



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

UM ALGORITMO HÍBRIDO BASEADO EM COLÔNIA DE FORMIGAS E PROGRAMAÇÃO LINEAR APLICADO AO PROBLEMA DE ROTEAMENTO DE VEÍCULOS CAPACITADOS.

LARISSA CAMILA PAPA

Orientador: Dr. Flávio Vinícius Cruzeiro Martins
Centro Federal de Educação Tecnológica de Minas Gerais

Coorientador: Dr. Rodrigo Tomás Nogueira Cardoso
Centro Federal de Educação Tecnológica de Minas Gerais

BELO HORIZONTE
AGOSTO DE 2017

LARISSA CAMILA PAPA

**UM ALGORITMO HÍBRIDO BASEADO EM COLÔNIA
DE FORMIGAS E PROGRAMAÇÃO LINEAR
APLICADO AO PROBLEMA DE ROTEAMENTO DE
VEÍCULOS CAPACITADOS.**

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: Dr. Flávio Vinícius Cruzeiro Martins
Centro Federal de Educação Tecnológica de Minas Gerais

Coorientador: Dr. Rodrigo Tomás Nogueira Cardoso
Centro Federal de Educação Tecnológica de Minas Gerais

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL
BELO HORIZONTE
AGOSTO DE 2017

Papa, Larissa Camila
P213a Um algoritmo híbrido baseado em colônia de formigas e programação linear aplicado ao problema de roteamento de veículos capacitados / Larissa Camila Papa. – 2017.
xv, 69 f. : il., tab., grafs.

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Modelagem Matemática e Computacional.
Orientador: Flávio Vinícius Cruzeiro Martins.
Coorientador: Rodrigo Tomás Nogueira Cardoso.
Dissertação (mestrado) – Centro Federal de Educação Tecnológica de Minas Gerais.

1. Sistemas multiagentes – Teses. 2. Algoritmo híbrido – Teses.
3. Levantamento de rotas – Processamento de dados – Teses.
4. Programação linear – Teses. I. Martins, Flávio Vinícius Cruzeiro.
II. Cardoso, Rodrigo Tomás Nogueira. III. Centro Federal de Educação Tecnológica de Minas Gerais. IV. 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

**“UM ALGORITMO HÍBRIDO BASEADO EM COLÔNIA DE FORMIGAS E
PROGRAMAÇÃO LINEAR APLICADO AO PROBLEMA DE ROTEAMENTO DE
VEÍCULOS CAPACITADOS”**

Dissertação de Mestrado apresentada por **Larissa Camila Papa**, em 10 de agosto de 2017, 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. Rodrigo Tomás Nogueira Cardoso (Coorientador)
Centro Federal de Educação Tecnológica de Minas Gerais

Prof. Dr. Rafael Frederico Alexandre
Universidade Federal de Ouro Preto

Prof. Dr. Vinicius Fernandes dos Santos
Universidade Federal de Minas Gerais

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

Visto e permitida a impressão,

Prof. Dr. José Geraldo Peixoto de Faria
Coordenador do Programa de Pós-Graduação em
Modelagem Matemática e Computacional

*A Deus toda glória
Para sempre, amém.*

Agradecimentos

À família, aos amigos, e aos grandes mestres e educadores Dr. Flávio Vinícius Cruzeiro Martins, Dr. Rodrigo Tomás Nogueira Cardoso e, não menos importante, Dr. João Sarubbi.

Ao Centro Federal de Educação Tecnológica de Minas Gerais, ao LAHMO, ao PPGMMC, e às pessoas incríveis que conheci nesta instituição.

Um obrigada especial à Rafaela Priscila e ao Rodrigo Borges, grandes apoiadores e companheiros desta caminhada.

Durante esse tempo fiz grandes amigos na Fundação João Pinheiro, dos quais também não deixarei de citar. Meus sinceros agradecimentos a todos vocês, com um carinho especial a turma da ATI: Renata, Thiago, Netinho, Cris UFO, João, Belle, Rodrigo, Marcus, Harlei, Denilson, Laura e Reinaldo.

Resumo

Neste trabalho é abordado o Problema de Roteamento de Veículos Capacitados. Ele compreende a obtenção de um conjunto de rotas, que devem ser percorridas por uma frota de veículos homogêneos e de igual capacidade, atendendo assim as demandas de um conjunto de clientes. Seu objetivo é a minimização do custo total das rotas, sabendo-se que elas devem iniciar e terminar no depósito central, além de cada cliente somente pode ser atendido uma única vez por um único veículo.

Este problema possui natureza NP-difícil, sendo comumente resolvido por meio de heurísticas. Outra técnica utilizada é a programação linear inteira, cuja particularidade corresponde a confiança na solução obtida, uma vez que por meio da solução final retornada pode-se provar que é o ótimo global. Como desvantagem, os problemas de natureza combinatória são, na maioria dos casos, computacionalmente custosos, tornando improvável encontrar uma solução ótima em tempo computacional aceitável.

A proposta deste trabalho é desenvolver um algoritmo híbrido que combine a natureza ágil das heurísticas com a precisão dos métodos de solução por Programação Linear. A heurística utilizada foi a Otimização por Colônia de Formigas (ACO), a qual irá trabalhar de forma híbrida com o CPLEX, software distribuído pela IBM.

Estuda-se a melhor forma de interação entre estes dois métodos, analisando os seus possíveis modos de comunicação. Estes compreendem duas diferentes formas de hibridização, uma caracterizada pela ausência do processo de realimentação, e a outra fazendo uso deste recurso.

Os resultados obtidos pelas hibridizações foram comparados com os produzidos pelos métodos executados individualmente, bem como com as soluções presentes na literatura.

Palavras-chave: Problema de Roteamento de Veículos Capacitados. Algoritmos Híbridos. Colônia de Formigas. Programação Linear.

Abstract

In this work the Capacitated Vehicle Routing Problem is approached. It comprises obtaining a set of routes, which must be covered by a fleet of homogeneous vehicles of the same capacity, thus meeting the demands of a group of customers. Their goal is to minimize the total cost of the routes, knowing that they must start and end in the central warehouse, and each customer can only be serviced once by a single vehicle.

This problem is NP-difficult in nature and is commonly solved by means of heuristics. Another technique used is the integer linear programming, whose particularity corresponds to the confidence in the obtained solution, since by means of the returned final solution it is possible to prove this is the global optimum. As a disadvantage, combinatorial problems are computationally costly, making it unlikely to find an optimal solution at acceptable computational time.

The purpose of this work is to develop a hybrid algorithm that combines the agile nature of heuristics with the precision of the solution methods by Linear Programming. The heuristic used was the Ant Colony Optimization (ACO), which will work in a hybrid way with CPLEX, software distributed by IBM.

The best way of interacting between these two methods is analyzed, analyzing their possible modes of communication. These comprise two different forms of hybridization, one characterized by the absence of the feedback process, and the other making use of this feature.

The results obtained by the hybridizations were compared with those produced by the methods performed individually, as well as with the solutions present in the literature.

Keywords: Capacitated Vehicle Routing Problem. Hybrid Algorithm. Ant Colony. Linear Programming.

Lista de Figuras

Figura 1 – Fluxo de resolução de um problema em Pesquisa Operacional	6
Figura 2 – Estratégias de resolução do VRP	7
Figura 3 – Representação possível para o problema de roteamento de veículos.	9
Figura 4 – Experiência realizada com formigas reais.	16
Figura 5 – Exemplo de um caso de atendimento às restrições (9) a (11).	24
Figura 6 – Exemplo de um caso de não atendimento à restrição (10).	24
Figura 7 – Exemplo de um caso de atendimento à restrição (13).	25
Figura 8 – Representação de uma formiga na ACO	26
Figura 9 – Representação gráfica de como a ACO assegura que a capacidade do veículo será respeitada.	27
Figura 10 – Modelos de comunicação do algoritmo híbrido	30
Figura 11 – Resultados dos testes realizados no CPLEX	35
Figura 12 – Resultados da ACO nos primeiros 30 minutos de execução.	37
Figura 13 – Evolução dos resultados durante o processo de realimentação do algoritmo (as partes cinzas indicam que não houve execução)	40
Figura 14 – Caracterização dos tipos de instâncias testadas	43
Figura 15 – Resultados para E-n23-k3	44
Figura 16 – Resultados para E-n22-k4	45
Figura 17 – [Grupo E] Boxplot	48
Figura 18 – [Grupo A] Boxplot	48
Figura 19 – [Grupo P] Boxplot	49
Figura 20 – [Grupo B] Boxplot	49
Figura 21 – [Grupo CMT] Boxplot	49
Figura 22 – [Grupo E] Teste de <i>Kruskal-Wallis</i>	50
Figura 23 – [Grupo A] Teste de Normalidade	51
Figura 24 – [Grupo A] Teste de <i>Kruskal-Wallis</i>	51
Figura 25 – [Grupo P] Teste de <i>Kruskal-Wallis</i>	52
Figura 26 – [Grupo B] Teste de Normalidade	52
Figura 27 – [Grupo B] Teste de <i>Kruskal-Wallis</i>	53
Figura 28 – Representação gráfica do teste de Tukey para o grupo de instâncias B	53
Figura 29 – Resultado do teste de Tukey para o grupo de instâncias B	54
Figura 30 – [Grupo CMT] Teste de Normalidade	54
Figura 31 – Resultado ANOVA	55
Figura 32 – Resultado do teste de Tukey para o grupo de instâncias CMT	55
Figura 33 – Estrutura de um arquivo com extensão <i>.lp</i>	59
Figura 34 – Estrutura de um arquivo com extensão <i>.mst</i>	60

Figura 35 – Interpretação dos arquivos <i>.lp</i> e <i>.mst</i>	61
Figura 36 – Parte do processo de criação do arquivo <i>.mst</i>	61
Figura 37 – Arquivo <i>.txt</i> gerado pela ACO contendo os dados da melhor formiga . . .	62
Figura 38 – Uma solução repassada pela ACO sendo interpretada pelo CPLEX . . .	63
Figura 39 – Arquivo <i>.txt</i> produzido pelo CPLEX	63

Lista de Tabelas

Tabela 1 – Resultados computacionais para os métodos heurístico e exato	36
Tabela 2 – Evolução dos resultados médios da ACO	38
Tabela 3 – Resultados do modelo de comunicação com realimentação	39
Tabela 4 – Resultados do modelo de comunicação sem realimentação	41
Tabela 5 – Resultados das instâncias do conjunto E (* são valores ótimos).	44
Tabela 6 – Resultados das instâncias do conjunto A.	46
Tabela 7 – Resultados das instâncias do conjunto P (* são valores ótimos).	46
Tabela 8 – Tempo de execução do CPLEX executado individualmente e em conjunto com o ACO (método híbrido) para encontrar o valor ótimo	47
Tabela 9 – Resultados das instâncias do conjunto B.	47
Tabela 10 – Resultados das instâncias do grupo CMT.	47

Lista de Algoritmos

Algoritmo 1 – Algoritmo <i>Ant System</i>	17
---	----

Lista de Abreviaturas e Siglas

VRP	<i>Vehicle Routing Problem</i>
CVRP	<i>Capacitated Vehicle Routing Problem</i>
ACO	<i>Ant Colony Optimization</i>
TSP	<i>Travelling Salesman Problem</i>
PLI	Programação Linear Inteira
IBM	<i>International Business Machines</i>
GRASP	<i>Greedy Randomized Adaptive Search Procedure</i>
ANOVA	Análise de Variância

Sumário

1 – Introdução	1
1.1 Objetivos	3
1.2 Organização do trabalho	3
2 – Fundamentação Teórica	5
2.1 Pesquisa Operacional	5
2.2 Problema de Roteamento de Veículos	6
2.2.1 Problema de Roteamento de Veículos Capacitados	8
2.3 Métodos de Solução	9
2.3.1 Algoritmos Exatos	9
2.3.2 Heurísticas	11
3 – Métodos empregados	15
3.1 Otimização por Colônia de Formigas	15
3.2 Programação Linear Inteira	19
3.2.1 CPLEX	21
4 – O Modelo Matemático	22
5 – Metodologia	26
5.1 A Heurística ACO	26
5.2 CPLEX	28
5.3 O Algoritmo Híbrido	28
6 – Análise e Discussão dos Resultados	32
6.1 Testes Computacionais	32
6.2 Resultados	42
6.3 Análise dos Resultados	48
6.3.1 Instâncias do Grupo E	50
6.3.2 Instâncias do Grupo A	50
6.3.3 Instâncias do Grupo P	52
6.3.4 Instâncias do Grupo B	52
6.3.5 Instâncias do Grupo CMT	54
7 – Conclusão	56
7.1 Trabalhos Publicados	57
7.2 Trabalhos Futuros	57

Apêndices	58
APÊNDICE A – O Processo de Híbridização	59
Referências	65

Capítulo 1

Introdução

Atualmente, com o crescimento das operações globalizadas, a sobrevivência das empresas está condicionada à sua adaptação a um mercado cada vez mais exigente. Com isso, faz-se crucial a modernização e inovação no que diz respeito aos seus processos produtivos e o portfólio empresarial, além da constante melhoria na gestão dos custos e das cadeias de suprimentos [Correa e Neto (2006)].

Guimarães (2011) afirma que a gestão da cadeia de abastecimento - *Supply Chain Management* – é atualmente um dos mais importantes critérios de sucesso da organização, englobando desde as funções de planejamento e controle de fornecimento, até o gerenciamento do fluxo dos produtos, serviços e informações. Dentro deste tema podemos destacar as operações logísticas, as quais são responsáveis pela aquisição, movimentação, armazenagem e entrega de produtos [Neto e Júnior (2002)].

Nos últimos anos o Problema de Roteamento de Veículos (*Vehicle Routing Problem* - VRP) vem sendo estudado com mais atenção uma vez que ele está diretamente ligado à lucratividade e satisfação dos clientes das empresas, despertando então, um enorme interesse em realizá-lo da forma mais eficiente possível. Como consequência existe um grande empenho nesta área para que se desenvolvam algoritmos cada vez mais eficientes.

Neste trabalho é abordado o Problema de Roteamento de Veículos Capacitados (*Capacitated Vehicle Routing Problem* - CVRP) [Junqueira (2013)], uma versão clássica do VRP. Ele compreende a obtenção de um conjunto de rotas, que devem ser percorridas por uma frota de veículos homogêneos, atendendo assim as demandas de um conjunto de clientes. Seu objetivo é a minimização do custo total das rotas, sabendo-se que:

- As rotas devem iniciar e terminar no depósito central;
- Cada cliente somente pode ser atendido uma única vez, e por apenas um único veículo;
- A capacidade do veículo deve ser respeitada.

Uma característica deste problema é que o mesmo possui natureza NP-difícil, sendo comumente resolvidos através da programação linear e heurísticas.

A técnica de Programação Linear (PL) tem como característica a busca pela solução ótima do problema, permitindo afirmar que a solução retornada é o ótimo global [Chaves (2003)]. Como desvantagem, a busca por solução em instâncias com espaço de busca muito grande torna-se computacionalmente inviável, pois gastaria um tempo computacional proibitivo.

