sobre dois pais para gerar dois filhos, transmitindo um segmento de informação dos pais para os seus descendentes. Uma partição de clientes contidas no cromossomo de uma solução pai é selecionada de forma aleatória e mapeada em conjunto com clientes remanescentes de outra solução pai que não estão contidas neste segmento para gerar um descendente.

Como exemplo, na Figura 26 temos dois pais (A J C G B E D I F H) e (G F B J D H I A C E), respectivamente. O primeiro ponto de corte (i) está entre o terceiro e quarto elemento, e o segundo ponto de corte (j) está entre o oitavo e o nono elemento em cada solução pai. Neste caso, a sublista (G B E D I) delimitada por (i, j) no Cromossomo 1 é herdada pelo Cromossomo 4. Consecutivamente, a sublista (J D H I A) delimitada por (i, j) no Cromossomo 2 é herdada pelo Cromossomo 3.

As posições laterais aos segmentos nos filhos com elementos duplicados são alteradas conforme a tabela "Mapeamento" presente na figura. Por exemplo, no Cromossomo 4, o "G" na primeira posição está repetido. Então, de acordo com a tabela, o correspondente a ele é o "J". Como resultado, temos (B G C J D H I A F E) e (J F A G B E D I C H).

4.1.2.7 Mutação

O operador de mutação implementado consiste na troca de quatro pares de genes sorteados aleatoriamente, alterando a ordem no qual um cliente é visitado. Na Figura 27 é representado o operador de mutação para um par de genes.

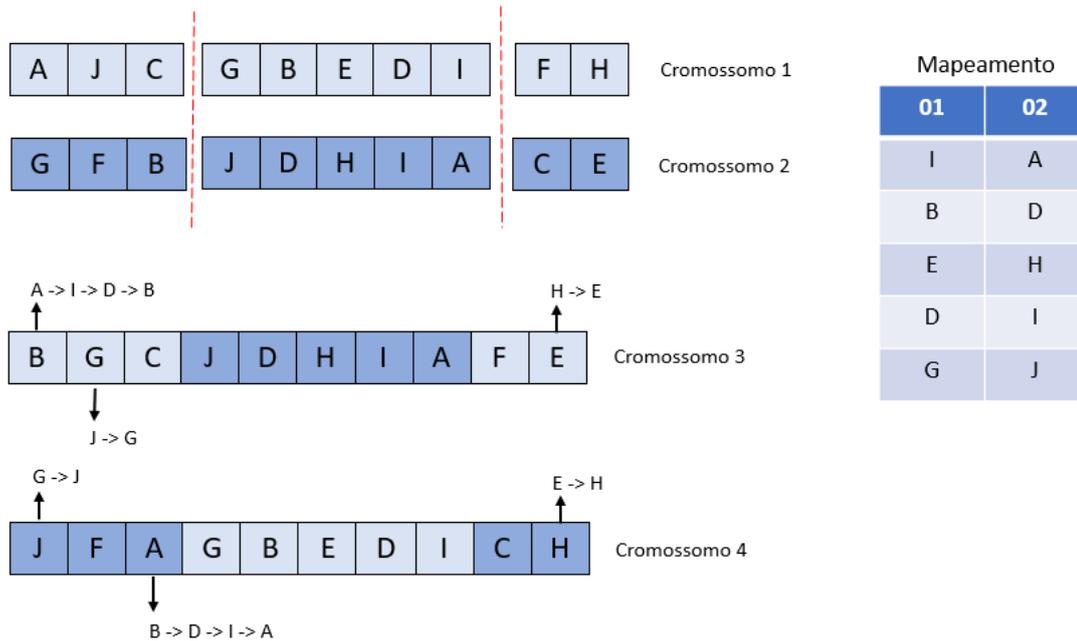
4.1.3 VNS, Busca Tabu e *Simulated Annealing*

As três meta-heurísticas, VNS, Busca Tabu e *Simulated Annealing*, partem de uma solução inicial do problema, e a partir dela, exploram o espaço de busca através do conceito de vizinhança. Assim sendo, os detalhes da implementação serão detalhados nas próximas seções.

4.1.3.1 Solução Inicial

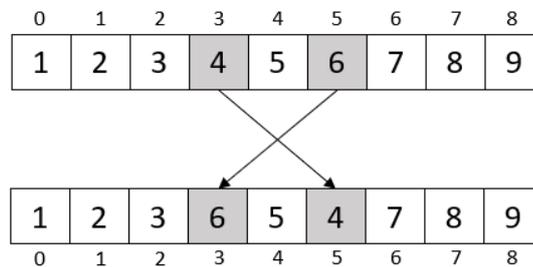
Do mesmo modo da população inicial do AG, a solução inicial é feita usando o CPLEX para resolver o modelo exato do CVRP limitando a capacidade do veículo. Usa-se as mesmas

Figura 26 – PMX - *Partially Matched Crossover*



Fonte: Elaborada pelo autor.

Figura 27 – Operador de Mutação



Fonte: Elaborada pelo autor.

porcentagens de volume, porém em execuções diferentes, já que essas meta-heurísticas precisam de apenas uma solução inicial.

4.1.3.2 Vizinhança

De modo a explorar o espaço de buscas, esses algoritmos utilizam o conceito de vizinhança. Uma vizinhança de uma solução são todas os movimentos que podem ser feitos na solução corrente, de tal maneira a gerar uma nova solução distinta. No caso dessa implementação, os movimentos são aplicados na sequência de clientes visitados de cada veículo (R_n) de uma solução, gerando assim, novas rotas, e por conseguinte, novos carregamentos. Os tipos de movimentos que são feitos na geração de novos vizinhos são os seguintes:

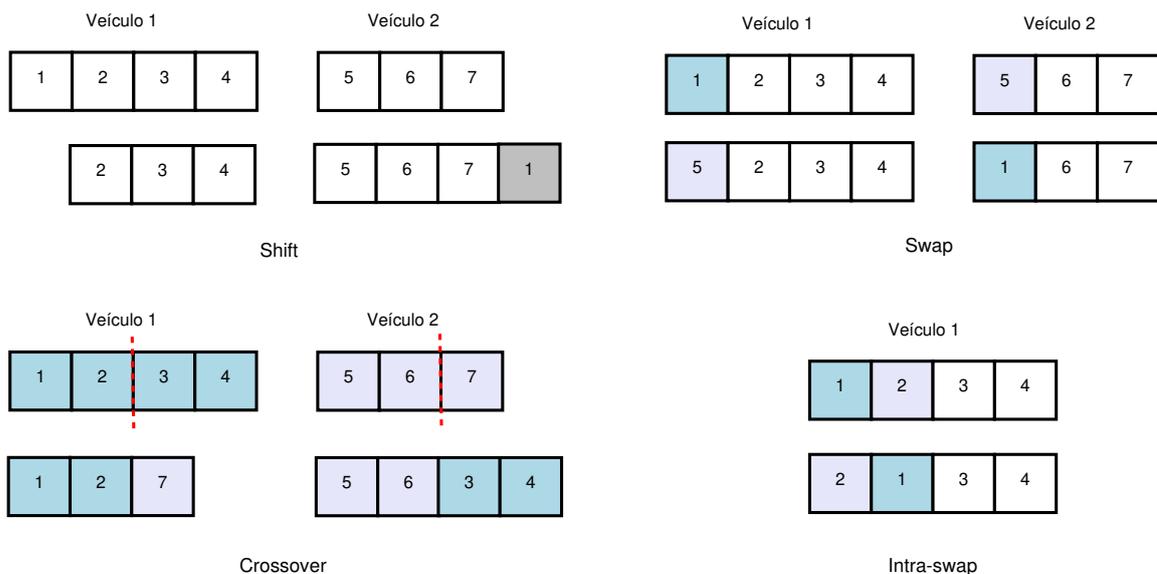
- *Shift*: para cada cliente $i \in R_j$, insere-se o cliente i na última posição em R_k e o

remove da rota R_j , com $R_j \neq R_k$.

- *Swap*: dados dois clientes $i \in R_k$ e $j \in R_l$, sendo $R_k \neq R_l$, troca-se i e j de posição.
- *Crossover*: sorteia-se dois pontos de corte, um em R_i e outro em R_j , sendo $R_i \neq R_j$. É construída uma nova R_i , que é formada pela primeira metade da R_i original e a segunda metade de R_j . O mesmo procedimento é feito para produzir uma nova R_j .
- *Intra-swap*: troca-se dois clientes $i, j \in R_k$ de posição.

Para a definição da vizinhança de uma solução, são realizados todos os movimentos *shift*, ou seja, todos os pares (cliente, posição) possíveis. O mesmo se aplica para todos os movimentos do tipo *swap*, *crossover* e *intra-swap*. Construída a vizinhança, usa-se a estratégia *Best Improvement Method*, no qual o melhor vizinho é retornado como solução a ser comparada com a corrente. A [Figura 28](#) ilustra os tipos de movimentos utilizados.

Figura 28 – Tipos de movimentos que são feitos na geração de novos vizinhos



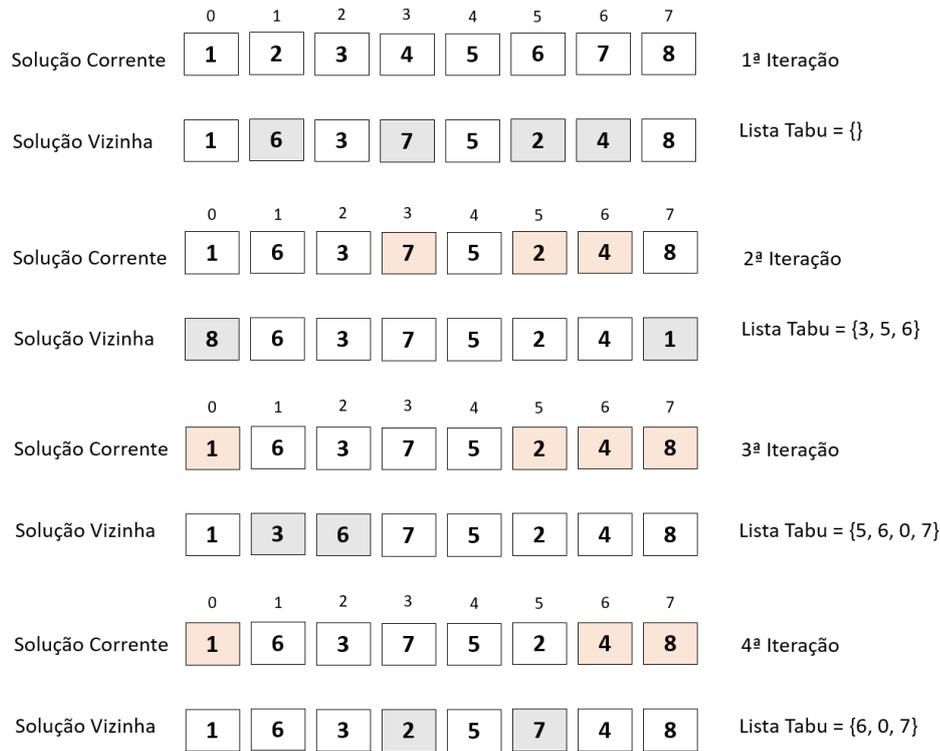
Fonte: Elaborada pelo autor.

