



UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Tecnologia

Marcos Ramon Ramalho

**Problema de Roteamento de Veículos voltado para a
entrega de cartas na cidade de Artur Nogueira**

Limeira
2019

Marcos Ramon Ramalho

**Problema de Roteamento de Veículos voltado para a entrega de cartas
na cidade de Artur Nogueira**

Dissertação apresentada à Faculdade de Tecnologia da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Bacharel em Sistemas de Informação, na área de Sistemas de Informação e Comunicação.

Orientador: Prof. Dr. Luis Augusto Angelotti Meira

Este exemplar corresponde à versão final da Dissertação defendida por Marcos Ramon Ramalho e orientada pelo Prof. Dr. Luis Augusto Angelotti Meira.

Limeira
2019

FOLHA DE APROVAÇÃO

Abaixo se apresentam os membros da comissão julgadora de defesa de dissertação para o Título de Bacharel em Sistemas de Informação na área de concentração de Sistemas de Informação e Comunicação, a que submeteu o aluno Marcos Ramon Ramalho, em 29 de novembro de 2019 na Faculdade de Tecnologia – FT/UNICAMP, em Limeira/SP.

Prof. Dr. Luis A. A. Meira
Presidente da Comissão Julgadora

Prof. Dr. Guilherme Palermo Coelho
FT/UNICAMP

Prof. Dr. Plínio Roberto Souza Vilela
FT/UNICAMP

Ata da defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria de Pós Graduação da FT.

Agradecimentos

Primeiramente gostaria de agradecer aos meus pais, Marcos Rogério Ramalho e Katia Regina da Costa Ramalho que sempre me apoiaram e estavam presentes durante a minha formação. Sem vocês eu não conseguiria ter chegado até aqui.

A minha esposa Stefanie Moraes que sempre me motivou a superar todos os desafios e momentos difíceis vivenciados no período acadêmico. Você é a razão da minha felicidade.

Ao Prof. Dr. Luis Augusto Angelotti Meira por toda a paciência em me orientar e ter me passado conhecimento suficiente para a realização desse trabalho.

Enfim, sou muito grato por todos os amigos e outros familiares que sempre estiveram ao meu lado e indiretamente também ajudaram bastante para que esse sonho fosse realizado.

Resumo

O Problema de Roteamento de Veículos (do inglês *Vehicle Routing Problem*, VRP) é uma generalização do Problema do Caixeiro Viajante (do inglês *Traveling Salesman Problem*, TSP). É um problema que têm no seu contexto várias rotas e veículos. O VRP pode ser encontrado em vários problemas cotidianos como entrega de cartas, coleta de lixo, transporte público (ônibus/mêtro), entre outros. Neste trabalho utilizamos dois algoritmos, o algoritmo de troca de vértice (TrocaV) e o de troca de arestas, também chamado 2-Opt. Ambos algoritmos foram retirados da literatura. Utilizamos uma variante do VRP que limita a capacidade da rota. Como modelo utilizamos o benchmark conhecido como PostVRP, que é um benchmark que possui instâncias com diversos pontos de entrega. Possui instâncias com no mínimo 20 pontos de entrega e, no máximo, 30.000 pontos de entrega. O algoritmo de Troca de vértice (TrocaV) foi executado em instâncias com até 1.000 pontos de entrega e o 2-Opt em instâncias com até 10.000. Comparando as soluções criadas pelos dois algoritmos, o 2-Opt obteve resultados superiores ao do TrocaV. Em alguns casos o custo do 2-Opt foi duas vezes menor que o TrocaV.

Palavras Chaves: 2-Opt, VRP, PostVRP.

Abstract

The Vehicle Routing Problem (VRP) is a generalization of the Traveling Salesman Problem (TSP). It is a problem that has in its context several routes and vehicles. VRP can be found in many everyday problems such as letter delivery, garbage collection, public transportation (bus / subway), among others. The Vertex Swap algorithm (TrocaV) ran on instances with up to 1,000 delivery points and 2-Opt on instances with up to 10,000. Comparing the solutions created by the two algorithms, 2-Opt obtained better results than TrocaV. In some cases the cost of 2-Opt was twice as low as TrocaV. In this work we developed two algorithms, the vertex exchange algorithm (TrocaV) and the edge exchange algorithm, also called 2-Opt. We use a variant of VRP that limits the capacity of the route. As a model we use the benchmark known as PostVRP, which is a benchmark that has instances with multiple delivery points. It has instances with a minimum of 20 delivery points and a maximum of 30,000 delivery points.

Keywords: 2-Opt, VRP, PostVRP.

Lista de Figuras

3.1	Modelo do mapa utilizado	26
3.2	Pontos de entrega em parte do mapa	26
3.3	Movimento de troca - 2-Opt	28
3.4	Estrutura de pontos - Movimento de troca. Fonte: Hasegawa, Ikeguchi e Aihara (1997)	28
4.1	Gráfico Tempo de Processamento (segundos) X Número de Instâncias para o algoritmo TrocaV.	31
4.2	Rota da Solução RealWorldPostToy 20-1 Algoritmo 2-opt.	34
4.3	Rota da Solução RealWorldPostToy 20-1 Algoritmo TrocaV.	34
4.4	Rota da Solução RealWorldPostToy 1000-0 Algoritmo 2-Opt.	35
4.5	Rota da Solução RealWorldPostToy 1000-0 Algoritmo TrocaV.	35

Lista de Tabelas

1.1	Taxonomia Utilizada	14
1.2	Comparação das Heurísticas do VRP. (Fonte (PRINS, 2004)).	16
4.1	Relação entre o algoritmo 2-opt e o de troca de vértices (TrocaV)	31
4.2	Resultados Obtidos - Algoritmo 2-opt sem clusterização, ou seja, única rota ou TSP.	32
4.3	Resultados Obtidos - Algoritmo 2-opt para PostVRP	32
4.4	Resultados Obtidos - Algoritmo TrocaV para o PostVRP	32
4.5	Resultados Obtidos - Algoritmo TrocaV sem clusterização, ou seja, rota única ou TSP	32
4.6	Relação entre o algoritmo 2-opt e o AG	33
4.7	Relação entre o algoritmo TrocaV e o AG	33

Sumário

1	Introdução	10
1.1	Escopo do Problema	11
1.1.1	Características do Cenário	11
1.1.2	Componentes do Gerenciamento de Decisão	12
1.1.3	Depósito	12
1.1.4	Tipos de Problemas	13
1.1.5	Divisão de Mercadorias	13
1.1.6	Veículos	13
1.1.7	Restrições de Tempo	14
1.2	Trabalhos Relacionados	14
2	Algoritmos de Otimização para VRP	18
2.1	Algoritmos Exatos	18
2.1.1	Algoritmos de Branch-and-Bound	18
2.2	Heurísticas Clássicas	19
2.2.1	<i>The Savings Algorithm</i>	19
2.2.2	<i>Cluster-First, Route-Second Heuristics</i>	19
2.2.3	Heurísticas Incrementais	19
2.3	Metaheurísticas	20
2.3.1	Heurísticas Construtivas	20
2.3.2	Mecanismos Populacionais	20
2.3.3	Mecanismos de Aprendizado	21
2.3.4	Busca Local	21
2.3.5	<i>Tabu Search (TS)</i>	23
2.3.6	<i>Hill Climbing (HC)</i>	24
2.3.7	Considerações	24
3	Metodologia Experimental	25
3.1	Algoritmo de troca de vértice entre dois pontos da rota (TrocaV)	27
3.2	Algoritmo 2-Opt	28
4	Resultados	30
4.1	Ambiente e Cenário dos Experimentos	30
4.2	Resultado dos Experimentos	30
5	Conclusões	36
	Referências bibliográficas	37

Capítulo 1

Introdução

O Problema de Roteamento de Veículos (do inglês *Vehicle Routing Problem*, VRP) pode ser observado no cotidiano das pessoas, em situações como fretamento de ônibus, coleta de lixo, entregas e atendimentos aos clientes, distribuição de bebidas, entregas de cartas ou de jornais, dentre tantas outras opções. Diminuir custos para realizar entregas, minimizar o comprimento das rotas, entregar produtos em um tempo menor são exemplos de problemas de roteamento presentes na maioria das empresas. Segundo Appa, Pitsoulis e Williams (2006) o VRP é encontrado em vários ambientes industriais e sua otimização se reflete diretamente em economia de recursos.