Segundo Lima, Santos e Araújo (2015), heurísticas são técnicas que trabalham para encontrar soluções subótimas com um custo operacional aceitável. Uma característica destas técnicas é que as mesmas não garantem que o resultado encontrado seja o ótimo, além de não poder quantificar o quão próxima esta solução está da solução ótima.

Na literatura, métodos exatos foram utilizados pelos autores Christofides, Mingozzi e Toth (1981) e Fukasawa et al. (2006) em seus estudos significativos e importantes para resolução do CVRP. Dentre os métodos heurísticos se destacam os trabalhos de Clarke e Wright (1964) e Gillett e Miller (1974). Os autores Lima, Santos e Araújo (2015) e Oliveira e Delgado (2015) apresentam em seus trabalhos as abordagens mais recentes acerca da resolução do CVRP.

A proposta deste trabalho é desenvolver um algoritmo híbrido que combine a natureza ágil da heurística de Otimização por Colônia de Formigas (*Ant Colony Optimization* - ACO) com a robustez dos métodos de solução por Programação Linear. O primeiro irá produzir soluções parciais de forma ágil, para assim, alimentar o segundo, que ficará responsável pela melhoria da solução enviada pela ACO, guiando-o para o mais próximo possível da solução ótima do problema.

O funcionamento do algoritmo híbrido terá como base o tipo de comunicação empregada, podendo esta ser classificada como unidirecional ou bilateral. Na comunicação unidirecional a heurística constrói uma solução e envia estes dados para o CPLEX, o qual é encarregado de tentar melhorá-la e encerrar a execução da abordagem proposta. A comunicação bilateral trabalha com a ACO enviando suas informações para o CPLEX que, após determinado tempo, devolve a solução por ele encontrada para a Colônia de Formigas, reiniciando assim o ciclo da hibridização.

O tipo de comunicação unidirecional, na qual a ACO alimenta o CPLEX, é caracterizada pela ausência do processo de realimentação. Na comunicação bilateral, a ACO e o CPLEX se comunicam em ciclos de tempo pré-definidos, com a heurística enviando seus dados ao CPLEX, e este produzindo informações que serão utilizadas na realimentação do primeiro método.

Pretende-se construir um algoritmo híbrido que resolva o Problema de Roteamento de

Veículos Capacitados, o qual produza soluções de qualidade superior às produzidas pelos métodos individualmente. Espera-se que a abordagem proposta neste trabalho encontre as mesmas soluções da literatura quando estas se tratarem do ótimo global, e quando tal característica não existir, que o híbrido consiga melhorar a qualidade da solução disponibilizada na literatura.

1.1 Objetivos

O objetivo geral deste trabalho é desenvolver e avaliar o impacto da hibridização da heurística Colônia de Formigas com Programação Linear, no contexto da solução do Problema de Roteamento de Veículos Capacitados.

Os objetivos específicos são:

- Implementar um modelo matemático do Problema de Roteamento de Veículos Capacitados via programação linear inteira.
- Aplicar a heurística Colônia de Formigas ao CVRP.
- Estudar e implementar diferentes técnicas de hibridização.
- Apresentar os resultados obtidos e compará-los à literatura.

Para resolver o modelo matemático via Programação Linear Inteira foi utilizado o software CPLEX, enquanto que a heurística ACO foi implementada através da linguagem de programação Java.

A comunicação do algoritmo híbrido, o qual compreende a união destes dois métodos, é realizada com base em arquivos de integração gerados com base nas características de cada método, de forma a garantir a correta interpretação dos mesmos.

Os resultados foram obtidos para essas três estratégias estudadas, sendo posteriormente comparados entre si e as soluções presentes na literatura.

1.2 Organização do trabalho

No capítulo 2, abordaremos os fundamentos teóricos acerca do Problema de Roteamento de Veículos Capacitados, os métodos de solução existentes na literatura, bem como os que já foram utilizados em sua resolução.

No capítulo 3, são apresentados os métodos utilizados neste trabalho como base para o algoritmo híbrido.

O capítulo 4 exibe-se o modelo matemático adotado para resolver o CVRP.

O capítulo 5 apresenta as características individuais de cada método e todos os detalhes acerca dos tipos de hibridização abordadas neste trabalho, assim como o processo de comunicação do algoritmo híbrido.

No capítulo 6, são feitos alguns testes computacionais para calibração do híbrido, seguidos da análise dos resultados produzidos.

Por fim, no capítulo 7 são abordadas as conclusões do trabalho.

Capítulo 2

Fundamentação Teórica

Neste capítulo serão apresentados conceitos sobre redes de transportes, roteirização, e o foco do trabalho que trata do Problema de Roteamento de Veículos Capacitados. Serão elucidadas suas características, o seu objetivo e sua representação. Também será explicado os métodos disponíveis na literatura para elaboração e desenvolvimento das soluções, bem como suas peculiaridades técnicas, e como estas influenciam no resultado final.

2.1 Pesquisa Operacional

Na Segunda Guerra Mundial, pesquisas massivas eram realizadas em busca de soluções para problemas de cunho militar, datando desta época as primeiras aparições do termo "Pesquisa Operacional". O crescimento desta área ao longo dos anos se deu pelo avanço nas importantes técnicas para formulação de problemas e a popularização dos computadores [Ferreira e Bachega (2011)].

Moreira ((2007) apud FERREIRA; BACHEGA, (2011)) afirma que a pesquisa operacional lida com problemas preocupados com a forma de conduzir e gerir específicas operações de uma companhia, estando intimamente ligada ao processo de tomada de decisão. Áreas como indústria, transportes, finanças, saúde, telecomunicações, entre outros, são exemplos nos quais a pesquisa operacional pode ser aplicada.

Ao trabalhar com este tema, deve-se considerar quatro etapas básicas, que se iniciam na formulação do problema, passa pela construção e execução de um modelo definido, e finda na aplicação dos resultados obtidos (Figura 1). Em pesquisa operacional, os problemas são representados por meio de modelos matemáticos, construídos, em geral, por meio de equações ou expressões matemáticas.

Segundo Moreira ((2007) apud FERREIRA; BACHEGA, (2011)), os principais elementos de um modelo matemático são:

Fonte: Adaptado de [Goldberg e Luna \(2000\)](#)

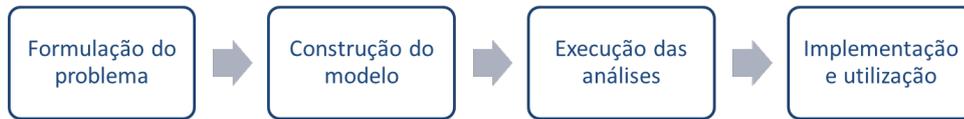


Figura 1 – Fluxo de resolução de um problema em Pesquisa Operacional

- Variáveis de decisão: são associadas as n decisões quantificáveis de um problema.
- Parâmetros: correspondem a valores fixos do problema.
- Restrições: representam as limitações dos recursos do problema, sendo expressas matematicamente por meio de equações e/ou inequações.
- Função objetivo: diz respeito a uma combinação linear das variáveis de decisão.

Ao construir um modelo matemático deve-se pensar na técnica de resolução a ser utilizada, dado que, se necessário, pequenas adaptações devem ser feitas para simplificar a obtenção da solução.

2.2 Problema de Roteamento de Veículos

Segundo [Martel e Vieira \(\(2008\) apud MELQUIADES, \(2011\)\)](#) a roteirização é um conjunto organizado de meios cujo objetivo corresponde ao atendimento de demandas presentes em arcos ou nós de qualquer rede de transporte. Esta corresponde ao conjunto de todas as vias de transporte de pessoas e de mercadorias que se interligam em uma determinada região por meio de infraestruturas [[Muller et al. \(2013\)](#)].

Esta rede de transporte pode ser representada por meio de uma estrutura computacional denominada grafo, a qual é composta por um conjunto de vértices e arestas. As arestas são responsáveis por interligar um par de vértices distintos e representam as vias de transportes, enquanto os vértices simbolizam as demandas. Nas arestas também ocorrerão a representação do(s) sentido(s) das vias, identificando um grafo como direcionado ou não direcionado. Um grafo direcionado é análogo a uma via de sentido único, enquanto o não direcionado pode ser associado a uma via de mão dupla.

Os problemas que envolvem roteirização se concentram no atendimento das demandas, e em como o serviço de transporte deve ser efetivo e flexível. Um planejamento correto implica efetivamente na qualidade e eficiência das entregas, principal finalidade das empresas de transporte atuais.

Dentro desse contexto surge o Problema de Roteamento de Veículos (*Vehicle Routing Problem* - VRP), proposto por [Dantzig e Ramser \(1959\)](#), de natureza combinatória e de grande dificuldade de resolução. Trabalha-se na alocação de uma quantidade de veículos para atender uma determinada quantidade de clientes distribuídos geograficamente, com foco na minimização de um custo de transporte associado [[Bittencourt \(2010\)](#)].

Este problema varia quanto à sua dimensão e de acordo com a quantidade de variáveis e restrições que o mesmo vai abordar na sua formulação, sendo que muitos podem ser considerados extremamente difíceis de tratar devido à complexidade que possuem. Esta dificuldade está diretamente ligada à característica combinatória, dificultando o desenvolvimento de algoritmos que sejam eficientes na busca por soluções.

A literatura classifica este tipo de problema como NP-difícil, uma vez que o esforço empregado computacionalmente para sua resolução cresce em formato exponencial de acordo com a complexidade do problema e o tamanho do espaço de busca.

As estratégias de resolução existentes compreendem a utilização de técnicas cujos resultados se aproximam da solução ótima, ou outras que resolvam o problema efetivamente. A [Figura 2](#) identifica as estratégias existentes dentro da literatura.

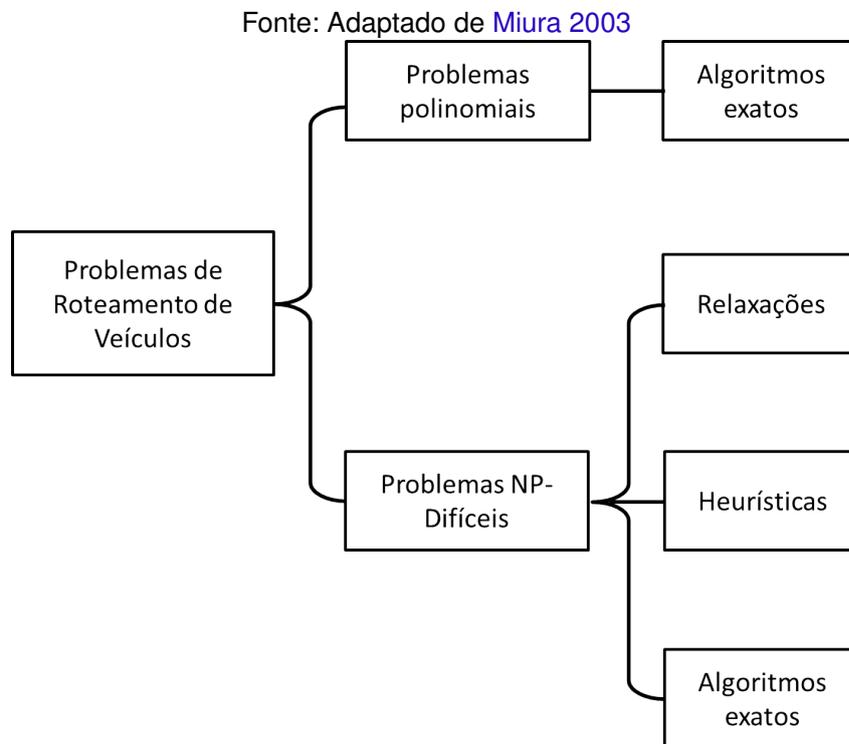


Figura 2 – Estratégias de resolução do VRP

Como visto na [Figura 2](#), em problemas NP-difíceis podem ser empregadas técnicas de relaxações, heurísticas e algoritmos exatos. As heurísticas trabalham de forma simples e eficiente, na busca por soluções próximas à ótima. Os algoritmos exatos são menos ágeis e

mais custosos em relação às heurísticas. Técnicas de relaxação são comumente utilizadas junto aos métodos exatos, promovendo a eliminação de algumas variáveis e restrições do problema [Miura 2003].

Uma versão clássica deste problema é o Problema de Roteamento de Veículos Capacitados (*Capacitated Vehicle Routing Problem* - CVRP), o qual trabalha na busca de um conjunto de rotas de custo mínimo, sendo que estas serão percorridas por um conjunto de veículos homogêneos que ficarão encarregados de cumprir com às demandas dos clientes.

2.2.1 Problema de Roteamento de Veículos Capacitados

Este problema foi proposto por Dantzig e Ramser (1959), herdando do VRP a sua natureza combinatória, e, portanto, também sendo classificado como NP- difícil. Neste problema tem-se um conjunto de veículos idênticos, com capacidades idênticas e conhecidas, disponíveis para trabalhar na entrega e/ou coleta de pedidos. Estes pedidos pertencem a clientes distribuídos geograficamente, cuja localização e demanda são previamente conhecidas, bem como as distâncias associadas às viagens entre eles [Junqueira (2013)].

Também é conhecida a distância entre o depósito central e cada um dos clientes. Neste depósito, também tratado na literatura como armazém, estará inicialmente concentrada toda a demanda a ser distribuída na rota. Uma consideração a ser feita é que todo veículo deverá partir do depósito, atender as demandas e, ao final da rota, retornar a este ponto de partida.

O objetivo deste problema é encontrar um conjunto de rotas de custo mínimo, que na maioria dos casos, conforme afirmado por Dilson (2010), corresponde à distância total a ser percorrida. Contudo, algumas restrições devem ser respeitadas como parte da resolução do problema, sendo elas:

- Cada cliente só pode ser visitado uma única vez;
- A demanda de cada cliente só pode ser efetuada integralmente, de uma só vez, e por apenas um veículo;
- A(s) rota(s) deve(m) ter início e fim no depósito;
- Deve-se respeitar a capacidade do veículo;
- Todos os clientes devem ser atendidos.

Assim como no VRP, podemos representar o CVRP através de um grafo não-direcionado $G=(V, A)$, no qual V é um conjunto de vértices formado pelo depósito, vértice inicial (v_0), e n clientes - (v_1, v_2, \dots, v_n), e A é o conjunto de arestas que unem os pares de vértices, na qual cada aresta (i, j) possui um custo associado $(c_{i,j})$.

A partir da análise dos vértices, do conjunto de arestas disponíveis, e dos dados acerca das

demandas e distâncias, será possível encontrar uma solução para o CVRP, conforme ilustra a Figura 3.