4.1.3.3 Lista Tabu

A meta-heurística Busca Tabu utiliza o conceito de lista tabu. Ela é uma fila que armazena movimentos proibidos com o objetivo de evitar ciclismo. Na implementação deste trabalho, a lista tabu armazena as posições das rotas que diferem entre a solução corrente e a vizinha. A lista possui tamanho variável, alterando de acordo com a quantidade de posições diferentes a cada iteração. A cada passo do algoritmo, o movimento proibido mais antigo é retirado da fila, e os novos são adicionados. Esse procedimento está demonstrado na [Figura 29](#). Inicialmente a lista tabu está vazia. Na segunda iteração, as posições que

divergem, acinzentadas na figura, são incluídas na lista tabu. Nota-se que a primeira posição é incluída, mas retirada logo em seguida, já que a cada iteração remove-se a primeira posição da fila. As posições restritas estão destacadas em laranja. O mesmo processo se repete nas iterações restantes.

Figura 29 – Comportamento da lista tabu



Fonte: Elaborada pelo autor.

4.2 Modelagem Multiobjetivo

Nesta modelagem, o objetivo de minimizar a distância total percorrida permanece e acrescenta-se o objetivo de minimizar a quantidade de veículos utilizados para a entrega dos itens. Fez-se a implementação do algoritmo genético multiobjetivo NSGA-II.

4.2.1 Algoritmo NSGA-II

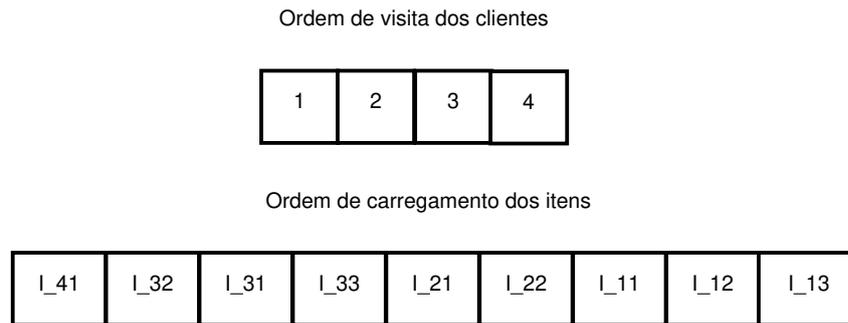
Para a implementação do algoritmo NSGA-II, foram definidos a população inicial, a codificação e decodificação do indivíduo e os tipos de operadores.

4.2.1.1 Codificação

O problema em questão foi codificado no NSGA-II em indivíduos como uma lista de itens, no qual cada posição da lista representa um item de um cliente específico. No intuito de complementar as informações do indivíduo, utiliza-se uma lista auxiliar para representar a

ordem em que cada cliente é visitado supondo que a rota é feita por um único veículo. A [Figura 30](#) mostra um exemplo de codificação de um cromossomo, no qual está representada a lista de clientes, em que o primeiro cliente visitado é o cliente 1, seguido pelo 2 e assim em diante, até concluir a rota com o cliente 4. O cliente 0 não é representado pois ele é o depósito. Na [Figura 30](#) também está representada a lista de itens, construída de forma inversa à ordem de visita dos clientes, garantindo a restrição de LIFO.

Figura 30 – Representação de um indivíduo no NSGA-II



Fonte: Elaborada pelo autor.

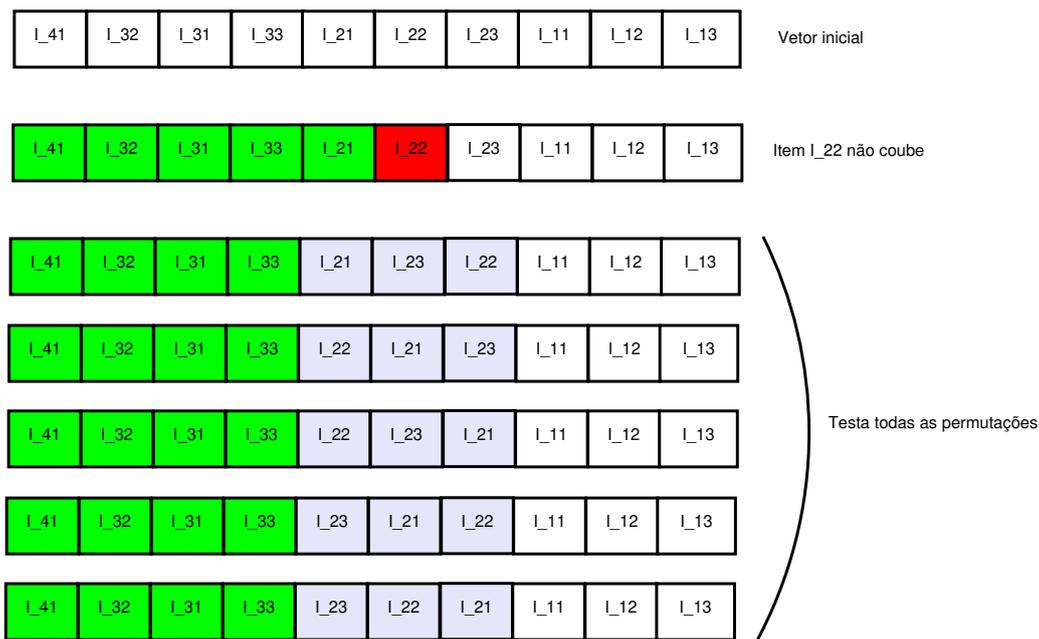
4.2.1.2 Decodificação e Avaliação do Indivíduo

Determinada a codificação do indivíduo, a sequência de itens é decodificada utilizando um algoritmo que percorre toda a lista. Dessa forma, o algoritmo verifica a cada posição se existe um carregamento para aquele item. Se existir e a demanda do veículo não for excedida, avança para o próximo item da lista. Caso ao final do processo, não for possível carregar todos os itens de um mesmo cliente em um mesmo veículo, remove-se todos os itens carregados daquele cliente e testa-se todas as ordens possíveis de carregamento, conforme ilustrado pela [Figura 31](#). Não obtendo êxito, busca-se um próximo cliente que não exceda a demanda restante do veículo e que seja possível o carregamento de seus itens, a fim de maximizar a utilização de espaço. Também aplica-se a técnica de testar todas as ordens possíveis com o objetivo de encontrar um carregamento válido. Terminada a estratégia, caso não consiga determinar um carregamento factível, um novo veículo é utilizado. A [Figura 32](#) apresenta a estratégia implementada. Com o carregamento feito, calcula-se a distância total percorrida de acordo com o número de veículos utilizados. O indivíduo é avaliado pela distância percorrida e a quantidade de veículos necessários. Quanto menor esses dois objetivos, melhor é o indivíduo.

4.2.1.3 População Inicial

Assim como na abordagem mono-objetivo, a população inicial foi construída utilizando a fase construtiva do GRASP. Utilizou-se a mesma [Equação 36](#) e os mesmos valores de α para se construir a RCL.

Figura 31 – Permuta a ordem dos itens do cliente

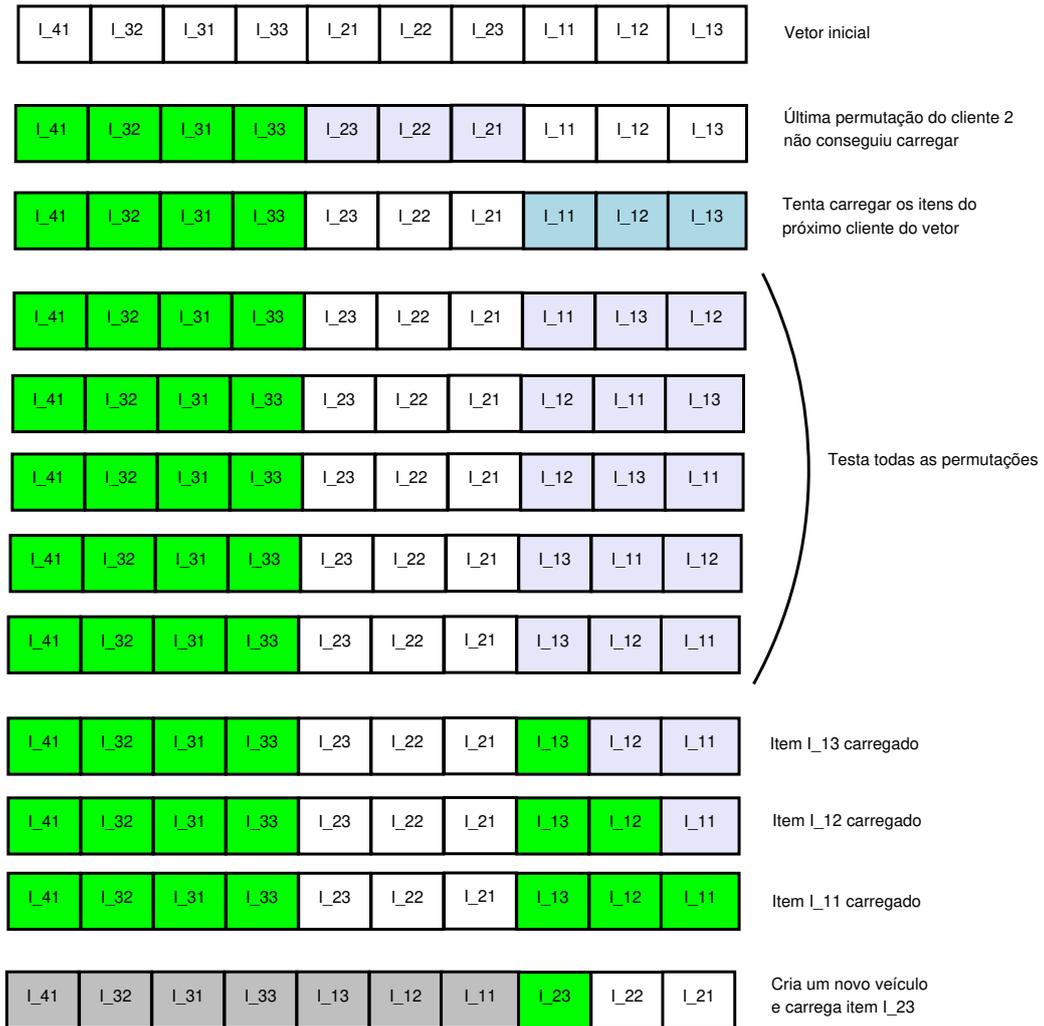


Fonte: Elaborada pelo autor.

4.2.1.4 Operações

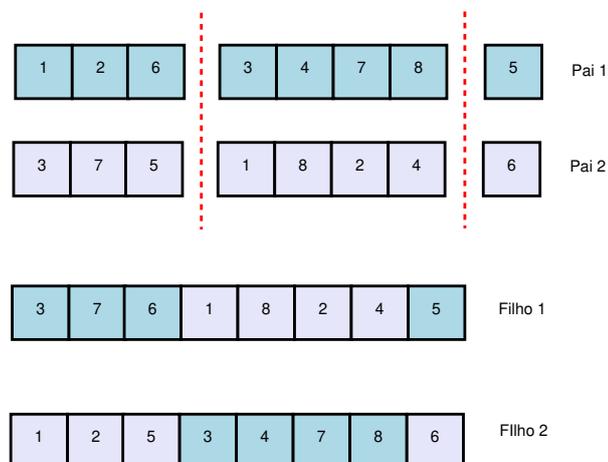
O operador de seleção implementado foi similar ao utilizado na abordagem mono-objetivo, no qual foi realizado a seleção por torneio duas vezes, uma para cada pai, em que cada indivíduo foi comparado com outros três, com o objetivo de selecionar o melhor deles. Em relação ao cruzamento, também utilizou-se o operador PMX. Porém, ele foi aplicado na lista de clientes e de itens, exemplificados pelas Figuras 33 e 34, respectivamente. No operador de mutação foi realizado troca de quatro pares de genes sorteados aleatoriamente, alterando a ordem no qual um cliente é visitado e a ordem em que um item é carregado (Figura 35).