Para realizar entregas pela menor rota e dentro de um tempo estipulado é preciso considerar um conjunto potencialmente grande de rotas candidatas. Devido a restrições de tempo da empresa, nem sempre é possível buscar a solução ótima. Neste contexto aplicado, soluções heurísticas podem ser suficientemente boas. Os autores Cordeau, Gendreau et al. (2005) relataram que nos últimos 30 anos, o número de publicações sobre o problema de roteamento de veículos tem aumentado consideravelmente. Também com o desenvolvimento tecnológico dos últimos anos os problemas foram ganhando mais variáveis com restrições e objetivos mais complexos que fazem se assemelhar com problemas de roteamento realísticos.

O roteamento de veículos é um problema que consiste na análise da divisão das entregas entre veículos e a ordenação das entregas de cada veículo. Um dos grandes desafios é a redução do custo do transporte. Segundo Appa, Pitsoulis e Williams (2006) estima-se que no Canadá e no Reino Unido, as despesas do transporte representam 15% do total das despesas nacionais. O autor diz que o problema de roteamento de veículos pode ser representado como um grafo não direcionado $G = (V, E)$, onde $V = \{v_0, v_1, \dots, v_n\}$ é um conjunto de

vértices e $E = (v_i, v_j) : v_i, v_j \in V, i < j$ é uma aresta. O vértice v_0 é um depósito no qual irão sair os veículos, enquanto os vértices restantes representam clientes.

O número de veículos é conhecido antecipadamente. O custo, em nosso caso, é a distância que é calculada em uma matriz $C = (c_{ij})$ definida em E . O custo de viagens são proporcionais às distâncias. O problema desse trabalho se difere de alguns casos de VRP em que o cliente pode ter uma demanda não negativa Q_i e um tempo de serviço não negativo S_i .

O objetivo deste trabalho é encontrar rotas de custo mínimo, sendo que o depósito será o ponto de partida e chegada e cada cliente deve ser visitado apenas uma vez.

Tratamos o problema em duas etapas. Na primeira etapa, fizemos a otimização de uma única rota, reduzindo o problema a um TSP.

Na segunda etapa, quebramos a rota de maneira a respeitar as restrições de comprimento. Trabalhamos assim com rotas cheias para todos os veículos. Quando uma rota enche, fechamos a rota e damos início a uma nova rota. Desta maneira, as rotas ficam cheias, a menos a última rota, que tem comprimento, em geral, bem menor.

1.1 Escopo do Problema

O Trabalho de Lahyani, Khemakhem e Semet (2015) constrói a taxonomia do problema de acordo com conceitos centrais em roteamento que estão freqüentemente presentes nas aplicações. Todas os componentes da taxonomia estão relacionadas às características que alteram a natureza do problema, focando nas propriedades relevantes. O autor adota uma granularidade média para a taxonomia, pois, poucas variáveis podem não conseguir descrever o problema como ele realmente é, assim, como muitas variáveis podem deixar o problema muito extenso. Nos parágrafos abaixo é apresentada a taxonomia do VRP.

1.1.1 Características do Cenário

Segundo Lahyani, Khemakhem e Semet (2015) os dados podem ser subdivididos em quatro classes: determinística, estocástica, estática e dinâmica. O problema de roteamento determinístico assume que os parâmetros do problema são conhecidos com certeza, enquanto os dados estocásticos possuem distribuições de probabilidade que estão associados a eles. No VRP estocástico as rotas podem não sair como planejado, pois, diferentemente da classe

determinística, os dados do cenário são aleatórios e não são conhecidos. Os três parâmetros mais comuns nessa forma são: demanda de clientes, tempos de serviços e entregas.

Para o VRP dinâmico, o plano de programação estabelecido no início do período de planejamento pode ser ajustado. Esse tipo de problema permite receber informações adicionais e alterar alguns parâmetros de problemas. Por exemplo, novas solicitações de clientes podem ocorrer durante o período de planejamento e devem ser consideradas enquanto as rotas dos veículos estão sendo executadas.

Em problemas estáticos, por sua vez, as variáveis são conhecidas e não se alteram. Um exemplo é o VRP com janelas temporais em que variáveis como tempo, demanda e localização são conhecidas e estáticas.

1.1.2 Componentes do Gerenciamento de Decisão

O Trabalho de Lahyani, Khemakhem e Semet (2015) cita que a gestão da cadeia de abastecimento é um conjunto de abordagens utilizadas para integrar de forma eficiente os clientes, fabricantes, armazéns e lojas. Os autores afirmam que esta técnica garante a produção dos produtos e a distribuição na quantidade certa, para a localização correta e no momento certo para minimizar os custos. Segundo os autores, integrar decisões de diferentes funções como compras, controle de estoque, terceirização, localização de depósitos, planejamento de produção e gerenciamento de distribuição são uma prática cada vez mais seguida por muitas empresas. O que está relacionado ao VRP são roteamento de inventário, localização, de produção e agendamento de veículo ou motorista. Em *Inventory Routing Problems* (Problemas de roteamento para estoques, IRP), o fornecedor define, além dos planos de roteiro, as quantidades a serem entregues usando os níveis de estoque.

1.1.3 Depósito

O depósito nos algoritmos clássicos de VRP é o lugar único onde partem e chegam os veículos. Isso torna o problema muito restritivo, pois, se comparado com a realidade muitas variáveis não são levadas em conta. Veículos podem possuir ponto de partida e ponto de chegada, não necessariamente no mesmo lugar. Empresas multinacionais, por exemplo, podem possuir vários depósitos.

1.1.4 Tipos de Problemas

Segundo Lahyani, Khemakhem e Semet (2015) existem quatro classes de problemas de roteamento de veículos. Problemas em que as mercadorias são entregues ou recolhidos, problemas em que materiais são carregados e descarregados, problemas onde as mercadorias são carregadas a bordo, quando a parte de entrega ou da rota está concluída e problemas de disjunctive viagem como Uber, por exemplo.

Os autores afirmam que o problema clássico do VRP lida com entrega ou recolhimento de mercadorias. Se assemelha com as entregas de compras online. As mercadorias são carregadas no depósito e, em seguida, descarregadas nos locais do cliente ou são retiradas e descarregadas no depósito.

1.1.5 Divisão de Mercadorias

Segundo Lahyani, Khemakhem e Semet (2015) no VRP clássico, cada cliente é servido por apenas um veículo. A possibilidade de várias visitas ao mesmo cliente caracteriza o VRP com entregas divididas (SDVRP) introduzidas por (DROR; TRUDEAU, 1989).

1.1.6 Veículos

Segundo Lahyani, Khemakhem e Semet (2015) veículos com diferentes características atendem melhor às necessidades dos clientes relacionadas às restrições físicas, às preocupações, a equipamentos logísticos específicos ou à demanda de variações. As primeiras características dos veículos e dos processos de carga / descarga podem ser alinhadas no planejamento. Por último, o regulamento dos motoristas também pode afetar significativamente as rotas do veículo. Um exemplo, é o caso do aplicativo Uber quando é cobrado um valor maior dependendo da quantidade de carros disponíveis em rota.

Para Lahyani, Khemakhem e Semet (2015) uma propriedade que também deve ser levada em conta é a capacidade de carga. A capacidade de um veículo pode ser medida de diversas formas, como quantidade de paletes, peso ou volume. Um veículo pode possuir diversos compartimentos de diferentes cargas. Um exemplo desse tipo de problema é o transporte de gado, pois, deve combinar o bem-estar do animal em conjunto com o problema do transporte, por isso, os animais devem ser transportados em compartimentos diferentes.

1.1.7 Restrições de Tempo

Determina o tempo em que várias entregas devem ocorrer. O tempo de carregamento e descarregamento das mercadorias também deve ser contabilizado. Não são permitidos atrasos. Porém, adiantamentos das entregas são permitidos. O mais comum são restrições de tempo devido às horas de trabalho dos motoristas.

Taxinomia Utilizada

Neste trabalho lidamos com uma variante to VRP com depósito único. Trata-se do problema de entregas de cartas na cidade de Artur Nogueira por carteiros a pé. Cada carteiro é um veículo. Os veículos são idênticos e possuem capacidade infinita, uma vez que são reabastecidos durante a rota. Entretanto, cada carteiros tem um comprimento máximo para sua rota. O comprimento é medido em tempo e é, em geral, 8h ou 6h, dependendo da instância.

Sendo assim, todo veículo deve iniciar e terminar no depósito, que é a central dos correios de Artur Nogueira, e executar uma rota limitada a 8h ou 6h dependendo da instância.