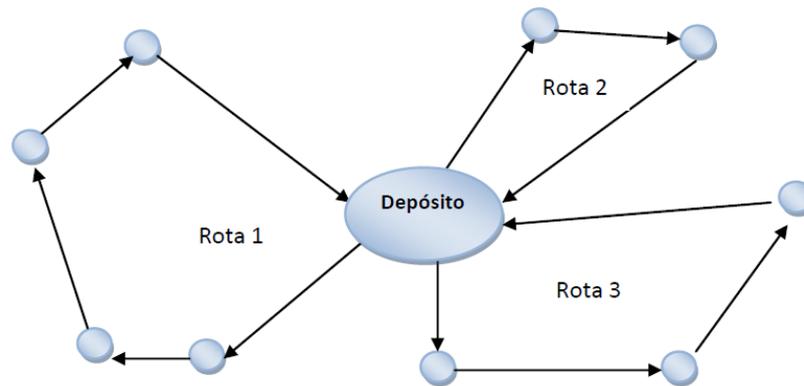


Figura 3 – Representação possível para o problema de roteamento de veículos.

2.3 Métodos de Solução

Na literatura existem duas metodologias diferentes para trabalhar com problemas de pesquisa operacional, sendo duas delas: métodos exatos e heurísticas. O primeiro tem como característica principal a garantia da otimalidade, enquanto o segundo é atrativo pelo esforço computacional empregado.

2.3.1 Algoritmos Exatos

Segundo [Estevam \(2003\)](#), algoritmos exatos, denominados enumerativos, sempre levam ao ótimo global, e essencialmente percorrem o espaço de soluções em busca daquela de melhor valor, podendo ser distinguidos pela forma como o espaço é varrido.

Um deles corresponde à enumeração explícita, caracterizada por percorrer todo o espaço de soluções em busca da que possua melhor métrica. Uma desvantagem deste método é que ele pode se tornar inviável para grandes instâncias, uma vez que o espaço de soluções de um problema combinatório cresce exponencialmente, tornando a tarefa de pesquisa árdua e computacionalmente custosa.

A outra forma é conhecida como enumeração implícita, cuja responsabilidade está em examinar apenas parte do espaço, utilizando limites obtidos por relaxações que afunilam o número de soluções disponíveis. Quando se diz que um problema é relaxado, deve-se interpretar que apenas um subconjunto das restrições é considerado em uma análise parcial.

Em [Reis \(2004\)](#) são citados alguns algoritmos exatos, dentre eles a Relaxação Lagrangeana e o *Branch and Bound*.

No caso da Relaxação Lagrangeana, algumas restrições são removidas da formulação inicial, e as desigualdades geradas por tal ação são embutidas na função objetivo por meio de um vetor de multiplicadores. A ideia principal ao utilizar esse vetor é penalizar o problema quando as restrições que foram removidas são violadas, transformando-o em uma proposta mais simples de resolver.

O algoritmo *Branch and Bound* é um método de enumeração implícita. Para evitar que todo o espaço de busca seja varrido, este método faz uso de estimativas no valor da solução ótima do problema, utilizando para isso rotinas que calculam os seus limites inferiores e superiores.

Enquanto os valores destes limites forem diferentes, ocorrerá a ramificação (ou *branching*), ou seja, o problema será particionado em outros subproblemas, que unidos equivalem ao original. Cada um é resolvido separadamente e, desta forma, a solução do melhor subproblema será a solução do problema inicial.

Segundo [Reis \(2004\)](#), o funcionamento do algoritmo *Branch and Bound* pode ser associado à busca em árvore, a qual tem como característica o particionamento de determinado problema de acordo com um dado critério. Durante a execução, os limites para os valores das soluções de cada subproblema são determinados e, a cada iteração, é selecionado o que forneceu o menor limite inferior até o momento.

Na literatura dos métodos exatos podemos citar os trabalhos de [Christofides, Mingozzi e Toth \(1981\)](#), [Fukasawa et al. \(2006\)](#) e [Santos \(2014\)](#).

Em seu trabalho, [Christofides, Mingozzi e Toth \(1981\)](#) propuseram um algoritmo exato que utiliza um limite Langrangeano obtido a partir de soluções do problema das q-rotas mínimas. Estas podem ser definidas como um conjunto de rotas com início e fim no depósito, formadas por uma sequência de clientes, as quais foram construídas a partir do acúmulo de suas demandas.

Um algoritmo *branch-and-cut-and-price* foi desenvolvido por [Fukasawa et al. \(2006\)](#) e aplicado a instâncias com mais de 135 clientes, obtendo resultados ótimos para todas estas.

Programação Linear Inteira foi o método escolhido por [Santos \(2014\)](#), tendo como base o modelo matemático desenvolvido por [[Toth e Vigo \(2002\)](#)]. Para resolver o problema, o autor utilizou o software GUSEK combinado a linguagem AMPL, sabendo que os dados que alimentaram seu modelo foram definidos a partir de um cenário hipotético por ele criado, fundamentando-se nos municípios da região de Formiga - MG.

2.3.2 Heurísticas

Algoritmos heurísticos buscam resolver problemas baseando-se em regras empíricas e não garantem a otimalidade da solução mas são executados em tempo polinomial, diferentemente dos algoritmos exatos [Estevam (2003)]. Os métodos heurísticos possuem tempos de execução inferiores aos dos métodos exatos, oferecendo soluções eficientes para os problemas abordados.

Na literatura, um dos métodos heurísticos usado para resolver problemas de roteamento corresponde ao *Simulated Annealing*. Proposto por Kirkpatrick, Gelatt e Vecch (1983), diz respeito a um método de busca local que simula processo de recozimento de metais, no qual a temperatura de um sólido é elevada até o seu ponto de fusão, seguido de um gradual e lento resfriamento. Durante esse recozimento o material passará por vários estados possíveis, e se o resfriamento for suficientemente lento, ao atingir o equilíbrio térmico, terá como resultado uma estrutura livre de imperfeições [Araújo (2001)].

O processo de busca deste algoritmo combina estratégias de caminhos aleatórios e melhoramento iterativo. Inicialmente a estratégia é permitir que movimentos ruins ocorram, o que resulta em uma melhora pequena da solução e, conseqüentemente, permite que o espaço de busca seja explorado. A medida que a temperatura diminui, o comportamento do método se altera, e o resultado final converge a um mínimo local.

Outra heurística conhecida como Busca Dispersa (Scattered Search) é uma heurística evolutiva, baseada em populações, proposta por Glover (1977), a qual constrói soluções pela combinação de outras. Este processo tem como suporte um conjunto de referência (denominado *Refset*), composto de boas soluções obtidas ao longo da execução do algoritmo. Sosa, Galvão e Gandelman (2007) explicam que, neste método, o conceito de "boa solução" vai além do valor numérico associado à ela, incluindo um conceito de "diversidade" ao *Refset*. Ao combinar as soluções do conjunto de referência, cria-se novas soluções no espaço de busca. A intenção desta operação é capturar informações que não estavam presentes nas soluções originais.

Outra heurística utilizada na literatura dos problemas de roteamento é a Otimização por Colônia de Formigas. Este método é inspirado no comportamento das formigas reais quando estão em busca de comida, mais especificamente no modo como estas encontram o menor caminho entre a fonte de alimentos e o formigueiro. Estes agentes biológicos utilizam mecanismos de adaptação, cooperação e paralelismo no processo de busca pelo alimento. Segundo Dorigo e Stützle (2004), o alto nível de organização destes insetos permitem que eles realizem tarefas complexas que, em algumas situações, não poderiam ser executadas por uma única formiga. A coordenação da atividade de localização de alimento giram em torno de uma substância química denominada feromônio, que em alta concentração

aumenta a chance de outras formigas seguirem o mesmo caminho.

Na literatura do CVRP, são vários os trabalhos desenvolvidos utilizando métodos heurísticos. O primeiro deles foi [Clarke e Wright \(1964\)](#), cuja publicação trabalha com a noção de economias para resolver o problema de roteamento. Este método, conhecido como heurística de Clarke e Wright, é considerado o pioneiro e se baseia no custo da combinação, ou união, de duas sub-rotas existentes; possuindo natureza gulosa e iterativa.

Outro trabalho bastante conhecido é o de [Gillett e Miller \(1974\)](#), que utiliza a heurística de varreduras, também conhecida como *Sweep Algorithm*. A técnica desenvolvida funciona em duas etapas distintas para a construção de uma solução. Na primeira, os pontos de demanda são agrupados em *clusters*, tendo como base um critério de proximidade. Estes *clusters* serão solucionados, de forma independente, na segunda etapa do algoritmo. A solução encontrada ainda passará por um procedimento de busca local, com a intenção de refinar a qualidade do resultado obtido.

O trabalho de [Galvão et al. \(1997\)](#) utiliza a metaheurística *Simulated Annealing* [[Kirkpatrick, Gelatt e Vecch \(1983\)](#)], combinada ao procedimento adaptativo auxiliar Busca Tabu [[Glover \(1986\)](#)] para solucionar o CVRP. Outros autores também utilizaram o *Simulated Annealing* em suas pesquisas, sendo eles [Osman \(1993\)](#), [Zeng, Ong e Ong \(2005\)](#) e [Chen, Golden e Wasil \(2007\)](#).

Outra heurística estudada na resolução do CVRP foi a Busca Dispersa, abordada nos trabalhos de [Bittencourt et al. \(2012\)](#) e [Sosa, Galvão e Gandelman \(2007\)](#). Nos dois trabalhos são utilizados a heurística de [Gillett e Miller \(1974\)](#) na geração das soluções iniciais, contudo utilizam de outras técnicas ao decorrer do algoritmo. [Bittencourt et al. \(2012\)](#) incorporam mais dois procedimentos, sendo um intra-rota e o outro inter-rotas. O primeiro utiliza a heurística 2-opt [[Croes \(1958\)](#)], responsável por retirar duas arestas da sub-rota e reinseri-las em posições invertidas, enquanto o segundo passo trabalha com a aplicação em sequência das heurísticas de realocação, intercâmbio e cruzamento. Já [Sosa, Galvão e Gandelman \(2007\)](#) incorporam a heurística de [Beasley \(1983\)](#), cuja estratégia se baseia primeiramente no roteamento e, posteriormente, no agrupamento das soluções.

Os autores [Zanneti, Marocco e Junior \(2014\)](#) utilizaram o método iterativo GRASP [[Resende e Ochi \(2009\)](#)], o qual é composto por uma fase construtiva, uma de busca local, e a fase de refinamento. Na fase construtiva foram utilizadas as estratégias denominadas inserção mais próxima (ADD) e inserção mais próxima randomizada GRASP (ADDR). Na estratégia ADD, a solução inicial é criada com base no menor custo de conexão entre vértices, ou seja, durante a construção da rota, o vértice adjacente que proporcionar o menor aumento ao custo será escolhido para ser inserido. Na ADDR buscou-se inserir a característica aleatória do GRASP e, para isso, o autor elaborou uma lista de vértices candidatos ordenados por

custo, elegendo os três mais econômicos, dos quais um era aleatoriamente selecionado para fazer parte da solução parcial. Os critérios de busca local implementados foram o com parada prematura, no qual a construção da rota é finalizada quando a capacidade restante do veículo não absorve a demanda do próximo cliente, e o de varredura, o qual apenas finaliza a rota quando a demanda de todos os demais vértices ainda não atendidos são analisadas. Na parte de refinamento da solução, o autor utiliza o método DROP, o qual irá trocar sistematicamente vértices de rotas diferentes. Esse processo se dá ao retirar um vértice de uma rota buscando inseri-lo em outra que absorve sua demanda.

A Otimização por Colônia de Formigas foi o método escolhido por [Tan et al. \(2012\)](#), que trabalhou com a heurística na sua forma pura e mais duas variantes, sendo elas ACO-S e ACO-T. Estas duas últimas possuem caráter elitista, pois fornecem um reforço adicional às arestas que pertencem ao melhor percurso encontrado até o momento. O que as diferencia é que a primeira utiliza a técnica *subtour reverse*, a qual trabalha realizando a troca de um nó com seu subsequente, enquanto a segunda utiliza a heurística 2-opt.

Outros trabalhos utilizaram ACO na resolução do CVRP foram os de [Bell e McMullen \(2004\)](#), [Mazzeo e Loiseau \(2004\)](#), [Reimann, Doerner e Hartl \(2004\)](#), [Dorner, Hartl e Lucka \(2005\)](#), [Li e Tian \(2006\)](#) e [Néia et al. \(2013\)](#).

Os autores [Gasparin \(2014\)](#), [Lima, Santos e Araújo \(2015\)](#) e [Oliveira e Delgado \(2015\)](#) optaram por combinar diferentes métodos heurísticos para resolver o problema estudado.

Em seu trabalho, [Gasparin \(2014\)](#) resolve o problema de roteamento de veículos capacitados em vários estágios. O primeiro consiste em designar as *p-medianas* através do método de [Teitz e Bart \(1968\)](#) para, em seguida, agrupar os clientes com base na localização das medianas, utilizando para isso o algoritmo de [Gillett e Johnson \(1976\)](#). Na fase de construção da solução inicial é usado o algoritmo de inserção mais distante, e para a etapa de refinamento das soluções, utiliza-se o algoritmo 3-opt [[Lin \(1965\)](#)]. Este possui a mesma dinâmica do 2-opt, com a única diferença do números de vértices a serem invertidos.

Os autores [Lima, Santos e Araújo \(2015\)](#) combinam um Algoritmo Genético (AG) à heurística de [Gillett e Miller \(1974\)](#) e à heurística de Descida de Encosta (*Hill-climbing*) [[Pearl \(1984\)](#)] para resolver o Problema de Roteamento de Veículos Capacitados. O primeiro é um método de busca probabilístico que se baseia no processo evolutivo, enquanto a Descida de Encosta é uma técnica de busca local em profundidade. As soluções iniciais do algoritmo genético são geradas pela heurística de varreduras, as quais passam por um processo de refinamento através do método de descida de encosta. Só após esta etapa os operadores de seleção, mutação, cruzamento e elitismo, característicos do AG, são aplicados ao problema.

Um algoritmo baseado no método de Monte Carlo juntamente da heurística de [Clarke e](#)

Wright (1964), denominando *Monte Carlo Savings*, foi proposto por Oliveira e Delgado (2015). A ideia dos autores foi utilizar o método de Monte Carlo na geração da lista de *savings* da heurísticas, admitindo à estratégia um certo grau de variabilidade, e permitindo que soluções diversificadas sejam encontradas.

Neste trabalho será abordado um algoritmo que combina a heurística de Otimização por Colônia de Formigas à técnica de Programação Linear Inteira, baseando em um modelo matemático mais recente na literatura.

Capítulo 3

Métodos empregados

Neste capítulo são abordados os métodos utilizados nesse trabalho, bem como o software utilizado para trabalhar com a programação linear.

3.1 Otimização por Colônia de Formigas

A otimização por colônia de formigas (*Ant Colony Optimization* - ACO) é uma metaheurística baseada no comportamento de formigas, que são insetos sociais capazes de se organizarem para executar tarefas de grande complexidade visando garantir a sobrevivência do formigueiro [Goliatt, Angelo e Barbosa (2013)].

A inspiração para tal método partiu da experiência de Goss et al. (1989) feita com formigas reais em busca de alimentos. Nesta experiência a colônia (N) destes insetos e a fonte de alimentos (A) dos mesmos se conectavam através de uma ponte dupla, cujos caminhos apresentavam distâncias diferentes (Figura 4).