Figura 32 – Testa-se o carregamento para os itens do próximo cliente da lista.



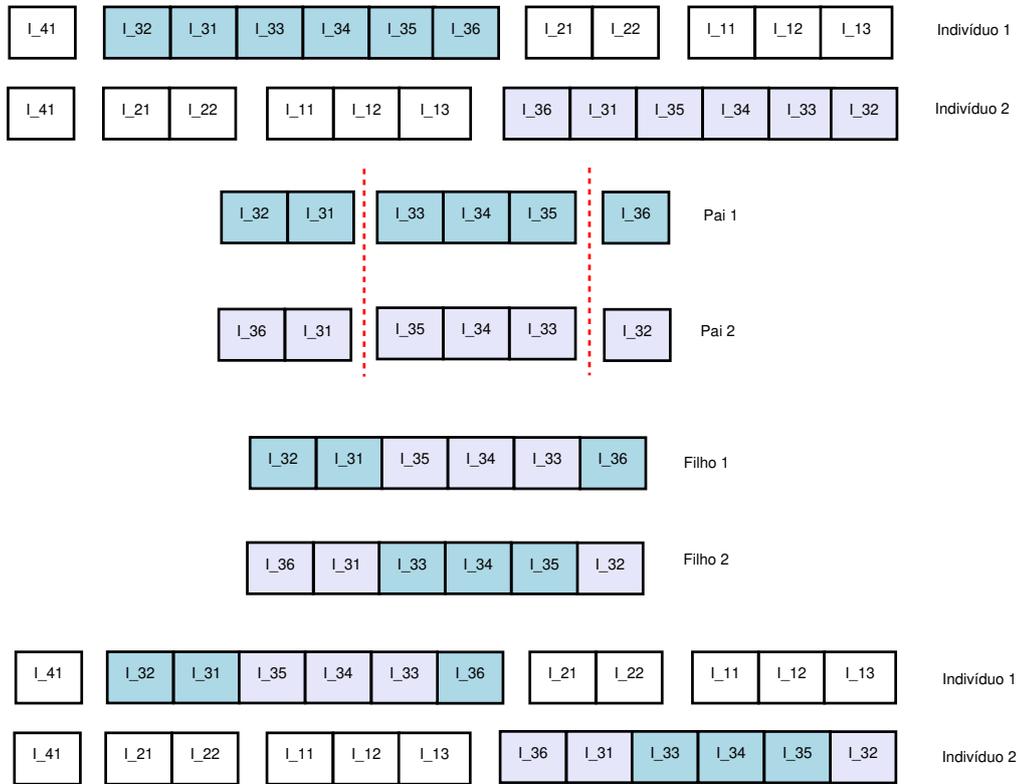
Fonte: Elaborada pelo autor.

Figura 33 – Cruzamento na lista de clientes



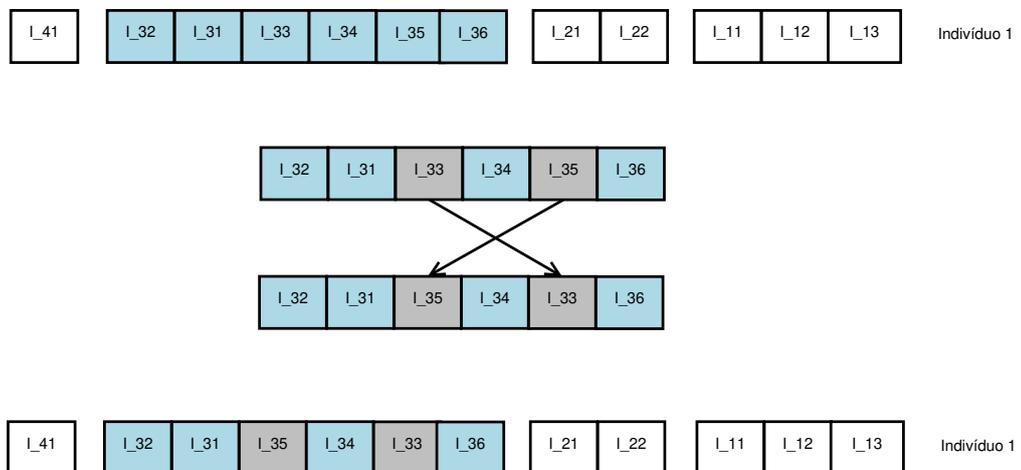
Fonte: Elaborada pelo autor.

Figura 34 – Cruzamento na lista de itens



Fonte: Elaborada pelo autor.

Figura 35 – Operador de mutação na ordem de carregamento dos itens de um cliente



Fonte: Elaborada pelo autor.

Capítulo 5

Experimentos Computacionais

Neste capítulo são apresentados os detalhes dos experimentos feitos bem como os resultados computacionais obtidos. Em seguida é mostrada a avaliação destes resultados de forma estatística.

5.1 Abordagem Mono-objetivo

Na modelagem mono-objetivo implementou-se os seguintes algoritmos: Algoritmo Genético, VNS, *Simulated Annealing* e Busca Tabu aplicado ao 3L-CVRP. Os parâmetros utilizados pelo Algoritmo Genético e *Simulated Annealing* podem ser vistos nas Tabelas 1 e 2, respectivamente. Para atender a restrição de estabilidade dos itens, usou-se $a = 0,75$, que é um parâmetro que determina o mínimo da área da base que deve ser suportada pelo topo de outro item ou pelo piso do veículo. Os algoritmos foram implementados na linguagem Java, utilizando um computador com processador Intel XEON E5506 @2.13GHz; 32 GB de memória RAM; Sistema Operacional Linux CentOS 6.6 64 bits.

Tabela 1 – Parâmetros do algoritmo genético

	Parâmetros
Tamanho da população	150
Taxa de cruzamento	0,85
Taxa de mutação	0,10
Elitismo	3

Para ser justo na comparação entre os algoritmos, foi usado o mesmo tempo de execução como critério de parada. Porém, esse valor variou entre as instâncias, já que elas possuem características particulares que dificultam a obtenção de uma solução factível. Os valores de tempo para cada instância podem ser vistos na [Tabela 3](#). Para solução do CVRP como solução inicial usando CPLEX foi definido o tempo de execução em 1800 segundos, no

Tabela 2 – Parâmetros do *Simulated Annealing*

	Parâmetros
Temperatura inicial	100
Número máximo de iterações por temperatura	100
Temperatura mínima	0,001
Taxa de decaimento da temperatura	0,95

qual ele encontrou soluções exatas e aproximadas dependendo do tamanho da instância testada.

Tabela 3 – Tempos de execução utilizados

Instâncias	Tempo (s)
01 a 06	3000
07	5000

As instâncias utilizadas foram propostas em [Gendreau et al. \(2006\)](#). Elas são instâncias modificadas definidas inicialmente em [Toth e Vigo \(2002\)](#). Utilizou-se as coordenadas, o valor das demandas dos clientes e a capacidade dos veículos. A partir desses dados, inseriu-se novas informações necessárias para o 3L-CVRP: altura ($H = 30$), largura ($W = 25$) e comprimento do veículo ($L = 60$), para cada cliente $i (i = 1, \dots, n)$, um número m_i de itens demandados foram gerados de forma randômica e uniforme variando entre 1 e 3, e para cada item criado, também gerou-se os valores de altura ($[0.2H, 0.6H]$), largura ($[0.2W, 0.6W]$) e comprimento ($[0.2L, 0.6L]$) de forma aleatória e uniforme. Feito isso, incrementou-se o número de veículos disponíveis de acordo com o valor original da instância em [Toth e Vigo \(2002\)](#), para que uma solução factível existisse. Para o cálculo da distância entre dois clientes, utilizou-se o conceito de distância euclidiana.

Os melhores resultados alcançados pelos algoritmos para o 3L-CVRP podem ser conferidos na [Tabela 4](#). Nela são apresentados os dados das instâncias, o *Best Known Solution* (BKS), os resultados divididos por algoritmos e pelos tipos de soluções iniciais. Já na [Tabela 5](#), são mostradas as médias das soluções obtidas para as quinze execuções realizadas para o AG e *Simulated Annealing*, já que os algoritmos VNS e Busca Tabu retornam sempre a mesma resposta para uma mesma solução inicial. O desvio-padrão dos resultados obtidos podem ser vistos na [Tabela 6](#). Para algumas instâncias testadas, houve algoritmos que não foram capazes de encontrar uma solução factível. Neste caso, foi colocado um traço nas tabelas no lugar dos resultados.

Tabela 4 – Melhores Resultados Computacionais 3L-CVRP

Instância	Clientes	Veículos	Itens	BKS	AG	Algoritmo VNS					Simulated Annealing					Busca Tabu				
						0,95	0,90	0,85	0,80	0,75	0,95	0,90	0,85	0,80	0,75	0,95	0,90	0,85	0,80	0,75
3l_cvrp01	15	4	32	300,69	340,72	360,46	423,69	363,90	-	-	356,69	353,42	336,03	332,30	330,86	375,85	381,92	360,46	387,03	-
3l_cvrp02	15	5	26	334,96	363,93	363,93	363,93	375,04	371,15	377,90	363,93	363,93	363,93	363,93	363,93	375,04	363,93	363,93	375,04	375,04
3l_cvrp03	20	4	37	374,81	482,32	-	541,07	-	-	-	461,72	469,00	453,66	462,04	456,30	-	-	-	509,31	-
3l_cvrp04	20	6	26	430,88	462,22	482,94	497,95	490,21	540,58	499,88	459,49	462,22	462,73	457,32	459,49	475,30	513,82	499,88	485,93	500,88
3l_cvrp05	21	6	45	436,48	623,95	-	-	-	-	-	-	-	594,00	588,27	571,34	-	-	-	-	-
3l_cvrp06	21	6	40	498,16	519,33	554,36	562,15	543,51	556,19	564,23	519,33	516,99	519,33	516,99	521,85	-	555,60	538,88	546,75	560,71
3l_cvrp07	22	6	46	767,46	1222,78	-	-	-	-	1192,14	-	-	982,32	1000,02	973,25	-	-	-	-	-

Tabela 5 – Média dos Resultados Computacionais 3L-CVRP

Instância	Clientes	Veículos	Itens	BKS	AG	Algoritmo VNS					Simulated Annealing					Busca Tabu				
						0,95	0,90	0,85	0,80	0,75	0,95	0,90	0,85	0,80	0,75	0,95	0,90	0,85	0,80	0,75
3l_cvrp01	15	4	32	300,69	355,82	360,46	423,69	363,90	-	-	383,98	388,35	388,21	380,49	375,87	375,85	381,92	360,46	387,03	-
3l_cvrp02	15	5	26	334,96	365,63	363,93	363,93	375,04	371,15	377,90	371,54	374,31	369,63	372,27	371,66	375,04	363,93	363,93	375,04	375,04
3l_cvrp03	20	4	37	374,81	527,26	-	541,07	-	-	-	506,01	516,93	506,06	515,42	501,96	-	-	-	509,31	-
3l_cvrp04	20	6	26	430,88	489,87	482,94	497,95	490,21	540,58	499,88	482,64	486,66	491,30	481,15	487,82	475,30	513,82	499,88	485,93	500,88
3l_cvrp05	21	6	45	436,48	743,47	-	-	-	-	-	-	-	751,11	750,26	713,02	-	-	-	-	-
3l_cvrp06	21	6	40	498,16	539,53	554,36	562,15	543,51	556,19	564,23	551,36	547,24	544,36	543,35	547,24	-	555,60	538,88	546,75	560,71
3l_cvrp07	22	6	46	767,46	1342,41	-	-	-	-	1192,14	-	-	996,34	1319,06	1236,79	-	-	-	-	-