O cenário é determinístico e estático. Mercadorias são entregues e não recolhidas. Cada cliente é visitado por um único veículo. Não há restrição de tempo para se realizar a entrega, porém, todas as entregas devem ser realizadas dentro do mesmo dia.

A qualidade da solução pode ser medida por diversos fatores. Em nosso trabalho estamos interessados, primeiramente, na soma dos comprimentos da rotas e, em segundo lugar, no número de veículos, que deve ser minimizado.

Tabela 1.1: Taxonomia Utilizada

VRP	Veículo	Comprimento da Rota	Cenário	Objetivo
Depósito único	Carteiro é o veículo. Todos os carteiros são idênticos e possuem capacidade infinita	É medido em tempo e, em geral, 6 a 8 horas	Determinístico e Estático	Soma do comprimento das rotas e diminuição do número de veículos

1.2 Trabalhos Relacionados

O trabalho de Cordeau, Laporte e Mercier (2001), consiste no problema de roteamento de veículos com janelas de tempo. O autor divide seu trabalho em duas partes. A primeira é o

estudo do VRP somente com a aplicação de janelas dinâmicas (VRPTW, Vehicle Routing Problem Time Window, em português, Problema de Roteamento de Veículo com Janelas Temporais) e a segunda se concentra em acrescentar mais depósitos a esse cenário.

O trabalho do Laporte (2009) descreve a evolução do VRP desde o período em que ele surgiu até os dias atuais. Segundo ele, o VRP atraiu a atenção da comunidade de pesquisa operacional. O autor cita que devido a criação desse problema, várias metaheurísticas novas estão sendo criadas. Afirmar também que os algoritmos exatos são consideravelmente mais difíceis de serem resolvidos que os TSP (Problema do Caixeiro Viajante). Ele frisa que as velhas metaheurísticas já são projetadas e possuem um roteiro estabelecido. Atualmente é necessário, segundo o autor, a criação de algoritmos mais simples e flexíveis para lidar com uma variedade maior de restrições. Capazes de incorporar tanto cenários estocásticos como dinâmicos que são muito comuns na prática.

O trabalho de Solomon (1987) analisa os projetos do VRP em conjunto com as restrições de tempo. Ele faz a descrição de algumas heurísticas e realiza uma comparação computacional entre elas. O autor criou diversos cenários para a realização da comparação, como por exemplo, problemas de agendamento com diferentes tipos de dados e normas. Ele conseguiu concluir que existem várias heurísticas boas em diversos tipos de cenários. O autor cita que a heurística *insertion-type* apresentou resultados bons em vários tipos de cenários.

O autor afirma em seu trabalho que o *Sweep Algorithm* teve um desempenho melhor que o *Saving Algorithm* e o *Insertion Algorithm* em vários cenários em que as populações iniciais eram geradas de forma aleatória. Para problemas estruturados, o inverso acontecia. Quando o autor comparou o *Savings Algorithm* e o *Insertion Algorithm*, o segundo obteve, em geral, um resultado melhor.

A qualidade da solução no trabalho de Solomon (1987) é medida em número mínimo de veículos, tempo de gasto no planejamento, mínima distância e tempo mínimo de espera. Assim, um horário com, por exemplo, menos veículos e um maior tempo total de planejamento será melhor do que um utilizando mais veículos, mas com um menor tempo total de planejamento.

O trabalho de Prins (2004) utiliza um algoritmo genético (Genetic Algorithm, GA) híbrido e relativamente simples que supera alguns algoritmos de *Tabu Search*(TS) projetados para a resolução do VRP.

Na Tabela 1.2 são apresentados alguns resultados obtidos por Prins (2004) em suas comparações entre o seu algoritmo genético (GA) desenvolvido, 14 algoritmos de *Tabu Search*

e 2 de *Simulated Annealing* (SA). Os critérios são a distância média em % das melhores soluções conhecidas (coluna Adbks), o número de melhores soluções conhecidas recuperadas (Nbks) e o tempo médio de computação em minutos (coluna Avg time).

Tabela 1.2: Comparação das Heurísticas do VRP. (Fonte (PRINS, 2004)).

Kind	Authors	Year	Adbks %	Nbks	Avg time	Scaled
TS	Taillard	1993	0.05	12	—	—
GA	Prins	2001	0.08	10	5.2	1.46
TS	Xu-Kelly (seven problems)	1996	0.10	5	103.0	11.00
TS	Gendreau et al.	1994	0.20	8	—	—
TS	Taillard	1992	0.39	6	—	—
TS	Rego and Roucairol	1996	0.55	6	—	—
GTS	Toth and Vigo	1998	0.55	4	3.5	0.20
TS	Gendreau et al.	1991	0.68	5	—	—
TS	Rego and Roucairol	1996	0.77	4	—	—
TS	Gendreau et al.	1994	0.86	5	46.8	1.00
GA	Prins (3000 Xovers)	2001	0.90	5	0.7	0.20
TS	Osman	1993	1.01	4	26.1	?
TS	Osman	1993	1.03	3	34.0	?
SA	Osman	1993	2.09	2	—	—

O trabalho de Gendreau et al. (1999) é uma ramificação do problema de VRP clássico. Neste problema o autor trabalha com uma frota de veículos heterogêneos com diferentes capacidades e custos fixos e variáveis. Essa vertente do VRP é denominada Problema de Roteamento de Veículos de Frota Heterogênea (HVRP). O autor desenvolveu a heurística de *Tabu Search*. Soluções de alta qualidade foram encontradas em um conjunto de instâncias de benchmark, incluindo até algumas soluções bem conhecidas. Ele afirma que esse problema é mais difícil de ser resolvido por uma técnica de busca local já que a tendência natural de um processo de busca é se mover em direção a um ponto local ótimo com a composição da frota errada. Para contornar isso o autor utilizou mecanismos para diferenciar a busca, implementando um mecanismo de mudança de frota. O autor afirma que esse mecanismo foi a chave para o sucesso do algoritmo.

O Trabalho de Sacramento, Pisinger e Ropke (2019) apresenta o VRP em um contexto diferente. Ao invés de serem utilizados caminhões ou carros o autor utilizou drones para a assistência e realização das entregas de mercadorias. Assim, uma economia considerável das rotas foi obtida. O drone trabalha em conjunto com o caminhão. Enquanto o automóvel realiza uma entrega o drone pode realizar outra ao mesmo tempo. Isso permite uma economia de produtividade e de combustível para os veículos da frota. O autor define um

modelo matemático definindo um problema semelhante ao caixeiro viajante voltado para a restrição janela temporal. Ele implementou a metaheurística *Adaptive Large Neighborhood Search* devido à dificuldade de resolver grandes instâncias. Segundo o autor os testes investigam, entre outras coisas, quão benéfico a inclusão da opção de entrega de drone é comparada com a entrega de todos os itens usando exclusivamente caminhões. Além disso, uma análise de sensibilidade detalhada é realizada em vários parâmetros de interesse do drone.

O Artigo (DE BRUECKER et al., 2018) apresenta uma abordagem de aprimoramento de modelo para o problema integrado de desenvolver horários de turnos e rotas de coleta de resíduos. Dada uma quantidade variável de resíduos a ser coletada. O objetivo é encontrar horários de deslocamento de custos fixos mínimos e rotas de coleta sob uma restrição de nível de serviço. Segundo o autor, embora os horário de pico do tráfego sejam mais baratos em termos de custos de mão-de-obra, a velocidade de coleta é, em média, mais baixa do que em turnos caros e não regulares. O autor lista os seus resultados da seguinte maneira: (1) As soluções podem ser encontradas dentro de um tempo de computação razoável para instâncias reais. (2) A abordagem de aprimoramento do modelo estima com precisão os tempos de coleta necessários e, portanto, encontra consistentemente uma solução viável. (3) As soluções não apenas resultam em economias consideráveis, mas também são comprovadas como sendo quase ótimas em comparação com um limite inferior prático baseado em rotas flexíveis.

Capítulo 2

Algoritmos de Otimização para VRP

Nesse capítulo apresentaremos alguns algoritmos de otimização que foram utilizados por diversos autores para a resolução do VRP. Primeiro será apresentado alguns algoritmos exatos, em seguida, as heurísticas clássicas e posteriormente as metaheurísticas. Com isso, poderemos acompanhar quais são os melhores métodos para a resolução do problema de roteamento de veículos.