Goss et al. (1989) observaram que inicialmente uma formiga saía do ninho para explorar o terreno em busca de alimento sendo que, a escolha sobre qual trajeto percorrer nesta etapa era arbitrária. Ao encontrar o alimento, o inseto efetuava o trajeto contrário até o ninho, realizando o depósito de uma substância química denominada feromônio, e, ao chegar ao local, outras formigas eram chamadas para partir ao encontro do alimento. Os autores também notaram que com o passar do tempo as formigas estavam percorrendo o caminho mais curto, além de que, corriqueiramente, alguns insetos partiam sozinhos para outro rumo.

Essa disposição das formigas de escolherem o caminho mais curto pode ser explicada pelas diferentes taxas de feromônio depositadas nas rotas. Os autores definem o feromônio como uma substância química de reconhecimento, usada para comunicação entre as formigas. À medida que uma formiga realiza seu trajeto, uma quantidade de feromônio é depositada

Fonte: Adaptado de [Goss et al. \(1989\)](#)

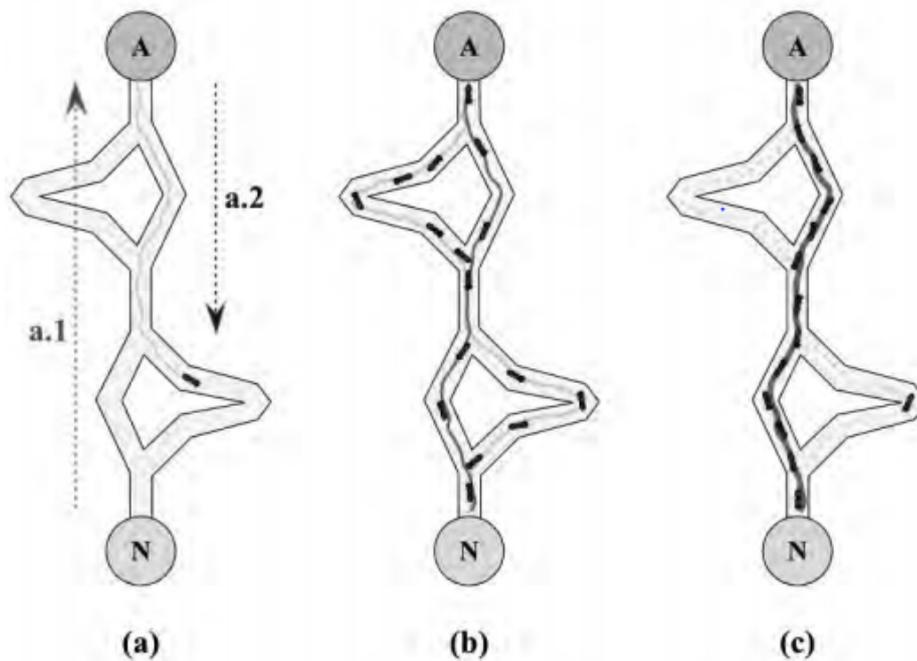


Figura 4 – Experiência realizada com formigas reais.

na trilha. Portanto, quanto mais formigas passarem por uma rota, maior será a taxa de feromônio existente naquele trajeto, e conseqüentemente, maior será o incentivo para que esta rota seja utilizada.

Assim como ocorre o processo de depósito de feromônio, também existe o processo de evaporação do mesmo, no qual a substância depositada pela formiga perde gradualmente a sua atratividade natural. Dessa forma, quando uma formiga opta por uma rota com custo menor, o feromônio nela depositado irá evaporar mais lentamente do que o que está presente no trajeto de maior comprimento, e, devido este comportamento, maior será a chance das outras formigas acabarem utilizando o caminho mais curto, uma vez que o mesmo se apresentará mais atrativo que o outro.

A partir da experiência de [Goss et al. \(1989\)](#) surgiu o conceito das formigas artificiais, cujo comportamento se baseia no das formigas naturais. Esses agentes computacionais constroem soluções baseando-se nos seguintes dados, conforme dito por [Neto e Filho \(2013\)](#):

- A trilha de feromônio artificial (esta trilha mudará de acordo com a execução do programa);
- A qualidade da solução encontrada até o momento, a qual será a responsável por indicar a quantidade de feromônio a ser depositado no caminho.

As formigas artificiais constituem a heurística construtiva *Ant System* (AS) proposta por [Dorigo et al. \(1991\)](#). Neste algoritmo as formigas construirão soluções de forma probabilística, baseando-se na trilha de feromônio artificial criada, assim como de acordo com a informação do problema a ser resolvido.

Da mesma forma que ocorre no processo natural, a formiga artificial deixará certa quantidade de feromônio no trajeto que percorrer. Contudo, no ambiente computacional, a decisão sobre qual rota escolher se baseará em uma função probabilística que envolve a distância do caminho e a quantidade de feromônio existente nele.

Esta heurística foi aplicada pela primeira vez na resolução do Problema do Caixeiro Viajante (TSP), e pode ser traduzida conforme o pseudocódigo abaixo, adaptado de [Ferreira, Monteiro e Teixeira \(2008\)](#). O principal objetivo consiste em minimizar o percurso da viagem, sabendo que todas as cidades devem ser visitadas por apenas um único vendedor [[Neto e Filho \(2013\)](#)].

Algoritmo 1: Algoritmo *Ant System*

Input: *numeroMaximoIteraes*, *quantidadeFormigas*

Output: S^*

$L^* \leftarrow \infty$

```

for  $t \leftarrow 1$  a numeroMaximoIteracoes do
  for  $k \leftarrow 1$  a quantidadeFormigas do
    while  $k$  não construir a rota  $S_k$  do
      | Selecione a próxima cidade conforme a regra  $p_{ij}^k(t)$ 
    end
    Calcule a distância  $L_k(t)$  do caminho descoberto pela formiga  $k$ 
    if  $L_k(t) < L^*$  then
      |  $S^* \leftarrow S_k(t)$ 
      |  $L^* \leftarrow L_k(t)$ 
    end
  end
  Atualize os feromônios
end
Retornar  $S^*$ 

```

Imitando o comportamento das formigas reais, estes agentes computacionais são posicionados em um grafo, em que os nós representam as cidades, e as arestas correspondem aos caminhos, nos quais se movimentarão até que determinada condição seja satisfeita. Quando isto acontece, a solução final do problema diz respeito ao caminho percorrido pelos insetos.

Inicialmente as formigas estarão concentradas no depósito central e todos os caminhos existentes serão inicializados com a mesma taxa de feromônio ($\tau_{i,j} > 0$).

Para prosseguir com sua caminhada, a formiga tomará como base de suas decisões o

resultado da regra probabilística mostrada na Equação 1, tendo em mente que este processo ocorrerá até que todas as formigas tenham visitado todas as cidades através de um caminho válido, ou seja, completamente interligado.

$$p_{ij}^k = \left\{ \frac{[\tau_{ij}(t)]^\alpha \times [\eta_{i,j}(t)]^\beta}{\sum_{l \in N_i^k} [\tau_{il}(t)]^\alpha \times [\eta_{i,l}(t)]^\beta} \right\}. \quad (1)$$

Segundo esta regra, denominada regra de transição, a probabilidade ρ de uma formiga k , que está situada na cidade i , escolher ir para a cidade j , é dada pelos seguintes parâmetros:

- $\tau_{ij}(t)$: feromônio presente na aresta (i, j) no instante t de visitação.
- α e β : variáveis que determinam a influência do feromônio τ_{ij} e da informação heurística η_{ij} , respectivamente.
- η_{ij} : Representa a atratividade da formiga visitar a cidade j depois de visitar a cidade i . Este parâmetro é determinado pela relação $1/d_{ij}$, no qual d_{ij} corresponde a distância entre a cidade i e j .
- N_i^k : esta variável diz respeito à vizinhança factível de uma formiga k , ou seja, todas as cidades adjacentes ainda não visitadas pela mesma.

O próximo passo diz respeito ao processo de atualização do feromônio nas arestas do grafo, este composto por dois eventos: o depósito e a evaporação da substância. O depósito procura reforçar positivamente a escolha pelo melhor caminho, enquanto a etapa de evaporação visa esquecer-se das decisões ruins tomadas nos passos anteriores, evitando também o crescimento indefinido do feromônio.

A quantidade $\tau_{ij}^k(t)$ da substância a ser depositada em uma aresta (i, j) , pela formiga k , é dada pela Equação 2:

$$\Delta\tau_{ij}^k(t) = \frac{Q}{L_k(t)} \quad (2)$$

na qual:

- Q corresponde ao valor total de feromônio exalado por uma formiga a cada iteração, e;
- $L_k(t)$ diz respeito ao comprimento completo da rota descoberta pela formiga k , naquele instante t .

Desta forma, o feromônio $\tau_{i,j}$ associado à aresta (i, j) será atualizado com base na equação 3:

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij}(t), \quad (3)$$

onde

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t), \quad (4)$$

- m : corresponde a quantidade total de formigas.
- ρ : trata-se da taxa de evaporação do feromônio ($0 < \rho \leq 1$).
- $\Delta\tau_{ij}(t)$: refere-se ao cálculo para depósito da substância.

Estes procedimentos serão executados até um determinado número máximo de iterações, ou até que o algoritmo atinja a estagnação, a qual ocorre quando todas as formigas seguem sempre o mesmo percurso, causando, conseqüentemente, o aumento excessivo de feromônio nas arestas de uma rota.

3.2 Programação Linear Inteira

A programação linear é uma técnica amplamente utilizada em problemas de pesquisa operacional devido à sua grande aplicabilidade, buscando solucionar problemas cujos modelos são definidos através de expressões lineares.

A principal tarefa deste método é maximizar ou minimizar a função desenvolvida para o problema, respeitando as restrições de igualdade e desigualdade existentes nele [Marins (2011)]. Esta função mede a eficiência do problema e as restrições servirão para garantir que as soluções estejam de acordo com as limitações impostas.

Os autores Ferreira e Bachega (2011) citam a segurança deste método, ressaltando a confiança na solução obtida pelo mesmo, considerando para isso, a correta elaboração do modelo.

Problemas de transportes, estoque, produção, mistura de componentes, são clássicas aplicações da programação linear, sem deixar de afirmar que a área para a utilização desta é bem mais ampla do que foi citado. Esta técnica tornou-se uma das mais eficazes dentre as existentes [Ferreira e Bachega (2011)], além de estar em constante evolução.

Um caso específico de programação linear corresponde à Programação Linear Inteira (PLI), na qual as variáveis são discretas, ou seja, assumem valores inteiros. Para este caso, classificamos o problema de Programação Linear Inteira Pura; mas quando apenas algumas variáveis assumem a condição de integralidade, dizemos que trata-se de um problema de Programação Linear Inteira Mista.

Os autores Pizzolato e Gandolpho (2013) apresentam como características fundamentais da PLI:

- Variáveis de decisão (n): x_j , sendo $j = 1, 2, \dots, n$;
- Uma função a ser otimizada;
- Restrições lineares;
- Não negatividade das variáveis de decisão, x_j .

As variáveis de decisão compõem as restrições do problema, e a função que se deseja otimizar, denominada função objetivo. Esta poderá ser construída visando maximizar ou minimizar o problema.

Tais informações são a base para a formulação do modelo de programação linear. Este possui a seguinte representação algébrica:

$$\text{Max (ou Min)} \quad f(x) = \sum_{j=1}^n c_j x_j \quad (5)$$

Sujeito a:

$$\sum_{j=1}^n a_{ij} x_j \begin{pmatrix} \leq \\ = \\ \geq \end{pmatrix} b_i \quad \forall i = 1, \dots, n \quad (6)$$

$$x_j \geq 0 \quad \forall j = 1, \dots, n \quad (7)$$

A Equação (5) corresponde a função objetivo, na qual x_j diz respeito à variável de decisão do problema, e c_j ao custo associado a esta.

A Equação (6) exhibe as restrições do problema, que podem ser de inferioridade, igualdade ou superioridade. Nesta também são apresentadas as variáveis a_{ij} , que corresponde a uma determinada quantidade especificada para a restrição, e b_i , que representa uma exigência a ser cumprida. Por último, a Equação (7) exhibe as condições de não-negatividade do modelo.

Existem no mercado diversos softwares que trabalham com programação linear, tornando-a viável e eficiente para modelos complexos, dentre os quais se destacam o LINDO - *Linear Discrete Optimizer*¹, o CPLEX² e o Gurobi³.

¹Disponível em: <www.lindo.com>

²Disponível em: <www-01.ibm.com>

³Disponível em: <www.gurobi.com>

3.2.1 CPLEX

O CPLEX é um pacote de otimização desenvolvido por Robert Bixby da empresa CPLEX Optimization, que em 1997 foi adquirida pela ILOG, esta adquirida anos mais tarde pela IBM.

Este software é amplamente utilizado na resolução de problemas lineares, inteiros e não-lineares de grande complexidade. Possui interface com várias linguagens de programação, dentre as quais cita-se C++, C, Java, Matlab; e pode ser adquirido de forma gratuita desde que usado para fins acadêmicos.

O CPLEX é disponibilizado de três maneiras diferentes, com o objetivo de atender as necessidades dos usuários. São elas:

- *CPLEX Callable Library*: esta opção permite que os programadores encapsulem os otimizadores ILOG CPLEX em aplicações desenvolvidas na linguagem C, ou mesmo qualquer outra que permita chamar funções escritas em C.
- *ILOG Concert Technology*: corresponde a uma biblioteca de classes e funções em C++, Java, .NET e Matlab que permite criar modelos para problemas de otimização e utilizar algoritmo para resolvê-los. Este modo provê estruturas de dados e de controle que integram a parte da otimização com o decorrer da aplicação em tempo de execução;
- *CPLEX Interactive Optimizer*: é software executável que permite desenvolver um problema, entregando iterativamente, ou apenas no final, os resultados encontrados.

Neste trabalho é utilizado o CPLEX para resolver os modelos matemáticos envolvidos, mais precisamente a opção *Interactive Optimizer*.

Capítulo 4

O Modelo Matemático

Tomando como base as informações apresentadas pela literatura, [Junqueira \(2013\)](#) definiu matematicamente o Problema de Roteamento de Veículos Capacitados. Este modelo matemático proporciona o desenvolvimento de um algoritmo que seja capaz de fornecer uma solução exata para o problema, assumindo que exista um número limitado de veículos iguais, com determinada capacidade, que se localizam inicialmente em um depósito central.

Os veículos serão responsáveis por entregar os pedidos dos clientes que estão espacialmente dispersos sobre uma área de atendimento. A demanda destes clientes, assim como a localização e o custo de viagem entre os mesmos (e entre eles e o depósito), é previamente conhecida. O objetivo então é a geração de rotas de custos mínimos para os veículos, levando em consideração as restrições do problema.

Considerando um grafo direcionado completo $G = (N, A)$, no qual N é o conjunto de nós que correspondem ao depósito (nó 1) juntamente dos clientes (nós 2 ... n), e A o conjunto de arestas (k, l) , o custo de um veículo v transportar a demanda D do cliente k (D_k), saindo do nó k até o nó l , é dado pela variável c_{kl} .