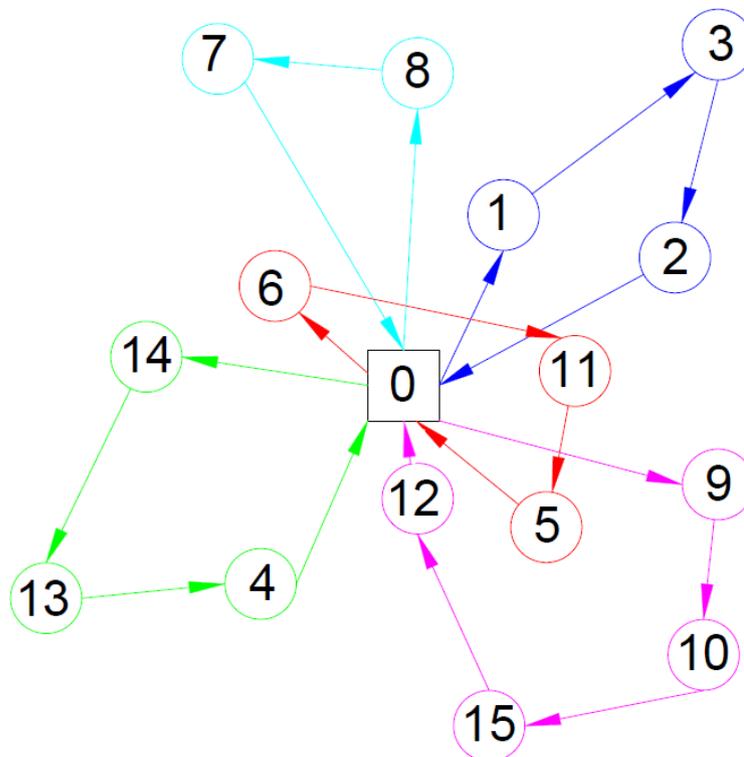
Tabela 6 – Desvio Padrão dos Resultados Computacionais 3L-CVRP

Instância	Clientes	Veículos	Itens	AG	Algoritmo VNS					Simulated Annealing					Busca Tabu					
					0,95	0,90	0,85	0,80	0,75	0,95	0,90	0,85	0,80	0,75	0,95	0,90	0,85	0,80	0,75	
3l_cvrp01	15	4	32	15,24	0,00	0,00	0,00	-	-	26,65	27,93	44,07	45,72	48,62	0,00	0,00	0,00	0,00	0,00	-
3l_cvrp02	15	5	26	3,89	0,00	0,00	0,00	0,00	0,00	5,72	5,01	5,91	8,79	6,97	0,00	0,00	0,00	0,00	0,00	0,00
3l_cvrp03	20	4	37	26,32	-	0,00	-	-	-	28,16	26,50	29,34	24,68	27,06	-	-	-	0,00	-	-
3l_cvrp04	20	6	26	14,66	0,00	0,00	0,00	0,00	0,00	18,39	18,50	15,87	16,82	16,55	0,00	0,00	0,00	0,00	0,00	0,00
3l_cvrp05	21	6	45	61,73	-	-	-	-	-	-	-	74,92	76,70	86,19	-	-	-	-	-	-
3l_cvrp06	21	6	40	17,68	0,00	0,00	0,00	0,00	0,00	25,82	23,73	21,16	20,70	21,43	-	0,00	0,00	0,00	0,00	0,00
3l_cvrp07	22	6	46	96,48	-	-	-	-	0,00	-	-	54,28	143,35	148,61	-	-	-	-	-	-

Como pode ser observado na [Tabela 4](#), o algoritmo *Simulated Annealing* superou os outros algoritmos em todas as instâncias testadas, exceto na instância "3I_cvrp02", no qual empatou com o Algoritmo Genético, VNS e Busca Tabu. Em relação aos melhores valores já encontrados na literatura, os algoritmos implementados não conseguiram superá-los em nenhuma instância testada. Uma observação importante presente nos resultados, é que a taxa máxima de ocupação usada nas soluções iniciais influenciou nos valores encontrados, já que um mesmo algoritmo alcançou resultados distintos para diferentes taxas, e até mesmo, possibilitou uma exploração no espaço de busca de modo a permitir encontrar uma solução factível. Já na análise dos valores de média dos resultados, mostrados na [Tabela 5](#), o AG superou os outros algoritmos em uma instância. Já o algoritmo VNS, juntamente com o Busca Tabu, obteve um resultado melhor em um problema teste. O Busca Tabu superou em duas instâncias e o *Simulated Annealing* em três. Como pode ser visto na [Tabela 6](#), o AG e o *Simulated Annealing* apresentaram uma menor variabilidade para a instância "3I_cvrp02".

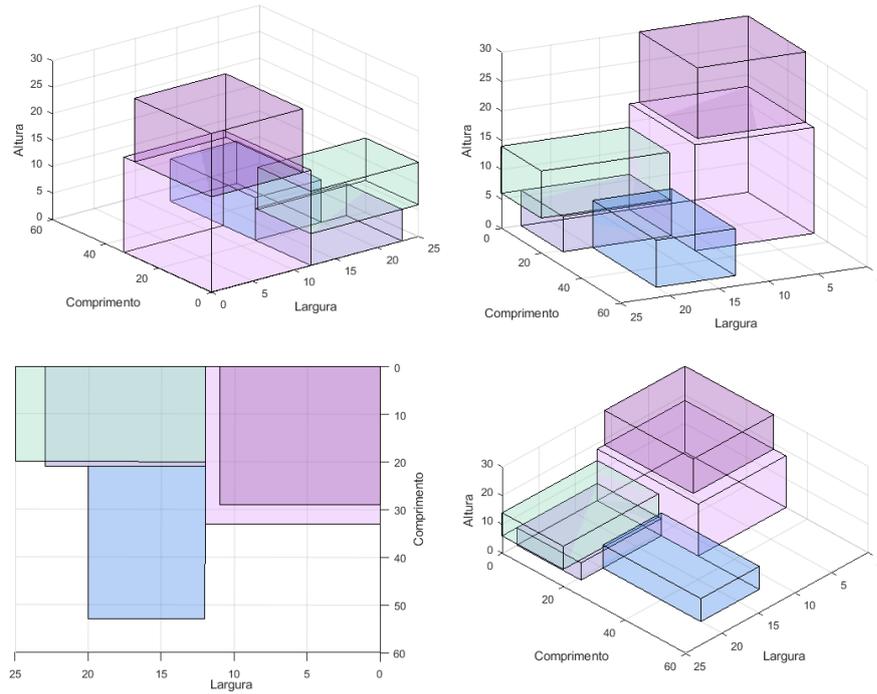
A [Figura 36](#) ilustra o roteamento obtido na solução da instância "3I_cvrp02" pelo algoritmo genético proposto. As [Figuras 37, 38, 39, 40 e 41](#) mostram o carregamento para as rotas correspondentes encontradas na solução.

Figura 36 – Solução roteamento da instância 3I_cvrp02



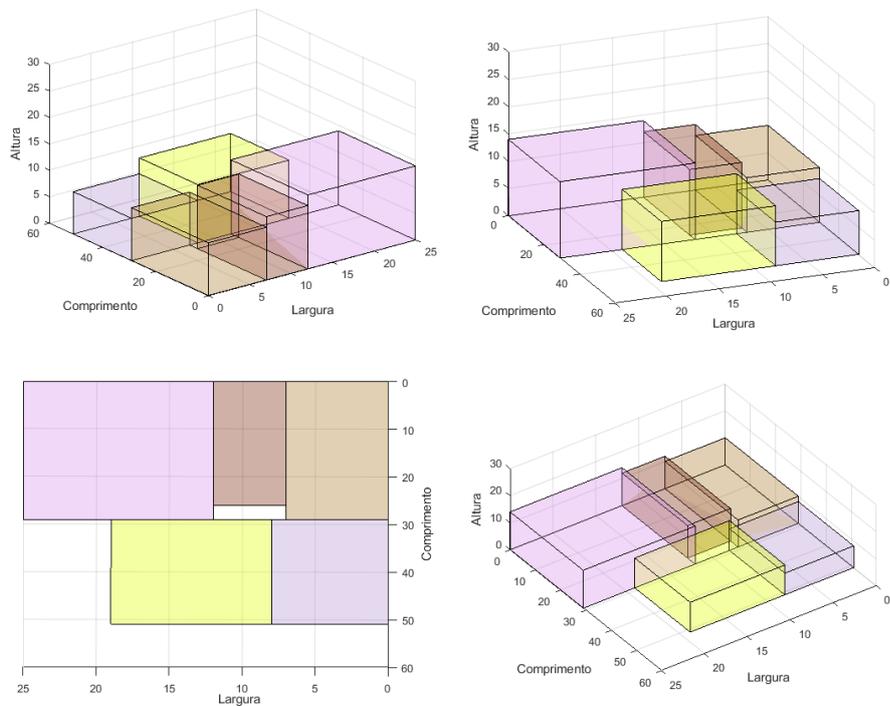
Fonte: Elaborada pelo autor.

Figura 37 – Solução carregamento da instância 3l_cvrp02 - Rota 6-11-5



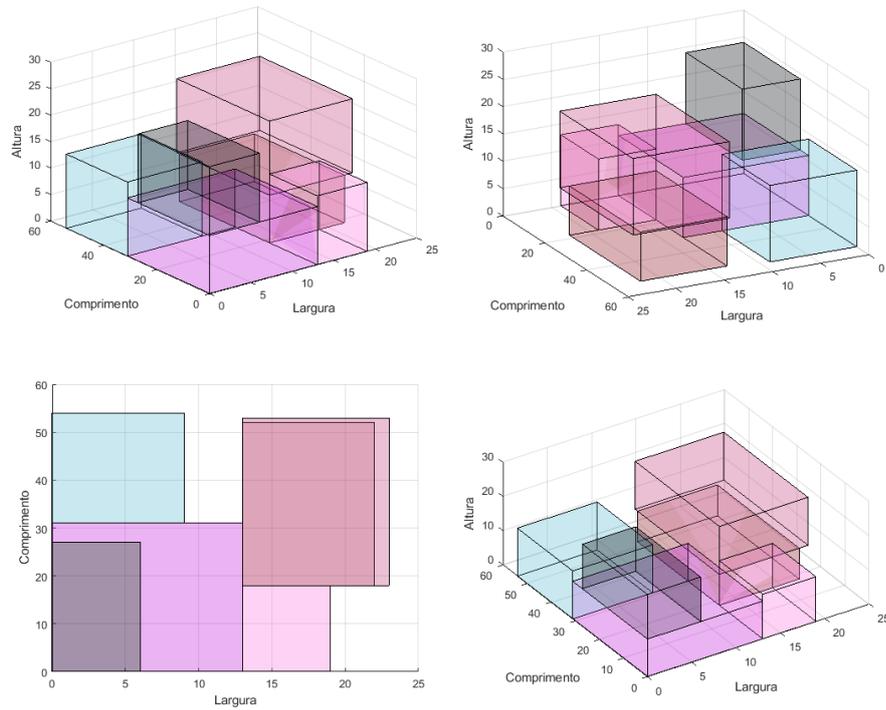
Fonte: Elaborada pelo autor.