2.1 Algoritmos Exatos

Segundo Laporte (2009) os algoritmos exatos nos últimos 40 anos tem evoluído consideravelmente. Segundo o autor, um dos primeiros algoritmos de *branch-and-bound* conhecidos para o VRP apareceu em Christofides e Eilon (1969). Esse foi o primeiro algoritmo usado para a solução do problema de roteamento de veículos.

2.1.1 Algoritmos de Branch-and-Bound

Nesses algoritmos, o número de veículos m é um parâmetro de entrada. Um m -TSP é então resolvido neste grafo pela ramificação dos arcos Laporte (2009). As restrições de domínio do VRP são tratadas através de regras. Segundo o autor uma versão melhorada para este algoritmo foi proposta por Carpaneto e Toth (1980). O algoritmo do TSP foi mais tarde proposto por Laporte, Mercure e Nobert (1986). Segundo Laporte (2009) o artigo de Christofides (1976) foi o primeiro a utilizar o nome “problema de roteamento de veículos”. O algoritmo de *branch-and-bound* descreve o caminho em rotas e não em arestas. A árvore de busca terá a sua

profundidade limitada a m . Vale destacar que esse método não consegue resolver instâncias grandes.

2.2 Heurísticas Clássicas

Segundo Laporte (2009) as heurísticas clássicas são chamadas assim, pois, não se deterioram com o tempo. Essa característica está presente nos algoritmos antigos que permanecem interessantes até hoje.

2.2.1 *The Savings Algorithm*

Segundo Laporte (2009) o *Savings Algorithm* (Algoritmo de Poupança) é um algoritmo de fácil implementação e gera resultados bons.

Esse algoritmo começa com um número n de rotas uma para cada ponto de entrega ($i \in V \setminus \{0\}$).

O algoritmo então seleciona de maneira gulosa duas rotas e conecta o final de rota i com o início da rota j , assim, maximizando a economia. Esse processo só é interrompido quando não se é mais possível conectar as rotas.

Esse algoritmo sofreu várias modificações que resultaram em melhora. Um exemplo seria o caso de Golden, Magnanti e Nguyen (1977) que definiram um peso para multiplicar a distância c_{ij} . Essa técnica é útil, pois, evita rotas circulares.

2.2.2 *Cluster-First, Route-Second Heuristics*

O *cluster-first, route-second*, segundo Fisher e Jaikumar (1981), realiza primeiro o procedimento de encontrar um número n de centróides e constrói um cluster para cada centróide com o objetivo de diminuir a soma das distâncias entre os veículos e os clientes. Vale lembrar que ao mesmo tempo esse algoritmo deve respeitar as restrições impostas na fase de planejamento. Uma rota é determinada em cada cluster.

2.2.3 Heurísticas Incrementais

Segundo Laporte (2009) os algoritmos de pós-otimização também podem trazer resultados significativos. Algoritmos *interroute* trabalham com n rotas e, por isso, melhoram cada uma

separadamente. Outro caso que também é viável são os movimentos entre as rotas que trabalham com mais rotas simultaneamente. A primeira classe de algoritmos basicamente remove um ou mais clientes de uma rota e os realoca.

2.3 Metaheurísticas

Segundo Laporte (2009) as metaheurísticas podem ser divididas em três classes: busca local, busca populacional e mecanismos de aprendizagem. O autor diz que a maioria das metaheurísticas pode ser considerada como um método de melhoria. Ele afirma que a tendência atual é utilizar algoritmos híbridos ao invés de utilizar apenas um paradigma. Assim misturando algoritmos como a busca local e a populacional trazem a tona resultados mais amplos e profundos. A mistura desses algoritmos se denomina algoritmos meméticos.

Segundo Cordeau, Gendreau et al. (2005), fornecer um esquema de classificação em um domínio de classificação combinatória é uma tarefa difícil devido ao nível de complexidade dos conceitos de alguns algoritmos. O autor afirma que em um nível macro as heurísticas da VRP combinam algum dos quatro elementos a seguir: 1) Construção de uma solução inicial; 2) Procedimentos de Melhoria; 3) Mecanismos Populacionais; 4) Mecanismos de aprendizagem.

2.3.1 Heurísticas Construtivas

Cordeau, Gendreau et al. (2005) afirma que as idéias por trás dos algoritmos construtivos são muito conhecidas e estão bem documentadas. Algumas heurísticas construtivas apresentados neste trabalho são o *The Savings Algorithm* e *cluster-first route-second*.

2.3.2 Mecanismos Populacionais

Segundo Cordeau, Gendreau et al. (2005) as combinações de soluções são o mecanismo básico para a busca populacional. Alguns algoritmos conhecidos são os algoritmos genéticos e os algoritmos meméticos. Os autores definem os algoritmos genéticos como codificações de soluções chamadas de cromossomos. A cada geração nova são selecionados dois cromossomos pais e são criados dois descendentes desses pais usando um operador de crossover. Aplica-se uma mutação aleatória a cada filho (apenas em uma pequena parte). Remove-se os piores descendentes da população e adiciona-se os descendentes gerados. Esse processo pode se repetir k vezes a cada nova geração de soluções. Eles afirmam que a idéia de

combinar soluções para gerar novas é muito importante para o conceito de memória adaptativa para a solução do VRP. Além disso, afirmam que vários autores utilizam rotas de boas soluções e as utilizam como base para geração de novos descendentes.

2.3.3 Mecanismos de Aprendizado

Cordeau, Gendreau et al. (2005) citam em seu trabalho dois tipos de mecanismos de aprendizado. Os autores falam sobre as redes neurais e o algoritmo de *Ant* sobre colônias de formigas. As redes neurais, segundo os autores, são um conjunto de modelos deformáveis que são essencialmente anéis que são candidatos a se tornarem rotas de veículos viáveis. Os anéis competem por vértices através de um mecanismo aleatório que atribui um vértice a um anel cujo processo de aprendizagem esteja elevado. Segundo os autores, as redes neurais ainda não conseguem competir com a maioria das heurísticas do VRP. Eles afirmam que os algoritmos *Ant*, também são um exemplo de modelos de aprendizagem. Esse algoritmo faz analogia as formigas, pois, elas põem o feromônio em sua trilha enquanto buscam alimento. Com o tempo, os caminhos que levam a lugares que contém mais alimento possuem um número maior de feromônio.

2.3.4 Busca Local

Segundo Laporte (2009) a busca local é um algoritmo de exploração do espaço. A cada iteração desse algoritmo um estado novo é obtido. Os movimentos geralmente são feitos entre estados próximos. Exemplos desse algoritmo são a *Tabu Search*, *Simulated Annealing*, *Deterministic Annealing* e *Neighborhood Large Search*. O conteúdo da pesquisa local é composto pelas regras que definem a vizinhança a ser explorada. A seguir são apresentados algumas características da Busca Local.

Vizinhança e Ponto Ótimo Local

Segundo Neri, Cotta e Moscato (2012), para entender a busca local é essencial ter uma noção de vizinhança. O autor afirma que uma solução S é denominada vizinho de S' se a primeira puder ser alcançada a partir da última em uma única etapa (operador de manutenção).

Ele descreve a vizinhança como um conjunto $N(S)$ de todos os seus vizinhos. O autor afirma que a existência de um operador de manutenção permite a opção de dividir as vizinhanças indiretamente, assim, não sendo necessário realizar a divisão exata dos lugares.

Movimentos podem ser considerados como modificações de uma solução. Sob uma medida de distância apropriada entre as soluções. Por isso, esses movimentos são denominados como “locais” que gerou o nome do algoritmo de busca. Segundo o autor, algoritmos de pesquisa local em espaços combinatórios e contínuos têm algumas diferenças devido às diferenças nos tipos de espaços de pesquisa subjacentes. Enquanto nos espaços combinatórios as buscas são finitas para problemas finitos, nos espaços contínuos a busca é considerada infinita e não numerável.

A partir disso, algumas outras diferenças também resultam nas noções de ótimos locais e na maneira como alguém está procurando soluções candidatas aprimoradas. Um ponto ótimo local é aquele que em sua vizinhança, não existe candidato melhor que ele.

O autor diz que em problemas combinatórios, o número de soluções candidatas na vizinhança de uma solução candidata atual S é enumerável e um ponto ótimo local pode ser definido como uma solução candidata S' .

Uma forma fácil de identificar se o ponto é um ótimo local é verificando os vizinhos ao seu redor e analisar se são melhores que a solução atual. Para problemas em espaços contínuos o espaço de decisão é um conjunto denso, por isso, é composto por um conjunto infinito de pontos. O autor relata que é impossível a verificação de soluções para a busca do ponto ótimo em espaços contínuos.