Cada veículo v pertence a um conjunto de veículos V e possui uma determinada capacidade C_v . Este veículo terá uma rota com início no nó 1 (depósito) na posição 0 do roteiro, e fim no nó 1 na posição n .

A variável d_{kl}^{tv} será a variável de decisão do problema, que terá valor 1 se o veículo v vai diretamente do nó k para o nó l na posição t , ou 0 caso contrário, tal que $(k, l) \in A$, $(t) \in N$ e $(v) \in V$.

Pela formulação de [Junqueira \(2013\)](#), assumindo que $d_{1l}^{tv} = 0$ para todo $l \in N \setminus \{1\}$, $t \in N \setminus \{1\}$

e $v \in V$, o CVRP pode ser modelado da seguinte forma:

$$\text{Min} \sum_{(k,l) \in A} \sum_{t \in N} \sum_{v \in V} c_{kl} d_{kl}^{tv} \quad (8)$$

Sujeito a

$$\sum_{l \in N} \sum_{t \in N} \sum_{v \in V} d_{kl}^{tv} = 1 \quad k \in N \setminus \{1\} \quad (9)$$

$$\sum_{l \in N} \sum_{t \in N \setminus \{1\}} \sum_{v \in V} t d_{kl}^{tv} - \sum_{p \in N} \sum_{t \in N} \sum_{v \in V} t d_{pk}^{tv} = 1 \quad k \in N \setminus \{1\} \quad (10)$$

$$\sum_{l \in N \setminus \{1\}} d_{1l}^{1v} \leq 1 \quad v \in V \quad (11)$$

$$\sum_{l \in N} d_{kl}^{(t+1)v} - \sum_{p \in N} d_{pk}^{tv} = 0 \quad k \in N \setminus \{1\}, t \in N \setminus \{n\}, v \in V \quad (12)$$

$$\sum_{k \in N \setminus \{1\}} \sum_{l \in N} \sum_{t \in N \setminus \{1\}} D_k d_{kl}^{tv} \leq C_v \quad v \in V \quad (13)$$

$$d_{kl}^{tv} \in \{0, 1\} \quad (k, l) \in A, t \in N, v \in V \quad (14)$$

A função objetivo apresentada na Equação (8) busca minimizar o custo total das viagens realizadas; a restrição da Equação (9) garante que cada cliente será visitado apenas uma vez, e, segundo a Equação (11), por um veículo que só poderá ser usado no máximo em uma única oportunidade, conforme ilustra a Figura 5.

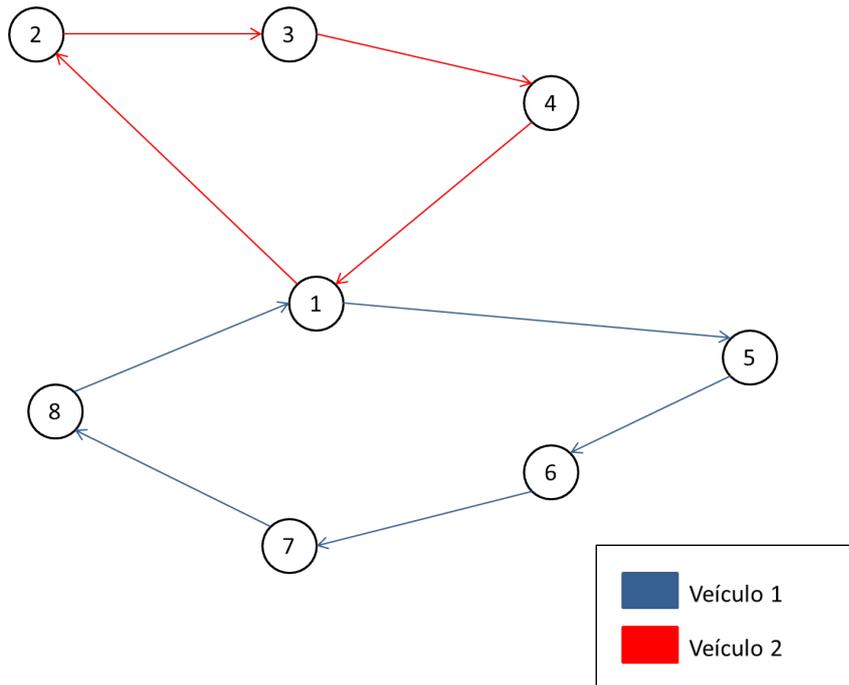


Figura 5 – Exemplo de um caso de atendimento às restrições (9) a (11).

A Equação (10) garante que o roteiro estará completamente conectado, evitando configurações conforme apresentadas na Figura 6.

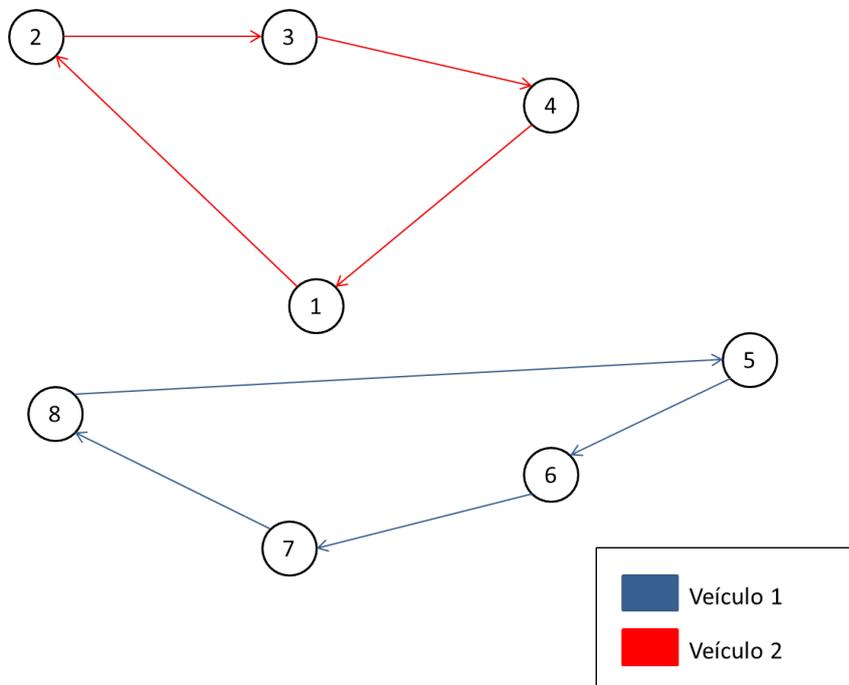


Figura 6 – Exemplo de um caso de não atendimento à restrição (10).

A Equação (12) afirma que se um veículo v viaja do cliente p para o cliente k na posição t , então o mesmo veículo, na posição $t + 1$, irá viajar do cliente k para o cliente l .

A restrição da Equação (13) vai garantir que a capacidade de cada veículo não seja excedida

(Figura 7), e a Equação (14) apenas define o domínio da variável de decisão.

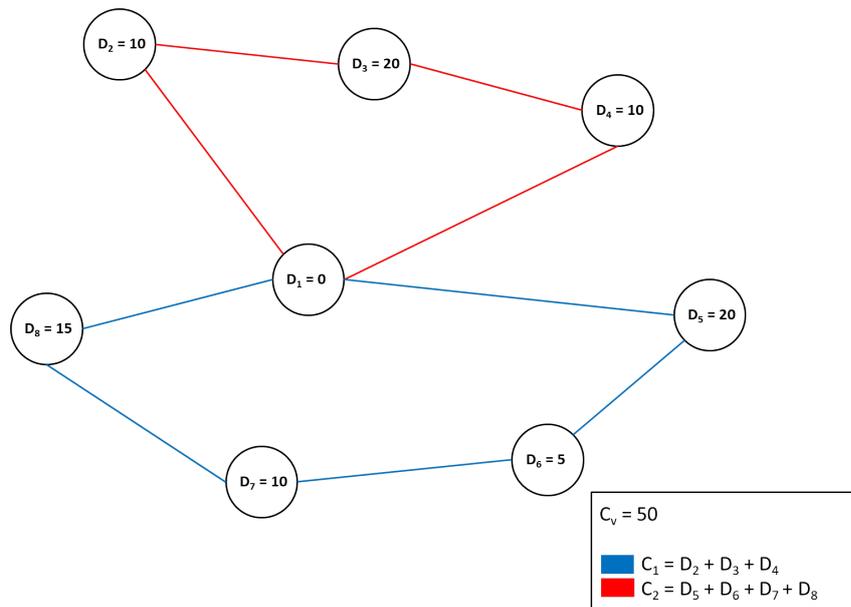


Figura 7 – Exemplo de um caso de atendimento à restrição (13).

Capítulo 5

Metodologia

Neste capítulo será apresentado o algoritmo híbrido proposto, os tipos de hibridização estudados, o funcionamento do método, a forma como é feita a comunicação entre a heurística ACO e o software CPLEX, e as estratégias usadas para garantir a execução do híbrido.

5.1 A Heurística ACO

A representação da formiga é a primeira parte da implementação, através da qual ocorre a codificação do problema de interesse. Esta codificação pode influenciar negativamente no resultado final, caso seja mal formulada. Neste trabalho, cada formiga produz um roteiro, o qual é composto por um conjunto de rotas. Estas rotas são computacionalmente representadas por meio de listas de números inteiros, os quais representam cada cliente visitado, conforme ilustra a Figura 8.

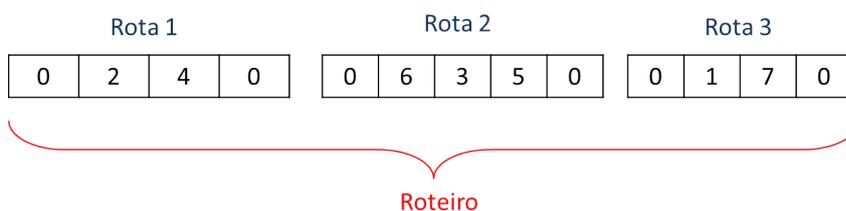


Figura 8 – Representação de uma formiga na ACO

A sequência dos números inseridos nestas listas (Figura 8) correspondem a ordem de visita dos clientes de cada rota, sendo esta com início e fim no depósito central (representado pelo número 0 (zero)).

Além do conjunto de rotas, a formiga também mantém uma lista dos clientes nos quais ela já passou com intuito de evitar que estes sejam visitados novamente. Esta lista é atualizada toda vez em que ela decide para qual cliente irá seguir, assegurando, dessa forma, que a

partir do momento em que o cliente é inserido nessa lista, não há chances dele ser visitado mais de uma vez.

A cada rota produzida, um veículo exclusivo é alocado a ela, evitando assim que ele seja usado em outra rota, o que por consequência garante que um cliente não será visitado por mais de um veículo.

Para respeitar a capacidade do veículo, a qual foi modelada na restrição da Equação 13, antes de decidir sobre qual caminho seguir, o algoritmo deverá verificar todos os clientes ainda não visitados computando a demanda de cada um deles. Caso essa seja maior do que a capacidade atual do veículo, esse cliente não entrará para o grupo de candidatos à próxima visitação, conforme ilustra a Figura 9.

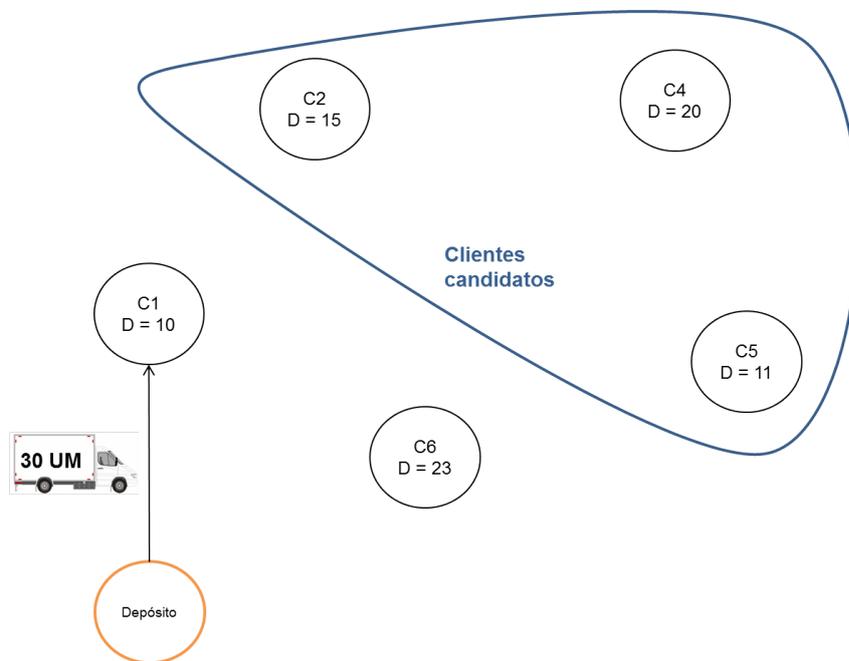


Figura 9 – Representação gráfica de como a ACO assegura que a capacidade do veículo será respeitada.

Como premissa do CVRP, sabe-se que uma rota sempre terá início e fim no depósito. Dessa forma, se, após analisar todos clientes disponíveis, averiguar que a capacidade do caminhão será ultrapassada, a única opção será retornar para o nó inicial. Com base nessa estratégia, a ACO garante que a rota estará completamente conectada, evitando subrotas ilegais, como a da Figura 6.

No processo de construção da rota, no qual o algoritmo deve decidir para qual cliente seguir, foi utilizado um parâmetro $q0$ para influenciar esta tomada de decisão. Na prática, um número entre 0 e 1 é sorteado e, caso este seja menor do que o valor determinado para $q0$, a formiga seguirá seu percurso de maneira gulosa, e, caso contrário, ela continuará o caminho de forma aleatória tendo como base o método da roleta.

Em uma decisão gulosa a formiga fará sua escolha pela rota com maior atratividade, ou seja, ela escolherá o caminho com maior índice de feromônio. Quando sua decisão se basear na aleatoriedade, este agente recorrerá ao método da roleta para prosseguir. Neste cenário, o índice de aptidão utilizado corresponde à atratividade do arco, o qual é representado na roleta proporcionalmente à sua aptidão. Dessa forma, arcos com alta aptidão terão uma porção maior da roleta, enquanto os de aptidão mais baixa é dada uma porção menor da roleta. Após girar a roleta, o resultado dirá à formiga qual caminho ela deverá seguir.

5.2 CPLEX

Neste trabalho é utilizado um recurso do CPLEX conhecido como *MIP start* (ou *Início de MIP*), o qual corresponde a um mecanismo que permite solucionar um problema de programação inteira por meio de sugestões, e que ajuda o CPLEX a localizar uma solução inicial para o problema informado.

Estas sugestões consistem em pares de variáveis e valores que são informados ao CPLEX, sendo este responsável por decidir sobre como construir um ponto de início a partir desses pares de variável/valor.