Figura 38 – Solução carregamento da instância 3l_cvrp02 - Rota 1-3-2



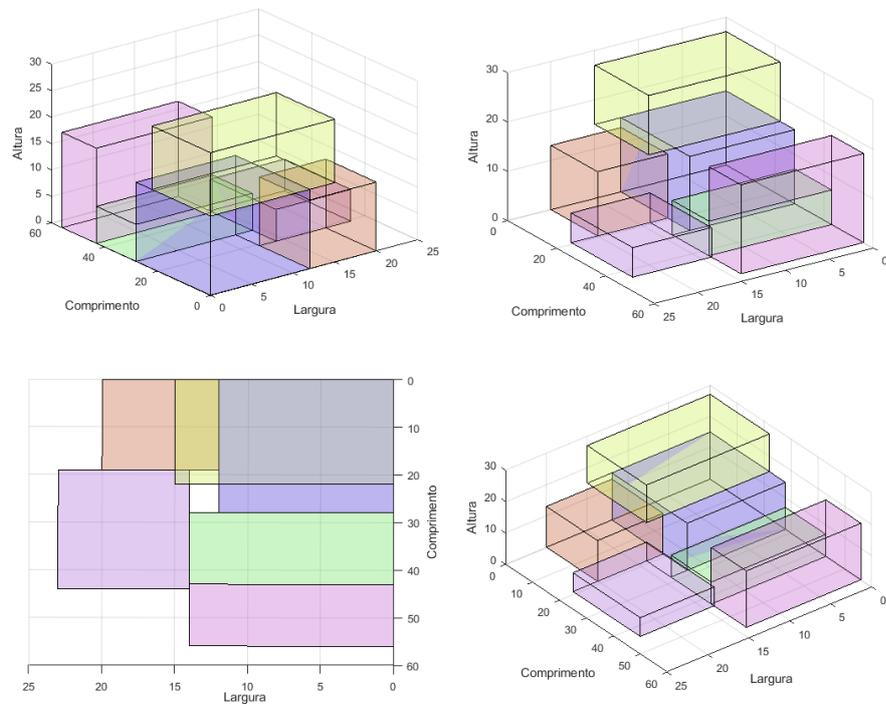
Fonte: Elaborada pelo autor.

Figura 39 – Solução carregamento da instância 3l_cvrp02 - Rota 14-13-4



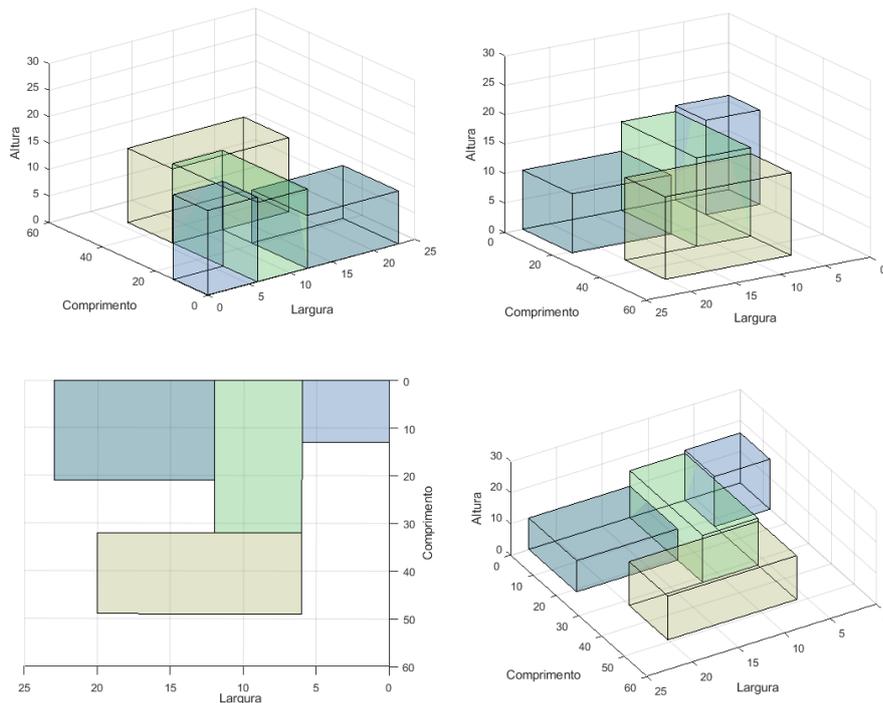
Fonte: Elaborada pelo autor.

Figura 40 – Solução carregamento da instância 3l_cvrp02 - Rota 9-10-15-12



Fonte: Elaborada pelo autor.

Figura 41 – Solução carregamento da instância 3l_cvrp02 - Rota 8-7



Fonte: Elaborada pelo autor.

5.1.0.1 Análise estatística dos resultados

Diante dos dados coletados utilizando cada um dos algoritmos para todas as instâncias testadas, foram feitos testes estatísticos para avaliar se existe diferença estatística significativa entre os quatro algoritmos implementados.

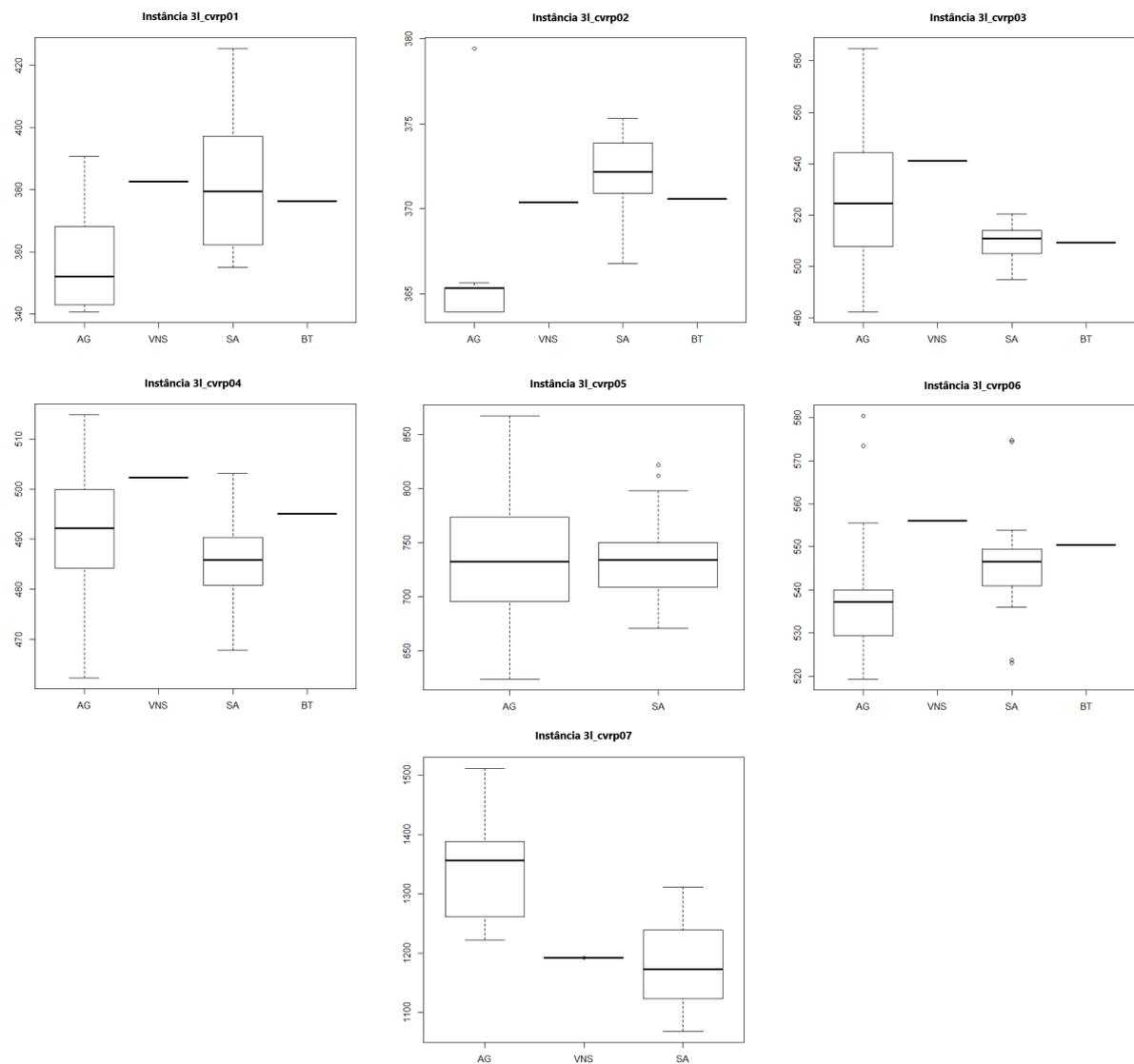
Cada instância foi executada quinze vezes para o AG e *Simulated Annealing*. Para realizar os testes estáticos em cada algoritmo, exceto no AG, foi usada a média das soluções de cada taxa máxima de ocupação utilizada na solução inicial. Definida as informações, foi feito inicialmente a avaliação de normalidade dos dados para decidir qual tipo de teste estatístico poderia ser utilizado (paramétrico ou não-paramétrico). Esta análise foi realizada utilizando os valores coletados e o teste de *Shapiro-Wilk* (SHAPIRO; WILK, 1965), complementado pela análise gráfica baseada em gráficos do tipo *boxplot*.

O *boxplot* ou diagrama de caixa é um gráfico estatístico que permite representar a distribuição de um conjunto de dados com base em parâmetros. Esse tipo de gráfico inclui a mediana, representada pela linha dentro do retângulo, o 1º e 3º quartil, os valores mínimos e máximos e eventuais *outliers*, que são valores muito discrepantes em relação à base de dados. Ele permite avaliar a simetria dos dados, sua dispersão e a existência ou não de *outliers*. Por meio desta ferramenta é possível avaliar de forma visual a normalidade dos dados, usando para isso, a simetria do gráfico. Para distribuições normais espera-se que a

mediana fique no centro do retângulo.

Os gráficos do tipo *boxplot* para cada instância estão apresentados na [Figura 42](#). Neles, o eixo X representa o algoritmo avaliado e o eixo Y representa os valores obtidos pela função objetivo em cada algoritmo implementado. Para algumas instâncias executadas no AG, os dados não apresentaram distribuição estatística normal, como pode ser verificado pelos *p*-valores menores que 0,05 retornados no teste de *Shapiro-Wilk*, destacados em negrito na [Tabela 7](#), e pela análise gráfica do *boxplot*.

Figura 42 – *Boxplots* dos resultados obtidos pelos algoritmos na abordagem mono-objetivo



Fonte: Elaborada pelo autor.

Diante da informação que existem dados que não apresentam distribuição normal, foi realizado o teste estatístico de *Kruskal-Wallis* (KRUSKAL; WALLIS, 1952). Este teste é semelhante ao teste da análise de variância (ANOVA) para detectar diferenças de trata-

Tabela 7 – Resultados do teste *Shapiro-Wilk*

Instância	<i>p</i> -valor			
	AG	VNS	<i>Simulated Annealing</i>	Busca Tabu
3l_cvrp01	0,03557	1	0,2255	1
3l_cvrp02	9,248E-7	1	0,1938	1
3l_cvrp03	0,9406	1	0,5223	1
3l_cvrp04	0,3761	1	0,961	1
3l_cvrp05	0,9443	-	0,3557	-
3l_cvrp06	0,01614	1	0,09764	1
3l_cvrp07	0,0928	1	0,6629	-

mentos em conjuntos de dados independentes. Porém, o teste de *Kruskal-Wallis* é um teste não-paramétrico, ou seja, não exige como pré-requisito a normalidade dos dados, ao contrário do teste ANOVA.