Classificações da Busca Local

O Trabalho de Neri, Cotta e Moscato (2012) afirma que a classificação da busca local pode ser definida de acordo com a natureza da lógica de busca:

- Estocástica: A geração da solução de experimento ocorre de maneira aleatória.
- Determinística: A geração de experimento ocorre de forma determinada.

De acordo com a quantidade de soluções:

- Solução única: O algoritmo trabalha com apenas uma solução concentrando todas as estratégias de busca nela
- Múltiplas Soluções: O algoritmo processa múltiplas soluções e interagem com elas em conjunto para formação de soluções experimentais.

Regra do pivot:

- Descida mais íngreme: O algoritmo seleciona as melhores soluções somente depois de todas as possibilidades tiverem sido analisadas.
- Guloso: O algoritmo faz a substituição da melhor solução assim que encontrar uma melhor que ela. Depois começa a percorrer o espaço dessa nova solução.

O autor afirma que com base nessa classificação pode-se tirar duas conclusões: Todos os algoritmos de busca e os algoritmos de otimização, no geral, podem ser visto como um procedimento lógico composto por duas operações: Geração de soluções aleatórias experimentais. Seleção das melhores soluções experimentais geradas. As classificações não devem ser consideradas de maneira binária. Devem ser tratadas mais como prioridades das fases de procedimento. Por exemplo, nem sempre um algoritmo é somente determinístico ou estocástico, entretanto, existe um certo grau de lógica e determinismo estocástico. Deve-se estabelecer se o algoritmo é mais estocástico ou mais determinístico.

Segundo Neri, Cotta e Moscato (2012) algoritmos que exploram toda a vizinhança de uma solução experimental são possíveis, assim como algoritmos que centralizam a pesquisa na nova melhor solução atual. Por outro lado, várias possibilidades intermediárias também podem ser implementadas e levar a algoritmos mais eficientes do que esses dois lados extremos.

A seguir são apresentados alguns algoritmos baseados na Busca Local.

2.3.5 *Tabu Search* (TS)

Esse algoritmo é uma metaheurística da Busca Local baseado em memória com capacidades de otimização global. Segundo Neri, Cotta e Moscato (2012) esse método pode ser considerada uma sofisticada extensão do Hill Climbing básico em que a melhor solução vizinha é escolhida como a próxima configuração, mesmo que seja pior que a atual. O TS mantém uma lista tabu de movimentos: uma solução vizinha é aceita somente se o movimento correspondente não for tabu. O autor afirma que o usuário pode escolher a regra que quiser para colocar como Tabu. Esse status Tabu não é permanente, podendo ser alterado conforme um número de etapas realizadas pelo algoritmo. Esse valor é denominado *Tabu Tenure*. Esse número pode ser fixo ou variar de acordo com o tempo. O último caso pode ajudar a reduzir ciclos longos na busca. Para se ter um algoritmo *Tabu Search* completo segundo o autor deve-se possuir estratégias adicionais para intensificar e diversificar a busca. Ele cita vários exemplos de estratégias como

a baseada em frequência e também a incorporação de várias vizinhanças em que o algoritmo oscila.

2.3.6 *Hill Climbing (HC)*

De acordo com Neri, Cotta e Moscato (2012) a maneira mais simples de se executar a busca local em um espaço combinatório é perturbando uma solução S experimental e obtendo uma nova solução S' . Se a solução S' gerada for melhor que a solução S , então, se substitui S por S' .

Esse procedimento é denominado subida da montanha (Hill Climbing, HC). É descrito como um procedimento de busca local sem memória e sem o conceito de vizinhança. Podem haver várias variantes do Hill Climbing .

Segundo Neri, Cotta e Moscato (2012) existe a descida mais íngreme, ou seja, um algoritmo que explora a vizinhança $N(s)$ completa da solução atual s , escolhe o melhor vizinho, aceita se for melhor.

Também existe HC aleatório (um algoritmo que escolhe um único vizinho aleatório s pertence $N(x)$ e aceita se é melhor que s).

No primeiro caso, segundo o autor, a solução irá encontrar um ponto ótimo local. No segundo caso, apenas o limite computacional se aplica, a menos que o algoritmo rastreie os vizinhos gerados e seja capaz de evitar duplicatas ou detectar quando a vizinhança está totalmente explorada.

Em muitos casos, segundo o autor, podemos possuir vizinhanças muito grandes que impossibilitam uma exploração por completo. Nessas situações, o autor, apresenta duas soluções. A primeira é utilizar uma amostra aleatória de um certo tamanho da vizinhança, para assim, realizar os procedimentos de busca e otimização. A segunda solução seria utilizar um HC aleatório mais simples.

2.3.7 **Considerações**

Os conceitos de busca local juntamente com os algoritmos apresentados nesse capítulo são importantes para o entendimento dos algoritmos que foram utilizados nesse trabalho e os seus resultados. Em nosso contexto os dois algoritmos utilizados para a resolução desse trabalho são uma metaheurística da busca local. Ambos são determinísticos e geram soluções únicas.

Capítulo 3

Metodologia Experimental

Nesse trabalho desenvolvemos dois algoritmos capazes de solucionar o problema de roteamento na entrega de cartas da cidade de Artur Nogueira. O objetivo é reduzir o número de carteiros e a soma dos comprimentos das rotas.

As instâncias utilizadas nesse trabalho foram obtidas em Zeni et al. (2016) e representam de forma realística os pontos de entrega e as ruas da cidade de Artur Nogueira. Na Figura 3.1 é apresentado o mapa da cidade de Artur Nogueira e na Figura 3.2 é apresentado os pontos de entrega dentro do mapa. Repare que os pontos azuis na Figura 3.2 representam os diferentes pontos de entrega.

Em nosso contexto, o carteiro é o veículo e possui um limite de comprimento da rota. Esse limite representa a quantidade de horas trabalhadas em um dia. Além disso, todos os carteiros devem obrigatoriamente sair e voltar ao depósito.

Os algoritmos usados para a resolução desse problema foram desenvolvidos na linguagem de programação Java. O programa realiza a leitura dos arquivos que contém os pontos de entrega e a matriz de adjacência. Foi implementada a restrição de capacidade do carteiro.

O primeiro algoritmo é um método de troca vertice (TrocaV) de uma posição inicial a uma posição final, baseado em alguns parâmetros da Busca Local citados pelo Laporte (2009). Enquanto o segundo é o algoritmo 2-opt citado por Beasley (1983) que corresponde a troca de duas arestas.

Os dois algoritmos consistem em uma busca local. Também podem ser classificados como algoritmos de melhoria incremental.

Em ambos os casos, foi otimizada a rota de apenas um veículo, ou seja, foi resolvido um TSP. Após a otimização do TSP, a rota única foi quebrada em rotas de comprimento máximo