Quando um início de MIP é utilizado, o CPLEX processa os dados informados antes de iniciar a otimização do problema. Se um ou mais dos inícios de MIP definirem uma solução, o software da IBM instalará a melhor dessas soluções como a solução inicial. Ter uma solução logo no início do processo de ramificação e corte permite que o CPLEX elimine partes do espaço de busca e com isso possa trabalhar em árvores de ramificação e corte menores. Mais detalhes acerca do funcionamento desta estratégia pode ser verificada no Apêndice [A](#).

5.3 O Algoritmo Híbrido

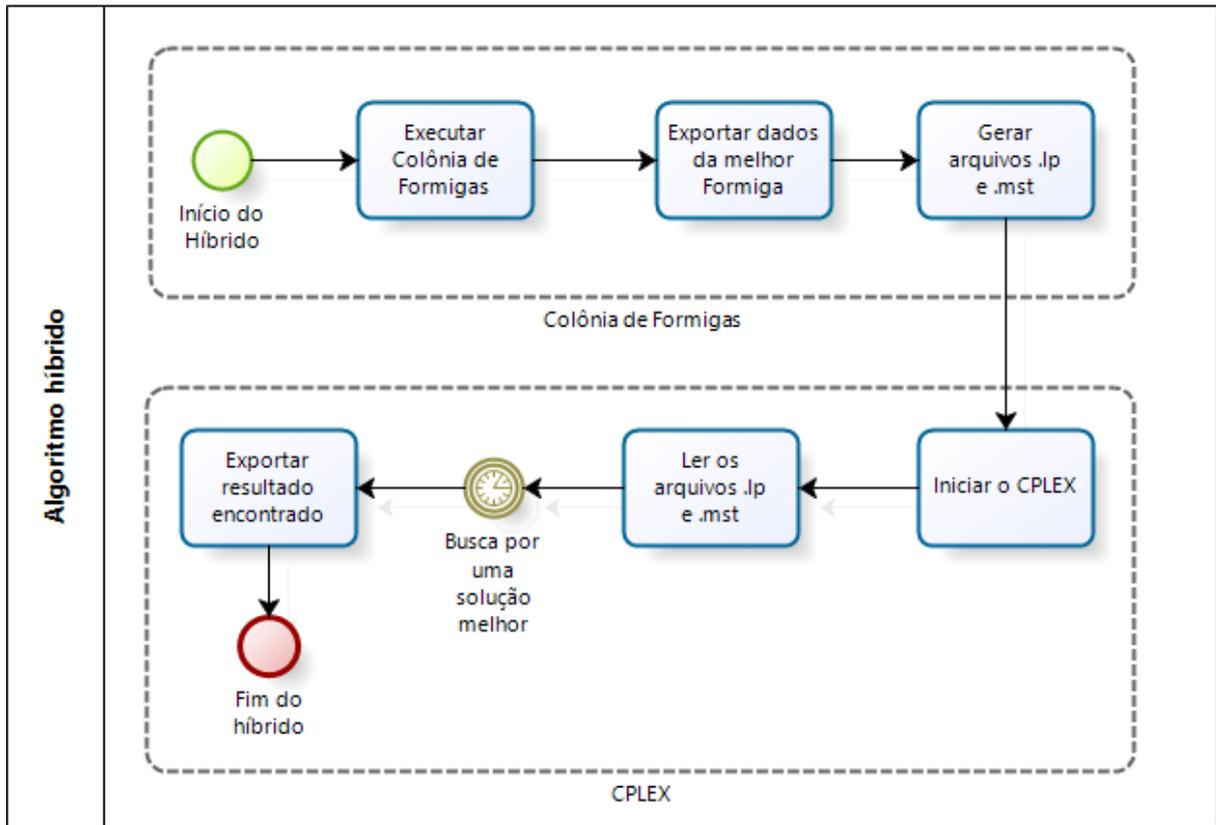
O algoritmo híbrido é caracterizado pela integração de dois métodos distintos, e para que tal proposta seja eficiente, é necessário garantir a comunicação entre as estratégias. Esta comunicação pode ser feita de duas diferentes maneiras, sendo esta unidirecional ou bidirecional, as quais irão determinar a necessidade da realimentação dos métodos.

Na comunicação unidirecional a heurística constrói uma solução inicial e a otimiza por um determinado tempo. Ao final, as informações processadas são exportadas e utilizadas pelo CPLEX para dar prosseguimento ao processo de melhoria do resultado. Este software executará a tarefa durante um tempo, e a solução final produzida por ele será a solução final do híbrido.

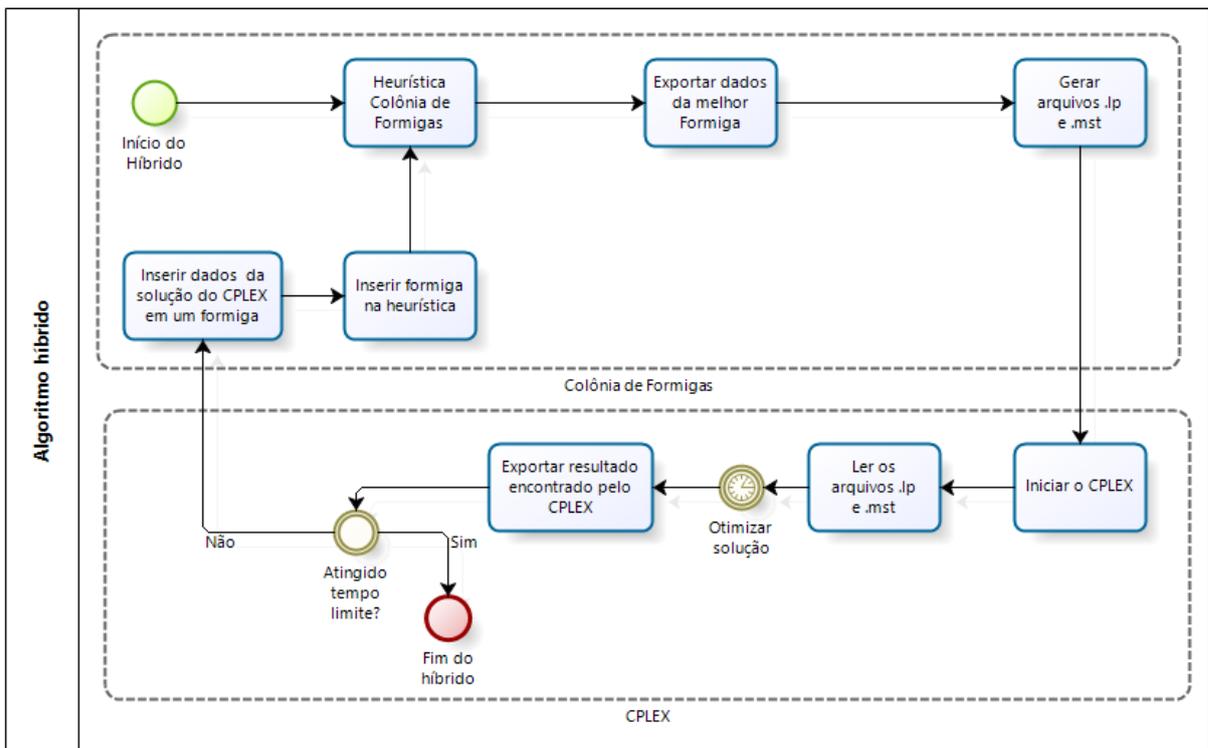
A realimentação existe quando a comunicação entre os métodos é bidirecional. Nesta, da mesma forma que o CPLEX utiliza os resultados produzidos pela heurística, esta passa a utilizar as informações produzidas pelo software da IBM.

Basicamente a primeira parte do processo permanece da mesma forma que ocorre na comunicação unidirecional, com a ACO fornecendo seus dados ao CPLEX, o qual deverá dar continuidade ao processo de otimização procurando melhorar a solução. No tipo de comunicação bidirecional o primeiro resultado produzido pelo CPLEX não corresponderá ao fim do processamento do híbrido, sendo estes dados usados pela heurística em um, ou mais, ciclo de hibridização.

As Figuras [10a](#) e [10b](#) ilustram, respectivamente, os tipos de comunicação unidirecional e bidirecional para o algoritmo híbrido, detalhando as etapas do processo.



(a) Sem realimentação: Comunicação unidirecional



(b) Com realimentação: Comunicação bidirecional

Figura 10 – Modelos de comunicação do algoritmo híbrido

Para garantir que a comunicação entre os métodos ocorra de forma correta é preciso trabalhar na integração respeitando as características de cada estratégia. Para isso, todas as informações geradas ao final de cada execução são registradas em arquivos, podendo assim ser utilizadas no outro método de forma eficiente.

Tais arquivos, denominados neste trabalho como arquivos de integração, são gerados de forma transparente ao usuário, o qual não precisará interagir com o algoritmo híbrido e interferir no processo de hibridização. Mais detalhes sobre tais arquivos podem ser encontrados no Apêndice [A](#).

Capítulo 6

Análise e Discussão dos Resultados

Neste capítulo serão apresentados os testes computacionais realizados, bem como uma análise crítica sobre os resultados obtidos. Planeja-se um experimento para validar a hipótese de que o método híbrido é melhor, em média, que os métodos individuais.

6.1 Testes Computacionais

Para melhorar a compreensão do leitor, recorda-se que no problema estudado neste trabalho a frota de veículos é homogênea, todos os clientes devem ser atendidos, e isto deve ocorrer exclusivamente por um único caminhão.

A estratégia proposta foi avaliada em um conjunto de instâncias retiradas da CVRPLIB¹. Para efeito de análise, a heurística e o modelo matemático foram executados separadamente para, logo após, serem estudados em conjuntos.

Para medir a qualidade das soluções foi utilizada a métrica conhecida como GAP. Este índice é definido de acordo com a Equação 15, sabendo que *solucaoEncontrada* corresponde ao resultado do método estudado e, *solucaoLiteratura* diz respeito a melhor solução encontrada na literatura.

$$GAP = \frac{\textit{solucaoEncontrada} - \textit{solucaoLiteratura}}{\textit{solucaoLiteratura}} \quad (15)$$

Uchoa et al. (2014) afirma que na literatura tornou-se comum seguir a convenção e utilizar a estratégia de arredondamento para o inteiro mais próximo. Em alguns trabalho as distâncias entre os pontos de interesse são calculadas pela distância euclidiana, mas o resultado é convertido em um valor inteiro de modo com que a distância final seja dada sobre esta mesma condição.

¹Disponível em: <<http://vrp.atd-lab.inf.puc-rio.br/>>

Neste trabalho, optou-se por apresentar os resultados em ponto flutuante, uma vez que um determinado grupo de instâncias a serem testadas estão representadas na literatura com valores decimais. Quando a rota encontrada pelo método proposto corresponder a mesma disposição disponibilizada no repositório da CVRPLIB, esta, conseqüentemente, diz respeito a solução ótima, e, portanto, será identificada na tabela de resultado por meio do símbolo *.

Para efeito de identificação dos resultados, o número de veículos utilizados na entrega é dado pela variável k . O número de clientes, somado ao depósito, compõem a variável n .

Os primeiros testes foram baseados nas instâncias inspiradas nos trabalhos de [Augerat et al. \(1995\)](#) e [Christofides e Eilon \(1969\)](#), variando o tamanho entre 22 e 50 clientes. Para análise, a heurística e o CPLEXo foram executados separadamente com o objetivo de analisá-los independentemente, para, em seguida, estudá-los em conjunto.

O CPLEX foi analisado durante o período de 4 horas, testando alguns de seus parâmetros para avaliar a melhor configuração do otimizador e a qualidade da solução produzida. Isto foi necessário pois à medida que o problema aumenta de tamanho, o CPLEX trabalha mais lentamente, consome mais memória e necessita de um tempo computacional muito grande para encontrar o ótimo do problema.

Os parâmetros apresentados a seguir foram configurados com base nas informações disponíveis no site da IBM, visando reduzir o tamanho da árvore de busca, o consumo de memória e, conseqüentemente, aumentar a eficiência do método.

- *mip strategy nodeselect*: Este parâmetro compreende as estratégias de seleção do nó da árvore de busca.
- *mip strategy bbinterval*: Corresponde a frequência com que um nó com melhor limite é selecionado pelo otimizador.
- *mip strategy variableselect*: Diz respeito a regra de seleção das variáveis dentro do processo de ramificação.

No caso do parâmetro *nodeselect*, pode-se trabalhar com a opção de Busca em Profundidade para evitar o crescimento exagerado da árvore de busca. Esta regra, segundo pesquisa realizada, raramente gera uma longa lista de nós inexplorados, fazendo com que esta árvore consuma menos memória principal do computador.

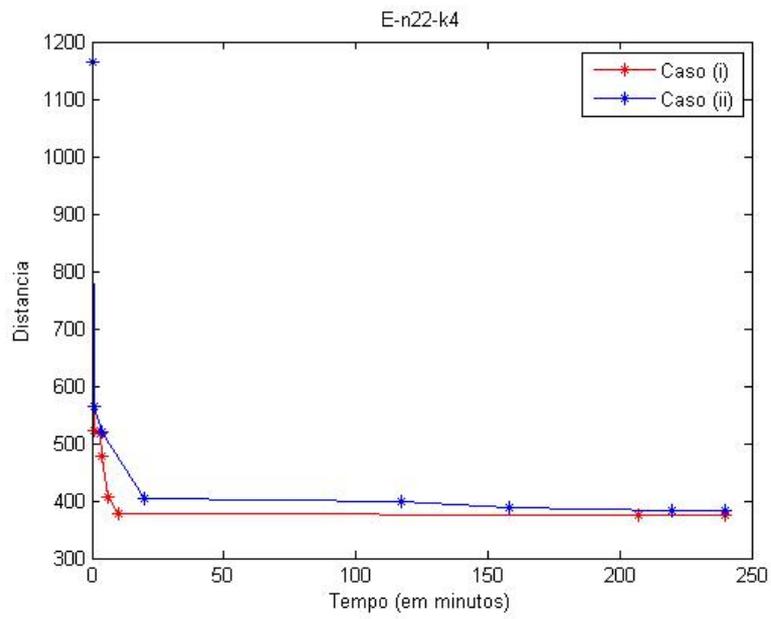
A opção padrão do *nodeselect* trabalha com a regra de busca do Melhor Limite, o que significa que o nó com o melhor valor da função objetivo deve ser selecionado. Caso se opte por esta regra, pode-se configurar o intervalo de seleção através do parâmetro *bbinterval*. O manual de usuário do CPLEX informa que mostrou-se benéfico manter o opção padrão do software, a qual selecionará o nó de melhor limite ocasionalmente.

A IBM informa que em problemas grandes e difíceis alterar o parâmetro *variableselect* pode influenciar na eficácia do otimizador. Ao mudar a regra para utilizar a estratégia de Ramificação Forte, aumenta-se o esforço computacional empregado na seleção da variável, gerando-se menos nós, e, conseqüentemente, consome-se menos memória. Esta opção se baseia na resolução parcial de subproblemas ramificados para identificar qual é o resultado mais promissor e, segundo pesquisado, é mais rápida que a opção padrão do otimizador.