Foram realizados um teste para cada instância. Os resultados do teste de *Kruskal-Wallis* (*p*-valor) são apresentados na Tabela 8. O teste estatístico comprova diferenças significativas entre os algoritmos em todas as instâncias, exceto na "3l_cvrp05". Esse fato é evidenciado pela análise do *p*-valor menor que 0,05 (nível de significância).

Tabela 8 – Resultados do teste *Kruskal-Wallis*

Instância	<i>p</i> -valor
3l_cvrp01	3,466E-07
3l_cvrp02	3,277E-08
3l_cvrp03	4,324E-06
3l_cvrp04	5,482E-07
3l_cvrp05	0,7875
3l_cvrp06	3,88E-06
3l_cvrp07	4,02E-06

Dado esse resultado, é possível analisar os resultados em pares para os que apresentaram diferença, determinando assim quais dos algoritmos mostraram diferença estatística significativa. Essa análise foi realizada utilizando-se o teste de *Wilcoxon-Mann-Whitney* (WILCOXON, 1945).

Os resultados desse teste estão apresentados nas Tabelas 9, 10, 11, 12, 13 e 14, no qual é exibido o *p*-valor resultante da análise pareada dos algoritmos. Percebe-se pelos resultados observando de forma global, em todas as instâncias, para todos os algoritmos, que houve *p*-valor menor que 0,05, ou seja, apresentaram diferenças significativas. Por exemplo, observando a Tabela 9, o *p*-valor resultante da análise pareada do Algoritmo Genético e do *Simulated Annealing* foi de 0,000424, menor que 0,05. Ou seja, existe diferença significativa entre as médias dos resultados obtidos para a instância "3l_cvrp01".

Como o AG conseguiu um valor de média menor que o *Simulated Annealing*, podemos afirmar que o AG obteve um melhor resultado para essa instância, considerando que se trata de um problema de minimização.

Tabela 9 – Resultados do teste *Wilcoxon* para instância 3l_cvrp01

	AG	VNS	SA	BT
AG	-	7,04E-06	0,000424	7,04E-06
VNS	7,04E-06	-	0,5324	2,92E-08
SA	0,000424	0,5324	-	0,5324
BT	7,04E-06	2,92E-08	0,5324	-

Tabela 10 – Resultados do teste *Wilcoxon* para instância 3l_cvrp02

	AG	VNS	SA	BT
AG	-	1,41E-05	4,94E-05	1,41E-05
VNS	1,41E-05	-	0,002976	8,27E-08
SA	4,94E-05	0,002976	-	0,002976
BT	1,41E-05	8,27E-08	0,002976	-

Tabela 11 – Resultados do teste *Wilcoxon* para instância 3l_cvrp03

	AG	VNS	SA	BT
AG	-	0,101	0,0251	0,101
VNS	0,101	-	6,87E-07	8,27E-08
SA	0,0251	6,87E-07	-	0,3294
BT	0,101	8,27E-08	0,3294	-

Tabela 12 – Resultados do teste *Wilcoxon* para instância 3l_cvrp04

	AG	VNS	SA	BT
AG	-	0,000277	0,2454	0,3294
VNS	0,000277	-	1,71E-05	8,27E-08
SA	0,2454	1,71E-05	-	1,71E-05
BT	0,3294	8,27E-08	1,71E-05	-

Tabela 13 – Resultados do teste *Wilcoxon* para instância 3l_cvrp06

	AG	VNS	SA	BT
AG	-	0,000278	0,05639	0,002976
VNS	0,000278	-	0,000278	8,27E-08
SA	0,05639	0,000278	-	0,02115
BT	0,002976	8,27E-08	0,02115	-

Tabela 14 – Resultados do teste *Wilcoxon* para instância 3l_cvrp07

	AG	VNS	SA	BT
AG	-	6,474E-07	0,0001836	-
VNS	6,474E-07	-	0,7563	-
SA	0,0001836	0,7563	-	-
BT	-	-	-	-

5.2 Abordagem Multiobjetivo

Na abordagem multiobjetivo, no qual os objetivos são minimizar a distância total percorrida e a quantidade de veículos utilizados, aplicou-se o algoritmo NSGA-II aos problemas CVRP e 3L-CVRP. Iniciou-se a estratégia a partir do CVRP pois a resolução do *Capacitated Vehicle Routing Problem* é uma etapa intermediária para solução do 3L-CVRP, já que este é uma variante do CVRP com restrições de tridimensionalidade.

5.2.1 Resultados CVRP

Os parâmetros utilizados na implementação do algoritmo NSGA-II para o CVRP podem ser vistos na [Tabela 15](#). Para validação e avaliação da estratégia implementada realizou-se experimentos comparando os resultados com os obtidos por [Goodson, Ohlmann e Thomas \(2012\)](#). Para resolver o problema, os autores utilizaram *Simulated Annealing* combinada

com a estratégia *cyclic-order neighborhoods*, que utiliza uma ordem cíclica para percorrer o espaço de busca.

Tabela 15 – Parâmetros utilizados no NSGA-II para o CVRP

	Parâmetros
Número de gerações	300
Tamanho da população	150
Taxa de cruzamento	0,80
Taxa de mutação	0,03
Número de execuções	30

As instâncias utilizadas em ambos os trabalhos podem ser encontradas na biblioteca CVRPLIB (*Capacitated Vehicle Routing Problem Library*), que é uma biblioteca específica para o problema do caixeiro viajante capacitado. Elas apresentam as seguintes informações: número de clientes, incluindo o depósito, o número mínimo de veículos necessários, a capacidade máxima dos veículos, o número do cliente seguido de suas coordenadas x e y , onde o cliente de número 1 é o depósito, e em seguida, a demanda de cada cliente. Assim como foi feito no cálculo das soluções ótimas introduzido por (UCHOA et al., 2017), utilizou-se o conceito de distância euclidiana com valores arredondados para o cálculo da distância entre dois clientes, obtendo assim apenas distâncias inteiras.

Com o objetivo de comparar os resultados aos da literatura, apenas a solução obtida com melhor custo de cada instância foi considerada. Os resultados alcançados pelo algoritmo e Goodson, Ohlmann e Thomas (2012) juntamente com as soluções ótimas (UCHOA et al., 2017) podem ser conferidos na Tabela 16.

Nesses experimentos, utilizaram-se as seguintes configurações de hardware: processador Intel Core i7-6500U CPU @2.60GHz; 16 GB de memória RAM; Sistema Operacional Windows 10 Pro 64 bits.

Como pode ser observado na Tabela 16, o algoritmo NSGA-II conseguiu superar os resultados do Goodson, Ohlmann e Thomas (2012) em cinco instâncias, que estão destacadas em negrito. Na instância "P-n16-k8", o algoritmo alcançou o ótimo encontrado na literatura, sublinhado na tabela. Uma informação importante, é que para algumas instâncias, como por exemplo, a "A-n60-k9", o algoritmo chegou muito próximo do resultado encontrado pelo Goodson, Ohlmann e Thomas (2012), mesmo com um número menor de veículos. Isso é importante em um cenário realista, pois poderia ser viável percorrer uma maior distância usando um carro a menos, gerando uma economia com o desgaste natural das peças do carro. Por isso, em problemas multiobjetivo, é importante avaliar os dois objetivos, para se definir uma melhor solução.

Tabela 16 – Resultados Computacionais CVRP

Instância	Goodson, Ohlmann e Thomas (2012)			Solução ótima		Algoritmo NSGA-II		
	# Veic	Melhor Custo	Média Custo	# Veic	Custo	# Veic	Melhor Custo	Média Custo
A-n32-k5	5	853,60	853,60	5	784,00	5	896,00	922,23
A-n33-k5	5	704,20	705,91	5	661,00	5	713,00	713,40
A-n33-k6	6	793,90	793,95	6	742,00	6	808,00	820,70
A-n34-k5	6	826,87	827,26	5	778,00	5	835,00	842,90
A-n36-k5	5	858,71	859,48	5	799,00	5	928,00	933,43
A-n37-k5	5	708,34	709,67	5	669,00	5	821,00	835,13
A-n45-k7	7	1264,99	1288,70	7	1146,00	7	1264,00	1266,83
A-n60-k9	10	1529,82	1535,82	9	1354,00	9	1538,00	1548,00
P-n16-k8	8	512,82	512,82	8	450,00	8	450,00	454,13
P-n19-k2	3	224,06	224,06	2	212,00	3	253,00	256,56
P-n20-k2	2	233,05	233,05	2	216,00	2	226,00	240,46
P-n22-k8	9	681,06	681,06	8	603,00	9	619,00	629,96
P-n23-k8	9	619,53	619,53	8	529,00	9	555,00	579,08
P-n60-k15	16	1087,41	1098,41	15	968,00	16	1092,00	1099,70

5.2.2 Resultados 3L-CVRP

Implementou-se o algoritmo NSGA-II aplicado ao 3L-CVRP. Os parâmetros utilizados pelo algoritmo estão exemplificados na Tabela 17. O algoritmo foi implementado na linguagem Java, e os testes foram feitos em um computador com processador Intel Core i7-6500U CPU @2.60GHz; 16 GB de memória RAM; Sistema Operacional Windows 10 Pro 64 bits. As instâncias testadas e o valor de α referente a restrição de estabilidade foram os mesmos utilizados na abordagem mono-objetivo.

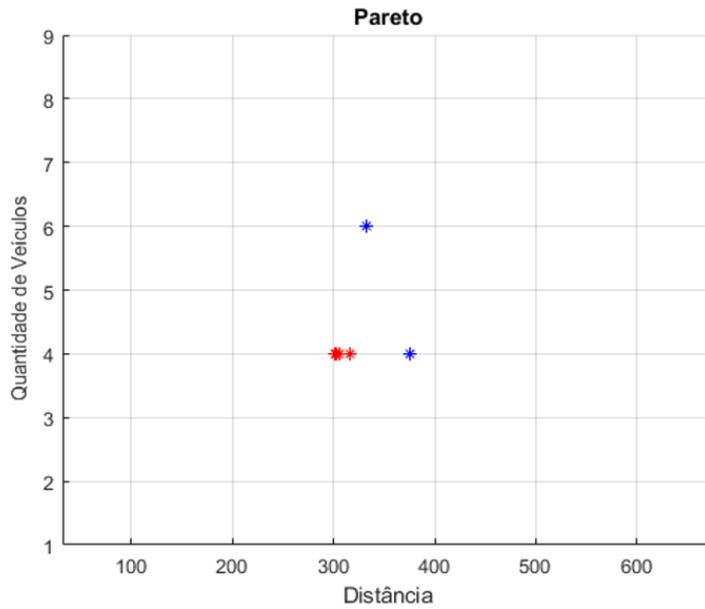
Tabela 17 – Parâmetros utilizados no NSGA-II para o 3L-CVRP

Parâmetros	
Número de gerações	3000
Tamanho da população	150
Taxa de cruzamento	0,85
Taxa de mutação	0,05

As fronteiras Pareto resultantes da execução do algoritmo juntamente com os resultados da literatura estão ilustrados nas Figuras 43, 44, 45 e 46. Os pontos em azul representam os resultados obtidos pelo algoritmo e os pontos em vermelho representam as soluções obtidas por Tarantilis, Zachariadis e Kiranoudis (2009), Fuellerer et al. (2010), Ruan et al. (2013), Tao e Wang (2015) e Escobar-Falcón et al. (2016).