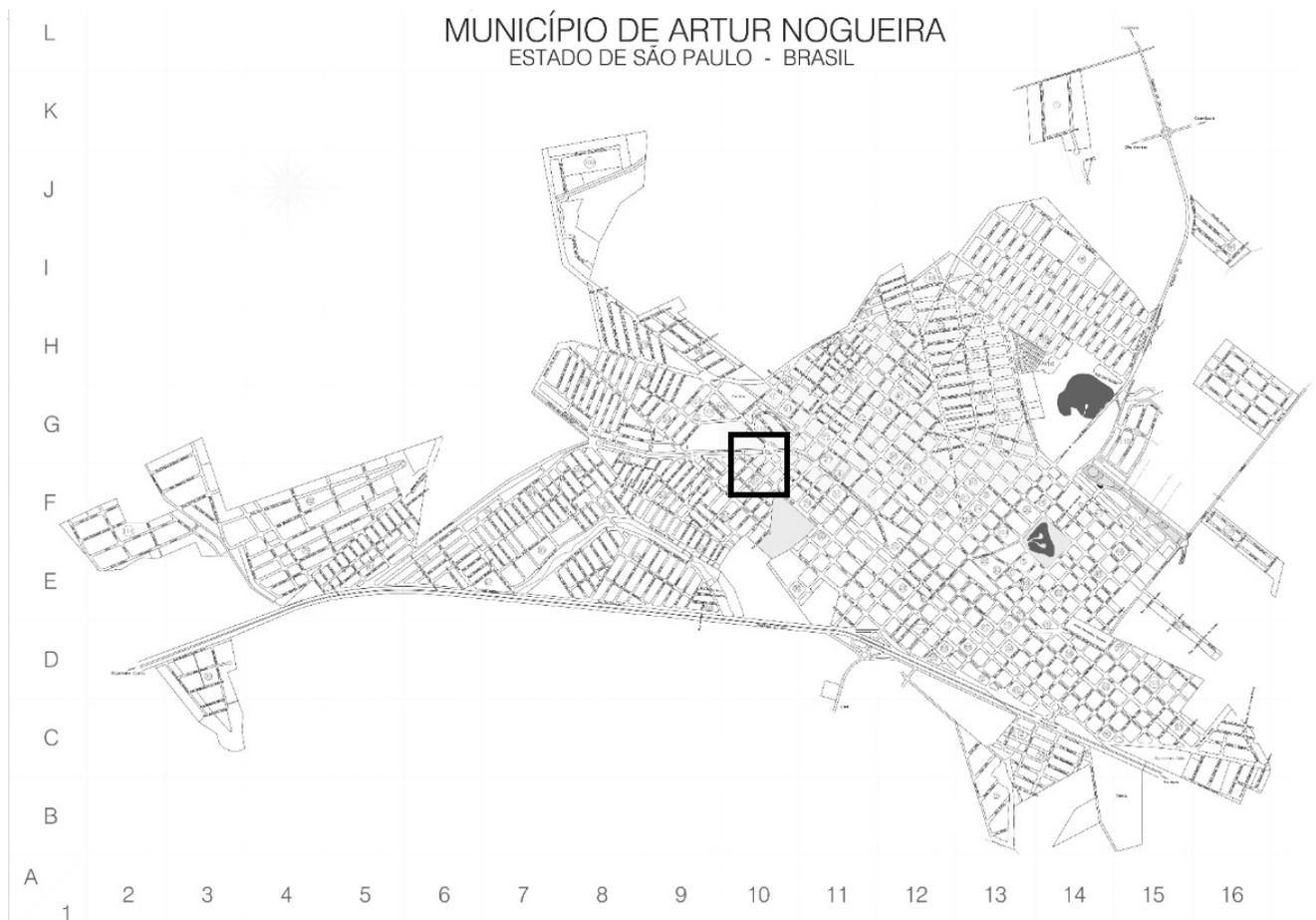


Figura 3.1: Modelo do mapa utilizado



Figura 3.2: Pontos de entrega em parte do mapa

de 6h ou 8h de acordo com a instância. Ou seja, executamos a metodologia de Beasley (1983) chamada *route-first cluster-second*. Primeiro a otimização da rota, depois a quebra em veículos.

3.1 Algoritmo de troca de vértice entre dois pontos da rota (TrocaV)

O algoritmo de troca de vértices consiste em selecionar duas posições, origem e destino, em uma rota que não sejam iguais e nem consecutivos e realizar a troca de um vértice da origem para o destino. Como critério de parada foi estabelecido o mínimo local, ou seja, após uma iteração do algoritmo sem melhoras na vizinhança. Para encontrar uma melhora utilizamos uma variável chamada Delta. Essa variável calcula a diferença entre o custo da rota antes e depois do movimento de troca de vértices. Se a variável obtiver um resultado negativo, significa que o algoritmo encontrou uma rota com custo menor que a atual e então a troca é realizada. Para que o melhor Delta seja encontrado o TrocaV realiza todas as trocas de vértices não consecutivos possíveis e assim retorna a rota que obteve melhor resultado.

Depois da otimização ser encerrada, o algoritmo entra na fase de clusterização. A rota única é quebrada em várias partes. O critério de divisão da rota é a restrição de comprimento, em geral, 6 a 8 horas dependendo da instância. Quando a restrição de comprimento da rota é ultrapassado, acrescenta-se um depósito depois do último ponto visitado. Quando um depósito é acrescentado, a rota é dividida.

Note que o movimento de um vértice define uma vizinhança para uma solução.

O seguinte pseudocódigo representa o algoritmo:

1. Geração de Solução Inicial aleatória contendo apenas 1 rota.
2. Enquanto houver vizinho melhor
 - (a) Delta = calcular diferença de custo de troca do vértice entre os dois pontos
 - (b) encontrar o melhor Delta
 - (c) Se Melhor Delta < 0, então:
realizar troca(posição1, posição2)
 - (d) caso contrário
Encerrar otimização
3. Clusteriza(Solucao)
4. Retorna Solução Clusterizada

Algoritmo 1: Pseudocódigo Algoritmo de Troca

3.2 Algoritmo 2-Opt

Segundo Hasegawa, Ikeguchi e Aihara (1997) o algoritmo 2-opt é uma meta-heurística que apresenta resultados próximo do ótimo. O autor descreve o algoritmo como uma troca entre dois pontos de diferentes arestas. Suponha que temos um conjunto de vértices A–B–H–I–J–C–D–E–F–G–A e gostaríamos de trocar a aresta A-B e a aresta C-D.

Nesse exemplo, o seguinte movimento é executado: o segundo vértice da primeira aresta irá se mover para a posição do primeiro vértice da segunda aresta. Depois de realizada essa movimentação, o primeiro vértice da segunda aresta é movimentado para o segundo vértice da primeira aresta. Todos os elementos entre esses dois vértices são invertidos. Na figuras 3.3 e 3.4 é possível enxergar essa movimentação de vértices entre as arestas.

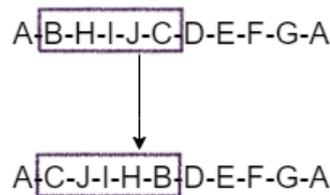


Figura 3.3: Movimento de troca - 2-Opt

Depois que o 2-Opt é executado, é realizada a etapa de clusterização. Assim como o TrocaV, o critério de divisão é o comprimento da rota, em geral de 6 a 8 horas dependendo da instância. Quando o comprimento da rota é ultrapassado, acrescenta-se um depósito dividindo a rota. Assim a rota que era única é quebrada em diversas outras rotas menores que respeitam a restrição de comprimento estabelecida.

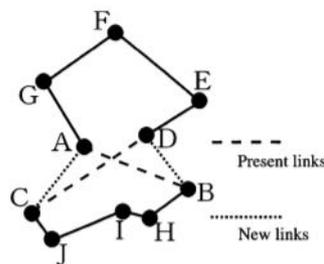


Figura 3.4: Estrutura de pontos - Movimento de troca. Fonte: Hasegawa, Ikeguchi e Aihara (1997)

O pseudocódigo do Algoritmo 2 representa o algoritmo desenvolvido:

1. Geração de Solução Inicial aleatória contendo apenas 1 rota.
2. Enquanto houveMelhora = true
 - (a) Para todas as arestas presentes na rota
 - (b) Selecionar duas arestas não consecutivas da solução e calcular o custo do movimento de troca de vértices entre as arestas
 - (c) Delta = calcular a diferença de custo da solução atual para a solução gerada
 - (d) Se $\Delta < 0$, então:
 - realiza troca de vértices entre as arestas
 - (e) houveMelhora=true
3. Se houveMelhora=false
 - Encerrar otimização
4. Clusteriza(Solucao)
5. Retorna Solução Clusterizada

Algoritmo 2: Pseudocódigo do algoritmo 2-opt

Capítulo 4

Resultados

4.1 Ambiente e Cenário dos Experimentos

Os dois algoritmos foram executados em um cenário determinístico e estático. Portanto, não foi necessário a repetição dos experimentos, pois, o resultado sempre será o mesmo.

Em todos os experimentos, o custo diz respeito à soma dos comprimentos das rotas.

4.2 Resultado dos Experimentos

O algoritmo 2-Opt teve desempenho superior ao TrocaV. Por ser mais eficiente, o algoritmo 2-Opt foi executado em instâncias com até 10.000 vértices.

O algoritmo TrocaV é menos eficiente e foi executado para instâncias de até 1000 pontos de entrega. Conforme mostra a Figura 4.1, podemos notar como o tempo de processamento aumenta quando a quantidade de entregas é 1000. Ele sobe consideravelmente em relação ao tempo de processamento quando a quantidade de entregas é 500.

A Tabela 4.1 realiza uma comparação entre os dois algoritmos. Observe que entre as instâncias RealWorldPostToy 20-1 e RealWorldPostToy 200-2 o número de carteiros continua igual, porém, o custo do algoritmo TrocaV foi maior em todos os experimentos. Um exemplo é a instância RealWorldPostToy 1000-0 que para o algoritmo TrocaV alcançou o total de 8 carteiros enquanto o 2-opt obteve 4.