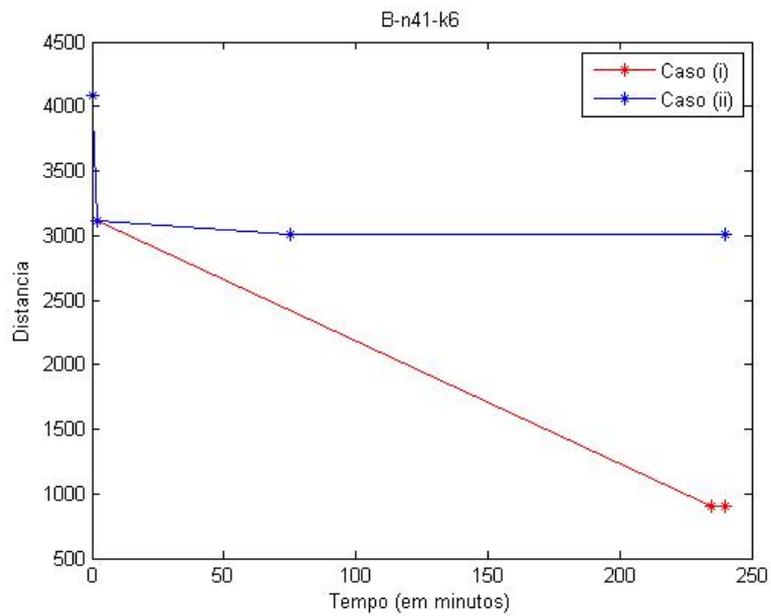
A influência destes parâmetros nas soluções finais e no processo de busca foram analisados em dois casos:

- Caso (i): seguindo as configurações padrões do otimizador.
 - Busca do Melhor Limite, com seleção do melhor nó ocasionalmente.
 - A seleção de variável ocorrerá com base no problema e em seu progresso.
- Caso (ii): de acordo com alterações no parâmetros citados anteriormente.
 - Busca em Profundidade.
 - Seleção de variável pela regra de Ramificação Forte.

O caso de teste que utiliza as configurações padrões do CPLEX se mostrou mais eficiente. O processo de melhoria e a solução final deste primeiro cenário foram superiores em relação às modificações realizadas para o caso (ii), conforme ilustra a Figura 11. Desta forma, neste trabalho a melhor condição para se trabalhar com o software da IBM corresponde à utilização das opções padrões do CPLEX.



(a) E-n22-k4



(b) B-n41-k6

Figura 11 – Resultados dos testes realizados no CPLEX

Para comparar a eficiência do CPLEX em relação à heurística, as instâncias foram executadas cada uma 11 vezes, pelo mesmo período de 4 horas, sendo seus resultados são apresentados na Tabela 1.

Tabela 1 – Resultados computacionais para os métodos heurístico e exato

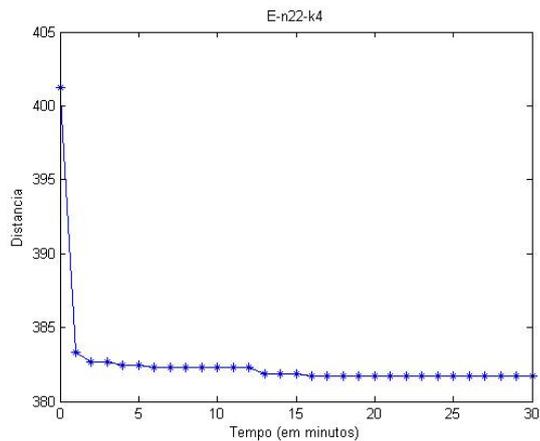
Instância	n	k	Solução Literatura	Colônia de Formigas		CPLEX	
				<i>Solução média</i>	<i>GAP</i>	<i>Solução</i>	<i>GAP</i>
E-n22-k4	22	4	375	381,69	0,02	375,28	0,00
E-n23-k3	23	3	569	594,08	0,04	568,56	0,00
E-n30-k3	30	3	534	555,57	0,04	553,09	0,03
E-n33-k4	33	4	835	854,92	0,02	899,78	0,08
B-n41-k6	41	6	829	839,84	0,01	944,47	0,14
B-n43-k6	43	6	742	760,43	0,02	911,62	0,23
B-n44-k7	44	7	909	950,98	0,05	1299,22	0,43
B-n45-k5	45	5	751	788,42	0,05	941,28	0,25
B-n45-k6	45	6	678	716,76	0,06	1009,18	0,49
E-n51-k5	51	5	521	589,93	0,13	1979,96	2,80

Os valores informados na coluna **Solução Literatura** correspondem ao ótimo dos problemas. Os resultados produzidos por cada método, acompanhados dos seus respectivos *GAP*, são apresentados nas colunas *Solução* e *GAP*.

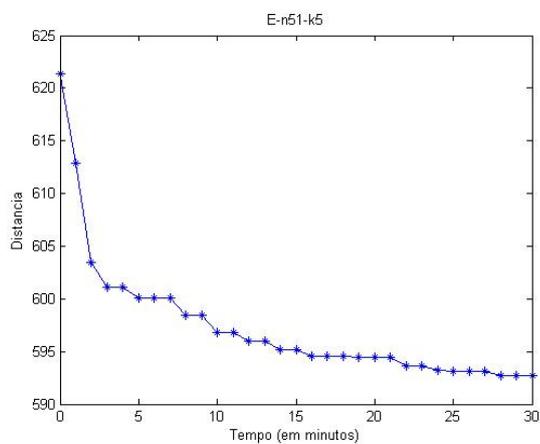
O CPLEX encontrou o resultado ótimo em dois casos testados, o que não ocorreu nas demais instâncias, uma vez que seu tempo de execução foi limitado às 4 horas de teste. A ACO, com o tempo também limitado a 4 horas de execução, não conseguiu chegar a nenhum resultado ótimo como ocorreu com o CPLEX.

Exceto pelas instâncias **E-n22-k4** e **E-n23-k3**, os resultados apresentados pelo CPLEX mostraram qualidades inferiores quando comparados a heurística, levando a conclusão de esta se mostrou ser mais eficiente que o software da IBM.

Em relação à heurística ACO, observou-se o processo de convergência, para melhor entender o seu comportamento no decorrer do tempo. À medida que se aproxima da solução da literatura, notou-se nas instâncias testadas um comportamento no qual há a estabilização dos resultados, que podem ser observados, por exemplo, nas Figuras [12a](#) e [12b](#).



(a) E-n22-k4



(b) E-n51-k5

Figura 12 – Resultados da ACO nos primeiros 30 minutos de execução.

Conforme ilustra a Tabela 2, os resultados mostraram que a estabilização ocorre rapidamente nas instâncias pequenas, e tende a acontecer mais tardiamente à medida em que as instâncias tornam-se mais complexas. Cabe ressaltar que os dados apresentados na coluna **Solução Literatura** correspondem aos ótimos dos problemas.

Tabela 2 – Evolução dos resultados médios da ACO

Instância	n	k	Solução Literatura	Resultados médios em:			
				30 minutos	1 hora	2 horas	4 horas
E-n22-k4	22	4	375	381,69	381,69	381,69	381,69
E-n23-k3	23	3	569	594,08	594,08	594,08	594,08
E-n30-k3	30	3	534	560,55	560,55	555,57	555,57
E-n33-k4	33	4	835	861,10	858,47	858,47	854,92
B-n41-k6	41	6	829	847,17	843,34	842,92	839,84
B-n43-k6	43	6	742	760,43	760,43	760,43	760,43
B-n44-k7	44	7	909	961,91	956,57	950,98	950,98
B-n45-k5	45	5	751	801,67	801,62	800,93	788,42
B-n45-k6	45	6	678	725,23	725,23	719,90	716,76
E-n51-k5	51	5	521	589,93	589,93	589,93	589,93

Com base nos resultados anteriores, os próximos experimentos correspondem à unificação dos métodos heurístico e exato, ou seja, a hibridização em si.

Neste trabalho buscou-se verificar a necessidade do processo de realimentação, o qual diz respeito à troca de informações entre a ACO e o CPLEX em períodos de tempo pré-determinados. Desta forma, duas análises foram feitas:

- A comunicação unidirecional ACO → CPLEX: a ACO envia sua solução para que o CPLEX continue o processamento, encerrando-se neste momento o algoritmo.
- A comunicação bidirecional ACO ↔ CPLEX: a ACO envia seu resultado ao CPLEX, que continua com o processamento por um determinado período de tempo. Ao final, este devolve à heurística a solução encontrada, e esta reinicia o ciclo da hibridização.

Para avaliar a realimentação, foram especificados dois casos de testes com total de 4 horas cada, com diferentes configurações de tempo de execução:

- Caso (iii): Neste cenário, a troca de informações entre a ACO ↔ CPLEX ocorreria ao final de ciclos de trinta minutos cada. Estes foram divididos de acordo com as seguintes configurações:
 - [5 - 25]: cinco minutos para ACO / vinte e cinco minutos para o CPLEX.
 - [15 - 15]: quinze minutos para ACO / quinze minutos para o CPLEX.
 - [10 - 20]: dez minutos para ACO / vinte minutos para o CPLEX.
- Caso (iv): Neste, o processo de troca de informações entre a ACO ↔ CPLEX ocorreria ao final de ciclos de uma hora cada, divididos de acordo com as seguintes configurações:

[10 - 50]: dez minutos para ACO / cinquenta minutos para o CPLEX.

[20 - 40]: vinte minutos para ACO / quarenta minutos para o CPLEX.

A Tabela 3 mostra os resultados do caso (iii) e do caso (iv) para diversas instâncias de teste. Para efeito de comparação, também são apresentados os valores ótimos e os resultados do CPLEX pelo mesmo período de tempo.

Pode-se perceber que as soluções não apresentaram variações significantes entre as configurações propostas.

Tabela 3 – Resultados do modelo de comunicação com realimentação

Instâncias	Literatura	Colônia de Formigas										CPLEX	GAP
		[5 - 25]	GAP	[15 - 15]	GAP	[10 - 20]	GAP	[10-50]	GAP	[20-40]	GAP		
E-n22-k4	375	381,69	0,02	377,26	0,01	375,27	0,00	379,43	0,01	379,43	0,01	375	0,00
E-n23-k3	569	568,56	0,00	568,56	0,00	568,56	0,00	568,56	0,00	568,56	0,00	569	0,00
E-n30-k3	534	534,00	0,00	534,26	0,00	557,13	0,04	536,43	0,00	534,00	0,00	553	0,03
E-n33-k4	835	863,06	0,03	857,40	0,03	862,76	0,03	860,61	0,03	860,82	0,03	900	0,08
B-n41-k6	829	846,61	0,02	843,93	0,02	843,18	0,02	848,70	0,02	847,23	0,02	944	0,14
B-n43-k6	742	761,51	0,03	762,20	0,03	763,08	0,03	761,51	0,03	762,42	0,03	912	0,23
B-n44-k7	909	953,33	0,05	952,58	0,05	948,92	0,04	959,64	0,06	951,33	0,05	1299	0,43
B-n45-k5	751	797,49	0,06	799,54	0,06	796,02	0,06	801,75	0,07	792,67	0,06	941	0,25
B-n45-k6	678	719,47	0,06	711,49	0,05	717,34	0,06	708,88	0,05	700,56	0,03	1009	0,49
E-n51-k5	521	599,36	0,15	594,33	0,14	599,62	0,15	591,23	0,13	587,17	0,13	1980	2,80

Desta forma, para entender melhor o processo de comunicação bidirecional do algoritmo híbrido, foi necessário levantar as informações entre as soluções trocadas pelos métodos. A Figura 13 ilustra, através de alguns resultados, como foi feita esta análise por meio de uma classificação de cores, onde o tom mais vermelho define a solução inicial e o tom verde indica a solução final estabilizada.

A transição de cores procura demonstrar a otimização da rota conforme a execução do algoritmo. A coluna "**Solução Inicial**" indica a o valor gerado inicialmente pela Colônia de Formigas, enquanto a coluna "**ACO**" mostra a solução produzida por esta heurística ao final de seu processamento. Por fim, a coluna "**CPLEX**" indica o resultado final após a execução do software.

Instância	Configuração	Solução Inicial	ACO		CPLEX																	
			ACO	CPLEX																		
E-n23-k3	5-25	629,79	595,87	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56
	15-15	616,94	594,07	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56
	10-20	603,19	594,08	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56
	10-50	618,23	598,05	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56
	20-40	616,67	594,08	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56	568,56
B-n41-k6	5-25	856,09	847,54	847,54	847,54	847,54	847,54	847,54	847,54	847,54	847,54	847,54	847,54	847,54	847,54	847,54	847,54	847,54	847,54	846,61	846,61	846,61
	15-15	854,66	849,03	849,03	848,25	848,25	848,25	848,25	848,25	848,25	846,30	846,30	846,30	846,30	846,30	846,30	846,30	846,30	846,30	843,93	843,93	843,93
	10-20	868,97	849,63	849,63	845,74	845,74	845,74	845,74	845,74	845,74	843,18	843,18	843,18	843,18	843,18	843,18	843,18	843,18	843,18	843,18	843,18	843,18
	10-50	861,06	848,70	848,70	848,70	848,70	848,70	848,70	848,70	848,70	848,70	848,70	848,70	848,70	848,70	848,70	848,70	848,70	848,70	843,18	843,18	843,18
	20-40	865,73	852,08	852,08	847,52	847,52	847,52	847,52	847,52	847,52	847,52	847,52	847,52	847,52	847,52	847,52	847,52	847,52	847,52	846,61	846,61	846,61
E-n51-k5	5-25	629,60	601,98	601,98	601,98	600,51	600,51	600,51	600,51	600,51	599,35	599,35	599,35	599,35	599,35	599,35	599,35	599,35	599,35	599,35	599,35	599,35
	15-15	618,26	603,19	603,19	594,32	594,32	594,32	594,32	594,32	594,32	594,32	594,32	594,32	594,32	594,32	594,32	594,32	594,32	594,32	594,32	594,32	594,32
	10-20	621,40	599,62	599,62	599,62	599,62	599,62	599,62	599,62	599,62	599,62	599,62	599,62	599,62	599,62	599,62	599,62	599,62	599,62	599,62	599,62	599,62
	10-50	642,89	601,89	601,89	591,23	591,23	591,23	591,23	591,23	591,23	591,23	591,23	591,23	591,23	591,23	591,23	591,23	591,23	591,23	591,23	591,23	591,23
	20-40	626,06	592,19	592,19	587,17	587,17	587,17	587,17	587,17	587,17	587,17	587,17	587,17	587,17	587,17	587,17	587,17	587,17	587,17	587,17	587,17	587,17

Figura 13 – Evolução dos resultados durante o processo de realimentação do algoritmo (as partes cinzas indicam que não houve execução)

Os resultados mostram que nas instâncias pequenas o CPLEX determina as soluções ótimas já no primeiro ciclo. Este panorama se inverteu com o crescimento do número de clientes na rota, passando a pertencer à heurística a qualidade sobre a solução final. Além disso, é possível observar que a ideia dos ciclos tornou-se ineficiente nas instâncias maiores. O CPLEX não consegue, no tempo determinado, melhorar as soluções da heurística, a qual também não conseguiu aperfeiçoar as soluções devolvidas pelo otimizador.