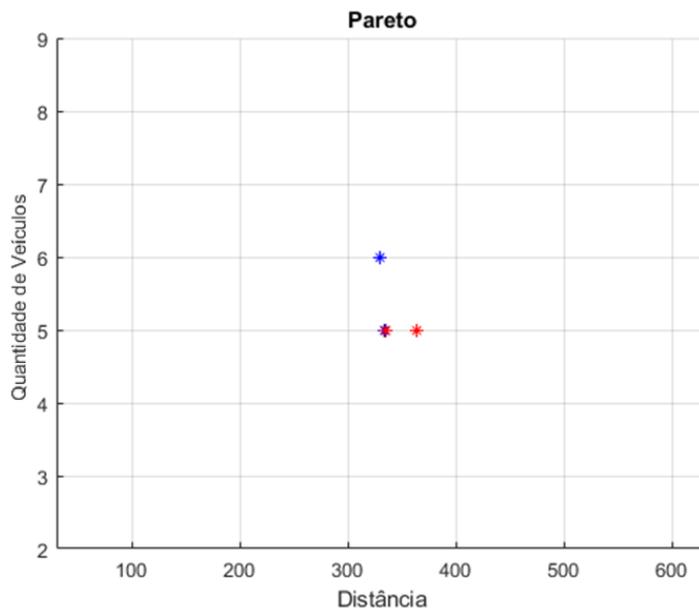
Como pode ser visto na Figura 44, as soluções obtidas pelo algoritmo na instância "3l_cvrp02" foram capazes de superar um resultado da literatura e empatou com os restantes para a quantidade de veículos informado na instância. Porém, utilizando um veículo a mais, o algoritmo implementado conseguiu uma menor distância total percorrida. No restante das instâncias testadas, o objetivo de distância das soluções encontradas não foram menores em comparação com a literatura para um número maior ou igual de veículos

Figura 43 – Fronteira Pareto-ótima da instância 3I_cvrp01. Os pontos em azul representam os resultados obtidos pelo algoritmo e os pontos em vermelho representam as soluções da literatura



Fonte: Elaborada pelo autor.

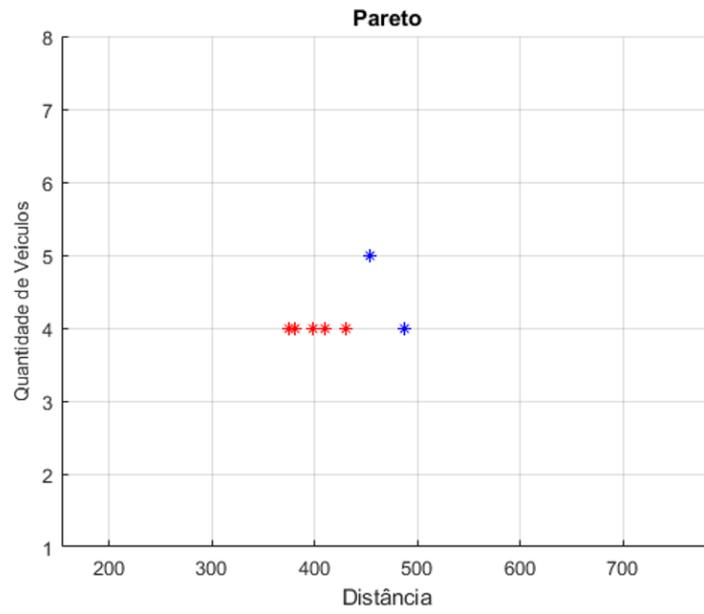
Figura 44 – Fronteira Pareto-ótima da instância 3I_cvrp02



Fonte: Elaborada pelo autor.

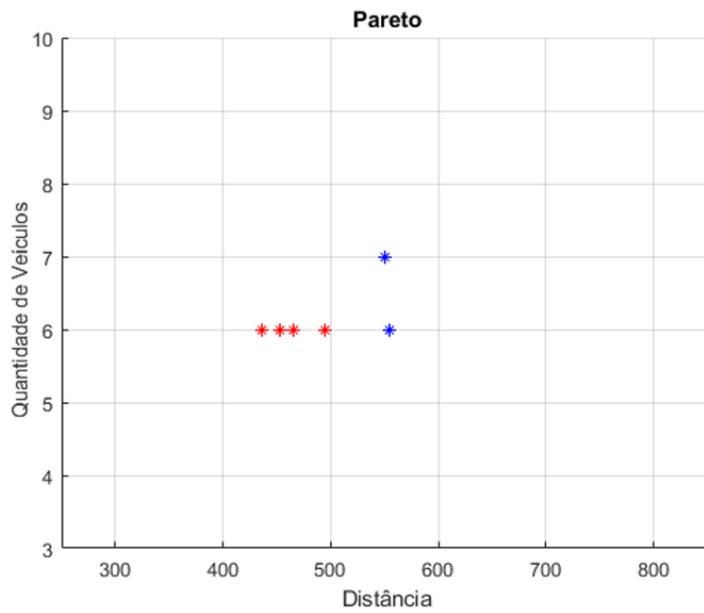
e a quantidade de veículos não foi menor ou igual para uma maior distância percorrida. Como as instâncias utilizadas foram criadas com um número pré-determinado de veículos, a abordagem multiobjetivo pode não ter sido favorecida, já que um dos objetivos foi a quantidade de veículos.

Figura 45 – Fronteira Pareto-ótima da instância 3I_cvrp03



Fonte: Elaborada pelo autor.

Figura 46 – Fronteira Pareto-ótima da instância 3I_cvrp05



Fonte: Elaborada pelo autor.

Capítulo 6

Considerações Finais

Este trabalho apresenta uma proposta de otimização para o problema do roteamento de veículos capacitados com restrições tridimensionais de carregamento em duas modelagens distintas, mono e multiobjetivo. No mono-objetivo, foi proposta a implementação de meta-heurísticas, mais especificamente os algoritmos: Algoritmo Genético, VNS, *Simulated Annealing* e Busca Tabu. Na abordagem multiobjetivo, foi feita a implementação do algoritmo NSGA-II aplicado ao CVRP e 3L-CVRP. Implementou-se o algoritmo para o CVRP pois a solução deste problema é uma etapa intermediária para se resolver o 3L-CVRP, já que este é uma versão do CVRP com restrições de tridimensionalidade.

Na abordagem mono-objetivo, os resultados obtidos pelos algoritmos apresentados foram avaliados em comparação com os melhores valores já encontrados na literatura, sendo que algoritmos não conseguiram superá-los em nenhuma instância testada. Fazendo uma análise dos valores obtidos entre os algoritmos, o *Simulated Annealing* conseguiu os melhores resultados, superando o AG, VNS e Busca Tabu em todas as instâncias testadas, exceto em uma, no qual empatou com o restante dos algoritmos implementados.

Em relação a abordagem multiobjetivo, avaliou-se o algoritmo utilizando instâncias da literatura encontradas na biblioteca CVRPLIB e comparadas com as melhores soluções já encontradas e os resultados obtidos por [Goodson, Ohlmann e Thomas \(2012\)](#). Os testes computacionais realizados sobre as instâncias produziram resultados satisfatórios para o CVRP, com soluções próximas às ótimas. Em comparação com a outra abordagem, a solução implementada neste trabalho apresentou resultados superiores para algumas instâncias testadas e em outras demonstrou resultados muito próximos, mesmo com um número menor de veículos.

Diante dos bons resultados para o CVRP, aplicou-se a mesma estratégia juntamente com o algoritmo de carregamento para instâncias do 3L-CVRP. Os resultados obtidos foram comparados aos da literatura, no qual o algoritmo implementado superou um resultado da

literatura e empatou com os outros para uma instância testada considerando o objetivo de distância percorrida para a quantidade de veículos disponibilizados. No entanto, adicionando um veículo a mais na solução, o algoritmo proposto conseguiu uma menor distância total. Essa situação é interessante em um cenário real, pois caso tenha-se um maior número de veículos disponíveis, pode-se percorrer uma menor distância, economizando gastos com combustível, desgaste de peças, entre outras despesas que norteiam a logística de entrega de produtos.

Diante de melhores resultados para o CVRP em comparação com o 3L-CVRP, conclui-se que o método proposto aplicado ao roteamento de veículos se mostrou eficaz, porém ao ser combinado com o carregamento de veículos, ele não conseguiu explorar o espaço de busca de maneira eficiente de modo a produzir resultados melhores do que os encontrados na literatura para a maioria das instâncias testadas.

6.1 Trabalhos Futuros

Como trabalhos futuros, sugere-se a utilização da estratégia de carregamento a partir da análise da área não ocupada restante no veículo ao inserir um novo item, de modo a permitir uma melhora nos resultados obtidos neste trabalho. Ainda considerando o carregamento dos itens, uma outra sugestão de trabalho futuro é avaliar uma melhor maneira de decidir a ordem que os itens de um mesmo cliente devem ser carregados. Uma boa ordem de carregamento dos itens permite uma melhor alocação de espaço dentro do veículo, permitindo que um veículo atenda um maior número de clientes.

Em relação ao roteamento de veículos, uma possibilidade de trabalho futuro é utilizar outras formas de soluções iniciais para as meta-heurísticas. Melhores soluções iniciais tendem a permitir uma melhor exploração do espaço de buscas, possibilitando encontrar melhores soluções finais.

No que se refere a abordagem multiobjetivo, uma alternativa de trabalho futuro interessante é combinar o 3L-CVRP com tempo, no qual um dos objetivos seria minimizar o tempo gasto pelo veículo nas entregas dos itens. Ainda em relação a abordagem multiobjetivo, uma outra possibilidade de trabalho futuro é utilizar movimentações, em que um dos objetivos seria minimizar a quantidade de movimentações de caixas a serem feitas durante a entrega de um item, desconsiderando a restrição que nenhum outro item pertencente à um outro cliente deve ser movimentado durante a entrega.

6.2 Publicações

Nesta seção são listadas as publicações originadas durante o desenvolvimento desta pesquisa.