A Figura 4.2 representa o desenho da solução do algoritmo 2-opt da instância RealWorldPostToy 20-1 e a Figura 4.3 representa o mesmo desenho para o algoritmo TrocaV. As Figuras 4.4 e 4.5 representam uma comparação entre o algoritmo TrocaV e o 2-opt para a

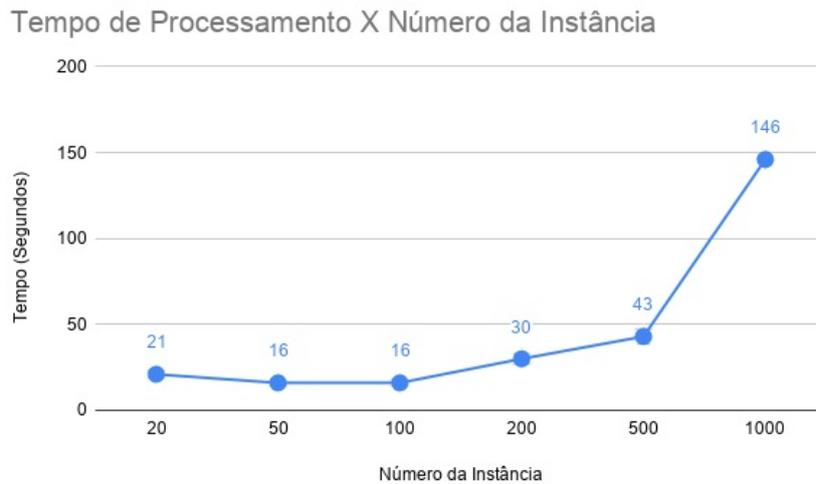


Figura 4.1: Gráfico Tempo de Processamento (segundos) X Número de Instâncias para o algoritmo TrocaV.

instância RealWorldPostToy 1000-0. Na Tabela 4.2 é apresentado os resultados do algoritmo 2-opt resolvido como um TSP, ou seja, com uma única rota.

Na Tabela 4.3 são apresentados os resultados do algoritmo 2-opt para o VRP. A Tabela 4.4 apresenta os resultados do algoritmo TrocaV para o VRP. A Tabela 4.5 contém os resultados do algoritmo de TrocaV para o TSP.

Tabela 4.1: Relação entre o algoritmo 2-opt e o de troca de vértices (TrocaV)

Nome Instância	tempo 2-opt	tempo TrocaV	custo 2-opt	custo TrocaV	2-opt n. cart.	TrocaV n. cart.
RealWorldPostToy 20-1	15s	21s	11.791,28	12.982,02	1	1
RealWorldPostToy 50-0	13s	16s	17.798,86	20.083,62	1	1
RealWorldPostToy 100-0	13s	16s	25.765,81	29.557,20	2	2
RealWorldPostToy 200-2	14s	30s	32.212,67	42.030,17	2	2
RealWorldPostToy 500-2	14s	43s	56.298,85	101.485,30	3	5
RealWorldPostToy 1000-0	14s	146s	76.701,12	166.364,74	4	8

A Tabela 4.6 compara os resultados do 2-opt com melhores resultados obtidos dentre os Algoritmos Genéticos (AG) desenvolvidos no Trabalho de Muller (2019). Os AG desenvolvidos pelo autor são canônicos e misturados com busca local, possui divisão das rotas no momento do crossover. No trabalho dele foi desenvolvido uma opção chamada *LS Enable*. Quando essa opção está habilitada, é executado um algoritmo de melhoria incremental. Foram utilizados

Tabela 4.2: Resultados Obtidos - Algoritmo 2-opt sem clusterização, ou seja, única rota ou TSP.

Nome	Número Entregas	Carteiros = Custo/ Max_Rota	Tempo	Custo
RealWorldPostToy 20-1	20	0.55	14s	11.442,55
RealWorldPostToy 50-0	50	0.82	13s	17.798,86
RealWorldPostToy 100-0	100	1.07	13s	17.798,86
RealWorldPostToy 200-2	200	1.46	13s	31.512,13
RealWorldPostToy 500-2	500	2.45	14s	52.988,07
RealWorldPostToy 1000-0	1000	3.24	22s	69.956,20
RealWorldPostToy 5000-0	5000	6.69	42s	144.579,21
RealWorldPostToy 10000-0	10000	9.06	2min 9s	195.786,03

Tabela 4.3: Resultados Obtidos - Algoritmo 2-opt para PostVRP

Nome	Número Entregas	Carteiros	Tempo	Custo
RealWorldPostToy 20-1	20	1	15s	11.442,55
RealWorldPostToy 50-0	50	1	14s	17.798,86
RealWorldPostToy 100-0	100	2	13s	25.765,81
RealWorldPostToy 200-2	200	2	13s	32.212,67
RealWorldPostToy 500-2	500	3	14s	56.298,85
RealWorldPostToy 1000-0	1000	4	22s	76.701,12
RealWorldPostToy 5000-0	5000	8	44s	158.155,65
RealWorldPostToy 10000-0	10000	10	1min 55s	195.786,03

Tabela 4.4: Resultados Obtidos - Algoritmo TrocaV para o PostVRP

Nome	Número Entregas	Carteiros	Tempo	Custo
RealWorldPostToy 20-1	20	1	21s	12.982,02
RealWorldPostToy 50-0	50	1	16s	20.083,62
RealWorldPostToy 100-0	100	2	16s	29.557,20
RealWorldPostToy 200-2	200	2	30s	42.030,17
RealWorldPostToy 500-0	500	2	30s	42.030,17
RealWorldPostToy 1000-0	1000	8	2min 35s	166.364,74

Tabela 4.5: Resultados Obtidos - Algoritmo TrocaV sem clusterização, ou seja, rota única ou TSP

Nome	Número Entregas	Carteiros	Tempo	Custo
RealWorldPostToy 20-1	20	0.60	21s	12.982,02
RealWorldPostToy 50-0	50	0.93	16s	20.083,62
RealWorldPostToy 100-0	100	1.33	16s	28.832,94
RealWorldPostToy 200-2	200	1.90	30s	41.108,90
RealWorldPostToy 500-0	500	4.21	30s	90.877,79
RealWorldPostToy 1000-0	1000	7.25	2min 35s	156.604,48

dois operadores de crossover chamados *Best Cost Route Crossover* (BCR) e *Order Based Crossover* (OX) ambos foram testados sozinhos e depois com a utilização da busca local.

Tabela 4.6: Relação entre o algoritmo 2-opt e o AG

Nome Instância	tempo 2-opt	tempo AG	custo 2-opt	custo AG	2-opt n. cart.	AG n. cart.
RealWorldPostToy 20-1	15s	60s	11.791,28	11.253,03	1	1
RealWorldPostToy 50-0	13s	180s	17.798,86	17.762,85	1	1
RealWorldPostToy 100-0	13s	420s	25.765,81	23.439,94	2	2
RealWorldPostToy 200-2	14s	1980s	32.212,67	32.211,41	2	2
RealWorldPostToy 500-2	14s	9240s	56.298,85	60.970,05	3	3
RealWorldPostToy 1000-0	22s	35.580s	76.701,12	99.477,30	4	5
RealWorldPostToy 5000-0	44s	328.860s	158.155,65	365.962,00	4	17

A Tabela 4.7 apresenta uma comparação entre os resultados do Algoritmo TrocaV com o AG apresentado pelo Muller (2019).

Tabela 4.7: Relação entre o algoritmo TrocaV e o AG

Nome Instância	tempo TrocaV	tempo AG	custo TrocaV	custo AG	TrocaV n. cart.	AG n. cart.
RealWorldPostToy 20-1	21s	60s	12.982,02	11.253,03	1	1
RealWorldPostToy 50-0	16s	180s	20.083,62	17.762,85	1	1
RealWorldPostToy 100-0	16s	420s	29.557,20	23.439,94	2	2
RealWorldPostToy 200-2	30s	1980s	42.030,17	32.211,41	2	2
RealWorldPostToy 500-2	43s	9240s	101.485,30	60.970,05	5	3
RealWorldPostToy 1000-0	146s	35.580s	166.364,74	99.477,30	8	5

Observe que o custo do AG é melhor que o 2-opt nas instâncias RealWorldPostToy 20-1 até RealWorldPostToy 200-2. Porém, em questão de tempo de processamento, o 2-Opt obteve resultados melhores. Em instâncias com mais de 200 pontos de entrega o 2-Opt se mostrou mais eficiente apresentando soluções com melhor qualidade e tempo de processamento menor que o AG.

Comparando os resultados do AG com o TrocaV podemos observar que em questão de tempo de processamento o TrocaV teve desempenho superior. Porém, o custo do AG foi menor em todas as instâncias testadas. O número de carteiros do TrocaV foi maior em instâncias

maiores que 200 pontos de entrega. Podemos notar que o AG apresentou soluções com mais qualidade que o TrocaV, entretanto, o tempo de processamento do AG é muito alto.

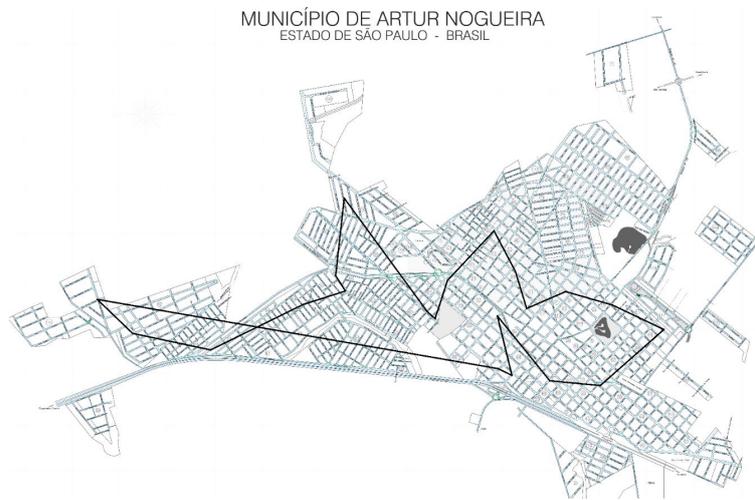


Figura 4.2: Rota da Solução RealWorldPostToy 20-1 Algoritmo 2-opt.



Figura 4.3: Rota da Solução RealWorldPostToy 20-1 Algoritmo TrocaV.



Figura 4.4: Rota da Solução RealWorldPostToy 1000-0 Algoritmo 2-Opt.



Figura 4.5: Rota da Solução RealWorldPostToy 1000-0 Algoritmo TrocaV.

Capítulo 5

Conclusões

Este trabalho utilizou o algoritmo de TrocaV e o 2-opt que podem ser usados tanto para resoluções de TSP como também VRP. Nós trabalhamos com o PostVRP que possui restrições nas capacidades de entrega dos carteiros.

Foi realizado a comparação a qualidade das soluções desses dois algoritmos. O 2-opt obteve desempenho superior, tanto em quantidade de carteiros quanto em questões de tempo de processamento. O algoritmo de TrocaV se mostrou ineficiente para instâncias maiores que 1000 pontos de entrega, pois, o tempo de processamento em relação a instância com 500 pontos de entrega foi quase quatro vezes maior. A qualidade da solução também piorou bastante. Enquanto a instância com 500 pontos de entrega teve o resultado de 5 carteiros, a instância com 1000 pontos teve um resultado de 8 para o algoritmo de TrocaV. Para as mesmas instâncias, o algoritmo 2-opt obteve 3 e 4 carteiros, respectivamente.

O 2-opt teve resultados com maior qualidade. Podemos notar que até a instância RealWorldPostToy 200-2 os dois algoritmos tiveram a mesma quantidade de carteiros. Porém, o custo da solução do 2-opt sempre foi inferior ao do algoritmo de Troca, assim também como o tempo de processamento. Quando executada a instância RealWorldPostToy 1000-0 o algoritmo 2-opt obteve um resultado de 4 carteiros e o algoritmo de TrocaV 8. O custo do algoritmo de TrocaV nessa instância foi maior que o dobro do resultado obtido pelo 2-opt.

Em um trabalho futuro será interessante o desenvolvimento do algoritmo Tabu Search (TS) para comparação com os algoritmos já implementados. Assim, executar o TS e o 2-Opt em instâncias maiores do que as já testadas nesse trabalho e comentar os resultados.

Referências bibliográficas

- APPA, G. M.; PITSOULIS, L.; WILLIAMS, H. P. **Handbook on modelling for discrete optimization**. [S.l.]: Springer Science & Business Media, 2006. v. 88.
- BEASLEY, J. E. Route first—cluster second methods for vehicle routing. **Omega**, Elsevier, v. 11, n. 4, p. 403–408, 1983.
- CARPANETO, G.; TOTH, P. Some new branching and bounding criteria for the asymmetric travelling salesman problem. **Management Science**, INFORMS, v. 26, n. 7, p. 736–743, 1980.
- CHRISTOFIDES, N. The vehicle routing problem. **Revue française d'automatique, informatique, recherche opérationnelle. Recherche opérationnelle**, EDP Sciences, v. 10, n. V1, p. 55–70, 1976.
- CHRISTOFIDES, N.; EILON, S. An algorithm for the vehicle-dispatching problem. **Journal of the Operational Research Society**, Springer, v. 20, n. 3, p. 309–318, 1969.
- CORDEAU, J.-F.; GENDREAU, M. et al. New heuristics for the vehicle routing problem. In: **LOGISTICS systems: design and optimization**. [S.l.]: Springer, 2005. p. 279–297.
- CORDEAU, J.-F.; LAPORTE, G.; MERCIER, A. A unified tabu search heuristic for vehicle routing problems with time windows. **Journal of the Operational research society**, Springer, v. 52, n. 8, p. 928–936, 2001.
- DE BRUECKER, P. et al. A model enhancement approach for optimizing the integrated shift scheduling and vehicle routing problem in waste collection. **European Journal of Operational Research**, Elsevier, v. 266, n. 1, p. 278–290, 2018.
- DROR, M.; TRUDEAU, P. Savings by split delivery routing. **Transportation Science**, INFORMS, v. 23, n. 2, p. 141–145, 1989.
- FISHER, M. L.; JAIKUMAR, R. A generalized assignment heuristic for vehicle routing. **Networks**, Wiley Online Library, v. 11, n. 2, p. 109–124, 1981.
- GENDREAU, M.; LAPORTE, G.; MUSARAGANYI, C.; TAILLARD, É. D. A tabu search heuristic for the heterogeneous fleet vehicle routing problem. **Computers & Operations Research**, Elsevier, v. 26, n. 12, p. 1153–1173, 1999.
- GOLDEN, B. L.; MAGNANTI, T. L.; NGUYEN, H. Q. Implementing vehicle routing algorithms. **Networks**, Wiley Online Library, v. 7, n. 2, p. 113–148, 1977.
- HASEGAWA, M.; IKEGUCHI, T.; AIHARA, K. Combination of chaotic neurodynamics with the 2-opt algorithm to solve traveling salesman problems. **Physical Review Letters**, APS, v. 79, n. 12, p. 2344, 1997.

- LAHYANI, R.; KHEMAKHEM, M.; SEMET, F. Rich vehicle routing problems: From a taxonomy to a definition. **European Journal of Operational Research**, Elsevier, v. 241, n. 1, p. 1–14, 2015.
- LAPORTE, G. Fifty years of vehicle routing. **Transportation Science**, INFORMS, v. 43, n. 4, p. 408–416, 2009.
- LAPORTE, G.; MERCURE, H.; NOBERT, Y. An exact algorithm for the asymmetrical capacitated vehicle routing problem. **Networks**, Wiley Online Library, v. 16, n. 1, p. 33–46, 1986.
- MULLER, F. F. Algoritmos Genéticos Aplicados Ao Problema De Roteamento De Veículos, 2019.
- NERI, F.; COTTA, C.; MOSCATO, P. **Handbook of memetic algorithms**. [S.l.]: Springer, 2012. v. 379.
- PRINS, C. A simple and effective evolutionary algorithm for the vehicle routing problem. **Computers & Operations Research**, Elsevier, v. 31, n. 12, p. 1985–2002, 2004.
- SACRAMENTO, D.; PISINGER, D.; ROPKE, S. An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. **Transportation Research Part C: Emerging Technologies**, Elsevier, v. 102, p. 289–315, 2019.
- SOLOMON, M. M. Algorithms for the vehicle routing and scheduling problems with time window constraints. **Operations research**, Informs, v. 35, n. 2, p. 254–265, 1987.
- ZENI, G. A.; MENZORI, M.; MARTINS, P. S.; MEIRA, L. A. Vrpbench: A vehicle routing benchmark tool. **arXiv preprint arXiv:1610.05402**, 2016.