- **Um Algoritmo Genético aplicado ao Problema de Roteamento de Veículos Capacitados**

Lucas Daniel Padia Rocha; Flávio V. Cruzeiro Martins

LI Simpósio Brasileiro de Pesquisa Operacional (SBPO), 2019, Limeira-SP

Disponível em: <<https://bit.ly/2T06CBA>>

- **Otimização para o Problema de Roteamento de Veículos Capacitados com restrições tridimensionais de carregamento**

Lucas Daniel Padia Rocha; Flávio Vinícius Cruzeiro Martins

XXXIX Congresso Nacional de Matemática Aplicada e Computacional (CNMAC), 2019, Uberlândia-MG

Referências

- ARROYO, J. E. C. **Heurísticas e metaheurísticas para otimização combinatória multi-objetivo**. Tese (Doutorado) — Universidade Estadual de Campinas, 2002. Citado na página [19](#).
- AZUMA, R. M. **Otimização multiobjetivo em problema de estoque e roteamento gerenciados pelo fornecedor**. Dissertação (Mestrado) — Universidade Estadual de Campinas, 2011. Citado na página [19](#).
- BARBAROSOGLU, G.; OZGUR, D. A tabu search algorithm for the vehicle routing problem. **Computers & Operations Research**, Elsevier, v. 26, n. 3, p. 255–270, 1999. Citado na página [31](#).
- BISCHOFF, E. E.; RATCLIFF, M. Issues in the development of approaches to container loading. **Omega**, Elsevier, v. 23, n. 4, p. 377–390, 1995. Citado na página [6](#).
- BORTFELDT, A. A hybrid algorithm for the capacitated vehicle routing problem with three-dimensional loading constraints. **Computers & Operations Research**, Elsevier, v. 39, n. 9, p. 2248–2257, 2012. Citado 2 vezes nas páginas [3](#) e [5](#).
- CAMPOS, V.; MOTA, E. Heuristic procedures for the capacitated vehicle routing problem. **Computational Optimization and Applications**, Springer, v. 16, n. 3, p. 265–277, 2000. Citado na página [2](#).
- CHU, P. C.; BEASLEY, J. E. A genetic algorithm for the multidimensional knapsack problem. **Journal of heuristics**, Springer, v. 4, n. 1, p. 63–86, 1998. Citado na página [6](#).
- CLARKE, G.; WRIGHT, J. W. Scheduling of vehicles from a central depot to a number of delivery points. **Operations research**, *Inform*s, v. 12, n. 4, p. 568–581, 1964. Citado 2 vezes nas páginas [3](#) e [4](#).
- COELLO, C. C. Evolutionary multi-objective optimization: a historical view of the field. **IEEE computational intelligence magazine**, IEEE, v. 1, n. 1, p. 28–36, 2006. Citado na página [19](#).
- CRAINIC, T. G.; PERBOLI, G.; TADEI, R. Extreme point-based heuristics for three-dimensional bin packing. **Inform**s **Journal on computing**, INFORMS, v. 20, n. 3, p. 368–384, 2008. Citado 2 vezes nas páginas [18](#) e [19](#).
- CUNHA, C. B. da et al. Heurísticas para o problema de bin-packing no contexto da distribuição física de produtos. **XL Simpósio Brasileiro de Pesquisa Operacional**, 2008. Citado na página [28](#).
- DANTZIG, G. B.; RAMSER, J. H. The truck dispatching problem. **Management science**, *Inform*s, v. 6, n. 1, p. 80–91, 1959. Citado na página [9](#).
- DARWIN, C.; WALLACE, A. On the tendency of species to form varieties; and on the perpetuation of varieties and species by natural means of selection. **Journal of the proceedings of the Linnean Society of London. Zoology**, Wiley Online Library, v. 3, n. 9, p. 45–62, 1858. Citado na página [22](#).

Deb, K. et al. A fast and elitist multiobjective genetic algorithm: Nsga-ii. **IEEE Transactions on Evolutionary Computation**, v. 6, n. 2, p. 182–197, April 2002. ISSN 1089-778X. Citado 3 vezes nas páginas 25, 26 e 27.

DÍAZ, A. et al. **Optimización heurística y redes neuronales**. [S.l.]: Paraninfo, 1996. Citado na página 20.

ESCOBAR-FALCÓN, L. M. et al. A matheuristic algorithm for the three-dimensional loading capacitated vehicle routing problem (3l-cvrp). **Revista Facultad de Ingeniería Universidad de Antioquia**, Universidad de Antioquia, n. 78, p. 09–20, 2016. Citado 3 vezes nas páginas 4, 5 e 58.

FEO, T. A.; RESENDE, M. G. Greedy randomized adaptive search procedures. **Journal of global optimization**, Springer, v. 6, n. 2, p. 109–133, 1995. Citado na página 38.

FUELLERER, G. et al. Metaheuristics for vehicle routing problems with three-dimensional loading constraints. **European Journal of Operational Research**, Elsevier, v. 201, n. 3, p. 751–759, 2010. Citado 5 vezes nas páginas 3, 5, 11, 17 e 58.

GENDREAU, M. et al. A tabu search algorithm for a routing and container loading problem. **Transportation Science**, INFORMS, v. 40, n. 3, p. 342–350, 2006. Citado 5 vezes nas páginas 2, 3, 4, 11 e 48.

GLOVER, F. Future paths for integer programming and links to artificial intelligence. **Computers & operations research**, Elsevier, v. 13, n. 5, p. 533–549, 1986. Citado na página 31.

GOLDBERG, D. E.; LINGLE, R. et al. Alleles, loci, and the traveling salesman problem. In: LAWRENCE ERLBAUM, HILLSDALE, NJ. **Proceedings of an international conference on genetic algorithms and their applications**. [S.l.], 1985. v. 154, p. 154–159. Citado na página 39.

GOODSON, J. C.; OHLMANN, J. W.; THOMAS, B. W. Cyclic-order neighborhoods with application to the vehicle routing problem with stochastic demand. **European Journal of Operational Research**, Elsevier, v. 217, n. 2, p. 312–323, 2012. Citado 4 vezes nas páginas 56, 57, 58 e 61.

GUIMARÃES, T. A. **Uma nova abordagem heurística para a resolução do problema do roteamento de veículos capacitados com restrições tridimensionais de carregamento**. Dissertação (Mestrado) — Universidade Federal do Paraná, 2012. Citado 4 vezes nas páginas 1, 2, 9 e 12.

HANSEN, P. The steepest ascent mildest descent heuristic for combinatorial programming. In: **Congress on numerical methods in combinatorial optimization, Capri, Italy**. [S.l.: s.n.], 1986. p. 70–145. Citado na página 31.

HOLLAND, J. H. Genetic algorithms. **Scientific american**, JSTOR, v. 267, n. 1, p. 66–73, 1992. Citado 5 vezes nas páginas 20, 22, 23, 24 e 25.

IORI, M.; SALAZAR-GONZÁLEZ, J.-J.; VIGO, D. An exact approach for the vehicle routing problem with two-dimensional loading constraints. **Transportation science**, INFORMS, v. 41, n. 2, p. 253–264, 2007. Citado na página 11.

KATAOKA, V. d. S. et al. **Avaliação de algoritmos evolucionários multiobjetivo para o problema de alocação de bancos de capacitores na presença de harmônicos**. Dissertação (Mestrado em Engenharia Elétrica) — Universidade Federal do Pará, 2017. Citado na página 27.

KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by simulated annealing. **science**, American Association for the Advancement of Science, v. 220, n. 4598, p. 671–680, 1983. Citado 2 vezes nas páginas 20 e 30.

KRUSE, R. et al. **Computational intelligence: a methodological introduction**. [S.l.]: Springer, 2016. Citado na página 23.

KRUSKAL, W. H.; WALLIS, W. A. Use of ranks in one-criterion variance analysis. **Journal of the American Statistical Association**, Taylor Francis, v. 47, n. 260, p. 583–621, 1952. Disponível em: <<https://www.tandfonline.com/doi/abs/10.1080/01621459.1952.10483441>>. Citado na página 54.

LAWLER, E. L. The traveling salesman problem: a guided tour of combinatorial optimization. **Wiley-Interscience Series in Discrete Mathematics**, 1985. Citado na página 9.

LIMA, M. **Custos Logísticos no Brasil**. 2014. Disponível em: <<https://www.ilos.com.br/web/custos-logisticos-no-brasil/>>. Citado na página 1.

LIN, S.-W. et al. Applying hybrid meta-heuristics for capacitated vehicle routing problem. **Expert Systems with Applications**, Elsevier, v. 36, n. 2, p. 1505–1512, 2009. Citado na página 10.

MARTELLO, S.; PISINGER, D.; VIGO, D. The three-dimensional bin packing problem. **Operations research**, INFORMS, v. 48, n. 2, p. 256–267, 2000. Citado 2 vezes nas páginas 7 e 18.

MIAO, L. et al. A hybrid genetic algorithm for the vehicle routing problem with three-dimensional loading constraints. **RAIRO-Operations Research**, EDP Sciences, v. 46, n. 1, p. 63–82, 2012. Citado 2 vezes nas páginas 4 e 5.

MLADENOVIC, N.; HANSEN, P. Variable neighborhood search. **Computers & operations research**, Elsevier, v. 24, n. 11, p. 1097–1100, 1997. Citado 2 vezes nas páginas 20 e 28.

PEARL, J. **Heuristics: intelligent search strategies for computer problem solving**. [S.l.]: Addison-Wesley Pub. Co, 1984. (The Addison-Wesley series in artificial intelligence). ISBN 9780201055948,0-201-05594-5. Citado na página 20.

PORTAL, G. et al. Uma busca tabu aplicada ao problema de roteamento com restrições de empacotamento tridimensionais. **Anais do XLI Simpósio Brasileiro de Pesquisa Operacional**, 2009. Citado 2 vezes nas páginas 3 e 5.

RUAN, Q. et al. A hybrid approach for the vehicle routing problem with three-dimensional loading constraints. **Computers & Operations Research**, Elsevier, v. 40, n. 6, p. 1579–1589, 2013. Citado 4 vezes nas páginas 4, 5, 14 e 58.

SHAPIRO, S. S.; WILK, M. B. An analysis of variance test for normality (complete samples)†. **Biometrika**, v. 52, n. 3-4, p. 591–611, 12 1965. ISSN 0006-3444. Disponível em: <<https://doi.org/10.1093/biomet/52.3-4.591>>. Citado na página 53.

SILVA, J. L. d. C.; SOMA, N. Y. Um algoritmo polinomial para o problema de empacotamento de contêineres com estabilidade estática da carga. **Pesquisa Operacional**, SciELO Brasil, v. 23, n. 1, p. 79–98, 2003. Citado na página 7.

TAO, Y.; WANG, F. An effective tabu search approach with improved loading algorithms for the 3l-cvrp. **Computers & Operations Research**, Elsevier, v. 55, p. 127–140, 2015. Citado 3 vezes nas páginas 4, 5 e 58.

TARANTILIS, C. D.; ZACHARIADIS, E. E.; KIRANOUDIS, C. T. A hybrid metaheuristic algorithm for the integrated vehicle routing and three-dimensional container-loading problem. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 10, n. 2, p. 255–271, 2009. Citado 4 vezes nas páginas 3, 4, 5 e 58.

TEIXEIRA, U. R. **Algoritmos baseados na metaheurística VNS para resolução do problema de programação de horários em escolas**. Dissertação (Mestrado) — Centro Federal de Educação Tecnológica de Minas Gerais, 2018. Citado na página 20.

TICONA, W. G. C.; DELBÉM, A. C. B. Algoritmos evolutivos para otimização multi-objetivo. **Relatório técnico. Notas Didáticas do ICMC-USP**, v. 76, 2008. Citado na página 28.

TOTH, P.; VIGO, D. **The vehicle routing problem**. [S.l.]: SIAM, 2002. Citado na página 48.

UCHOA, E. et al. New benchmark instances for the capacitated vehicle routing problem. **European Journal of Operational Research**, Elsevier, v. 257, n. 3, p. 845–858, 2017. Citado na página 57.

WILCOXON, F. Individual comparisons by ranking methods. **Biometrics Bulletin**, [International Biometric Society, Wiley], v. 1, n. 6, p. 80–83, 1945. ISSN 00994987. Disponível em: <<http://www.jstor.org/stable/3001968>>. Citado na página 55.

ZHU, W. et al. A two-stage tabu search algorithm with enhanced packing heuristics for the 3l-cvrp and m3l-cvrp. **Computers & Operations Research**, Elsevier, v. 39, n. 9, p. 2178–2195, 2012. Citado 2 vezes nas páginas 3 e 5.

ZUBEN, F. V. **Computação evolutiva: Uma abordagem pragmática**. UNICAMP, 2000. Citado na página 22.