Além disso, o software da IBM não melhorou os resultados da ACO, isto porque precisa de mais tempo para processar os dados das instâncias maiores. No caso da heurística, concluiu-se que este método começou a atingir sua condição de estagnação para os tempos definidos.

Com base nestas informações, os próximos casos de testes não irão considerar o processo de realimentação. Serão observados as soluções geradas pela comunicação unidirecional entre a ACO → CPLEX, dedicando um tempo de execução maior para o software da IBM.

Nestes experimentos foram levantados dois tipos de configuração diferentes, para o mesmo tempo total de 4 horas de execução, excluindo a comunicação bidirecional:

- Caso (v): Neste cenário, a troca de informações entre a ACO → CPLEX ocorreria ao final de um tempo pré-determinado. Este foi definido de duas formas distintas:
 - [10 - 230]: dez minutos para ACO / duzentos e trinta minutos para o CPLEX.
 - [20 - 220]: vinte minutos para ACO / duzentos e vinte minutos para o CPLEX.

Tabela 4 – Resultados do modelo de comunicação sem realimentação

Instâncias	Solução Literatura	Colônia de Formigas				CPLEX	GAP
		[20 - 220]	GAP	[10 - 230]	GAP		
E-n22-k4	375	375,28	0,00	375,28	0,00	375	0,00
E-n23-k3	569	568,56	0,00	569,00	0,00	569	0,00
E-n30-k3	534	544,27	0,02	553,85	0,04	553	0,03
E-n33-k4	835	863,06	0,03	860,57	0,03	900	0,08
B-n41-k6	829	846,24	0,02	846,69	0,02	944	0,14
B-n43-k6	742	762,38	0,03	763,50	0,03	912	0,23
B-n44-k7	909	973,07	0,07	964,20	0,06	1299	0,43
B-n45-k5	751	803,68	0,07	806,32	0,07	941	0,25
B-n45-k6	678	724,46	0,07	727,73	0,07	1009	0,49
E-n51-k5	521	590,12	0,13	607,22	0,17	1980	2,80

Observando a Tabela 4, nota-se que os resultados obtidos não apresentaram melhoras significativas quando comparados aos encontrados no processo de realimentação. Desta

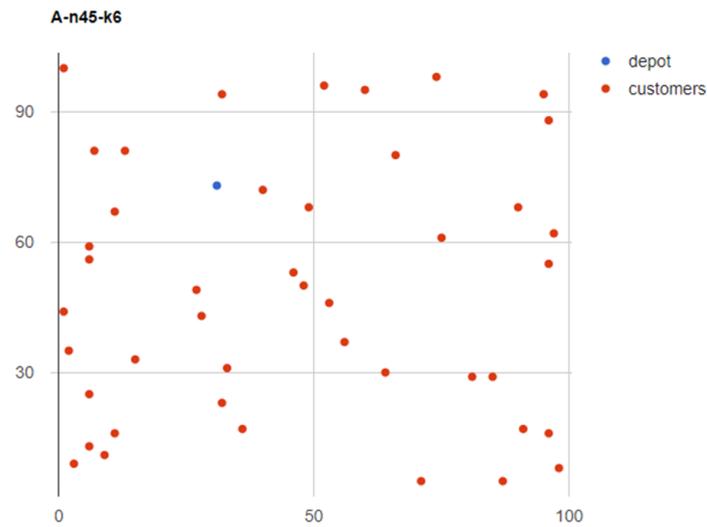
forma, considerando os resultados produzidos no total de 4 horas, e se atentando ao fato de que o tipo de comunicação não interferiu nos resultados, entende-se que é necessário dar mais tempo de processamento ao algoritmo híbrido.

Desta forma, optou-se por utilizar o método de comunicação unidirecional, se atentando à característica de estabilização dos resultados produzidos pela ACO e à dificuldade de processamento do CPLEX em problemas maiores para determinar o tempo de execução de cada método dentro da abordagem híbrida, chegando assim a uma correta calibração do método proposto neste trabalho.

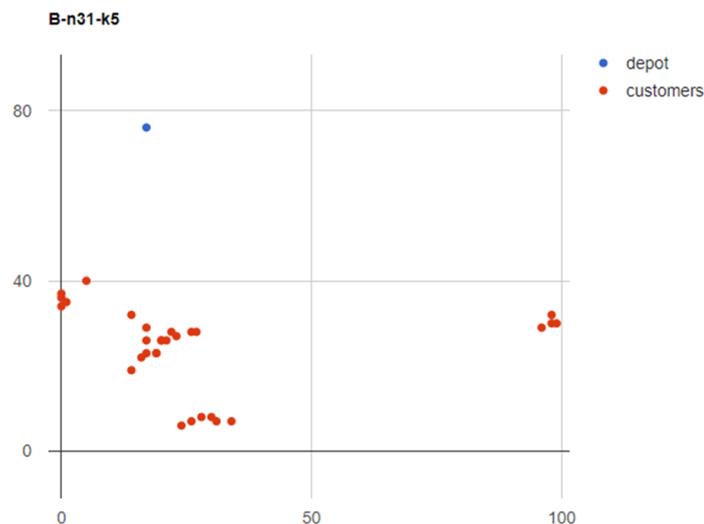
6.2 Resultados

Após as análises anteriores, os resultados dos testes serão apresentados e analisados para mais instâncias da literatura, considerando para isso que o algoritmo híbrido proposto terá um total de 10 horas de execução, com duas horas dedicadas à execução da heurística, e oito horas reservadas ao processamento do CPLEX.

As instâncias dessa seção foram retiradas da CVRPLIB, sendo organizadas neste trabalho em conjuntos denominados E, A, P, B e CMT. Nos conjuntos E, A, P e CMT os pontos (clientes) estão dispersos graficamente (Figura 14a), enquanto no conjunto B estes encontram-se clusterizados (Figura 14b). Em todos, exceto pelo conjunto CMT, as soluções da literatura apresentadas correspondem aos valores ótimos.



(a) Instância cujos pontos estão dispersos



(b) Instância cujos pontos estão agrupados

Figura 14 – Caracterização dos tipos de instâncias testadas

O primeiro grupo de instâncias testadas foram as do conjunto E. Seus resultados são listados na Tabela 5, e comparando-os conclui-se que o algoritmo híbrido foi superior ao CPLEX e à ACO.

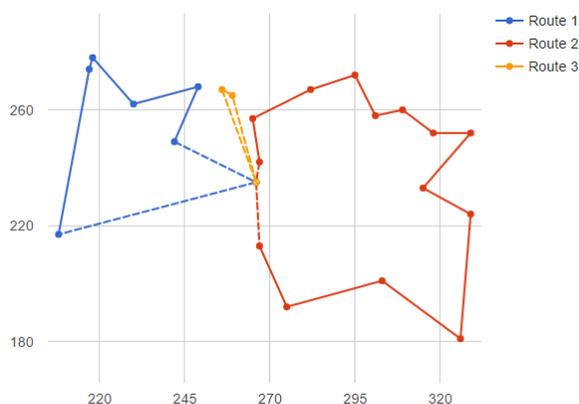
As soluções apresentadas são superiores às encontradas pelos métodos individuais, assim como o **GAP Médio** é melhor do que os dos demais. Além disso, ao observar o indicador de **Desvio Padrão**, verifica-se que o híbrido apresentou índices de variabilidade das soluções melhores do que a heurística para este conjunto.

Analisando os resultados, a abordagem híbrida se mostrou superior ao CPLEX e à Colônia de Formigas em todos os casos testados. Além disso, o baixo índice de desvio padrão do híbrido demonstra como as soluções produzidas foram de qualidade.

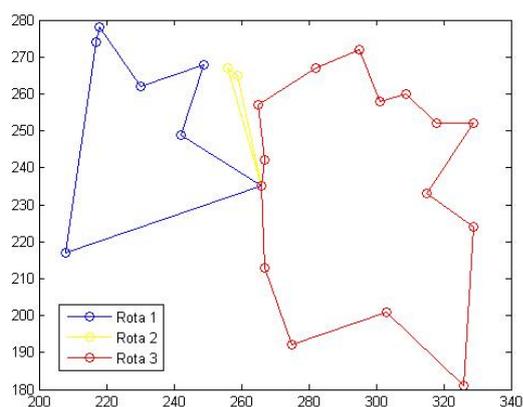
Tabela 5 – Resultados das instâncias do conjunto E (* são valores ótimos).

Instância	n	k	Solução Literatura	Híbrido			CPLEX		Colônia de Formigas		
				Solução Média	Desvio Padrão	GAP	Solução	GAP	Solução média	Desvio Padrão	GAP
E-n22-k4	22	4	375	375,28*	0,000	0,000	375,28*	0,000	381,69	0,000	0,018
E-n23-k3	23	3	569	568,56*	0,000	0,000	568,56*	0,000	594,08	0,000	0,044
E-n30-k3	30	3	534	540,96	1,801	0,013	540,20	0,012	554,82	1,050	0,039
E-n33-k4	33	4	835	855,81	1,370	0,025	936,54	0,122	857,76	3,841	0,027
E-n51-k5	51	5	521	588,54	0,261	0,129	635,15	0,219	588,69	4,392	0,130
GAP Médio				0,034			0,071		0,052		
Desvio padrão do GAP Médio				0,055			0,097		0,045		

Outro ponto observado é que nas instâncias **E-n22-k4** e **E-n23-k3** o algoritmo híbrido conseguiu, assim como o CPLEX, encontrar as mesmas soluções da literatura. As Figuras 15 e 16 realizam uma comparação das rotas disponíveis na CVRPLIB, com as rotas encontradas pela estratégia proposta, comprovando tal informação.

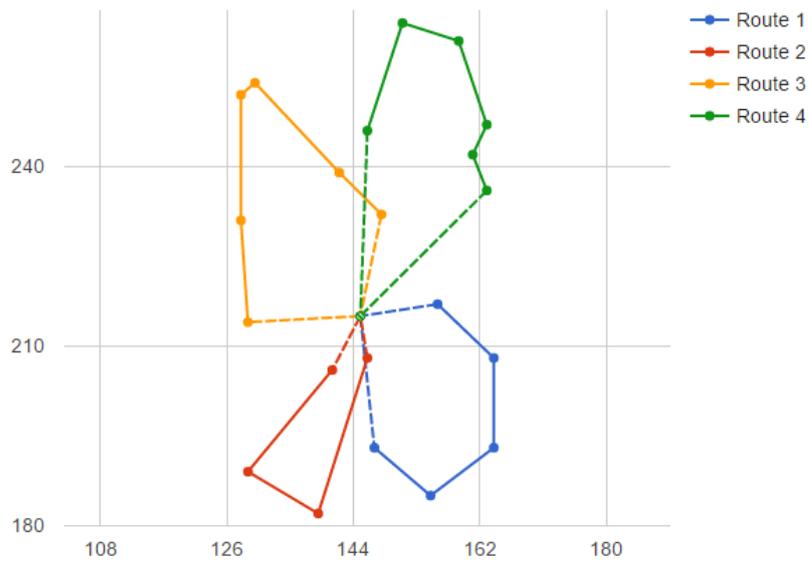


(a) Rota da literatura

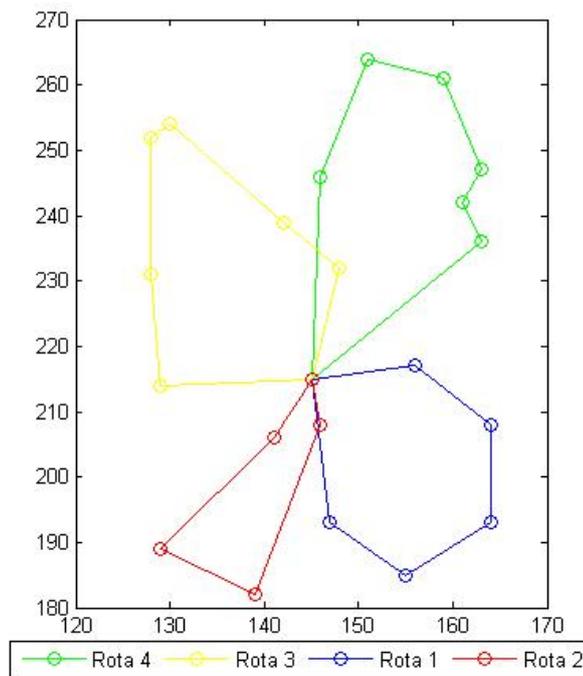


(b) Rota encontrada pelo híbrido

Figura 15 – Resultados para E-n23-k3



(a) Rota ótima



(b) Rota do híbrido

Figura 16 – Resultados para E-n22-k4

O próximo grupo testado compreende as instâncias do conjunto A (Tabela 6). O híbrido não conseguiu atingir os mesmos resultados da literatura, assim como não apresentou resultados tão próximos quanto os obtidos nas instâncias do grupo E.

Ainda assim, em uma análise individual, ao observar os dados da Tabela 6, a estratégia híbrida proposta neste trabalho apresentou resultados melhores do que os mostrados pelos métodos individualmente.

Tabela 6 – Resultados das instâncias do conjunto A.

Instância	n	k	Solução Literatura	Híbrido			CPLEX		Colônia de Formigas		
				Solução Média	Desvio Padrão	GAP	Solução Média	GAP	Solução Média	Desvio Padrão	GAP
A-n32-k5	32	5	784	831,95	2,28	0,061	872,42	0,113	829,60	3,149	0,058
A-n33-k5	33	5	661	682,82	17,45	0,033	722,78	0,093	696,74	5,049	0,054
A-n33-k6	33	6	742	770,53	14,22	0,038	780,93	0,052	782,64	0,741	0,055
A-n34-k5	34	5	778	829,35	7,89	0,066	849,98	0,093	835,61	3,415	0,074
A-n36-k5	36	5	799	867,69	6,88	0,086	849,07	0,063	868,54	5,464	0,087
A-n37-k5	37	5	669	747,67	25,26	0,118	709,03	0,060	755,70	10,269	0,130
A-n38-k5	38	5	730	765,22	5,01	0,049	791,35	0,084	759,90	3,652	0,041
A-n39-k5	39	5	822	883,47	20,64	0,075	897,61	0,092	902,33	7,920	0,098
A-n44-k6	44	6	937	1038,11	12,30	0,108	1070,13	0,142	1042,23	20,239	0,112
A-n45-k6	45	6	944	1009,94	9,53	0,070	1664,40	0,763	1011,55	13,799	0,072
A-n45-k7	45	7	1146	1256,25	13,47	0,096	1516,20	0,323	1251,85	5,988	0,092
A-n46-k7	46	7	914	999,52	5,39	0,094	1275,81	0,396	989,43	9,372	0,083
A-n48-k7	48	7	1073	1192,49	8,62	0,111	4434,20	3,133	1190,89	9,361	0,110
GAP Médio				0,077			0,416		0,082		
Desvio Padrão do GAP Médio				0,028			0,841		0,026		

Ao observar os resultados apresentados na Tabela 7, o CPLEX obteve as mesmas quatro soluções ótimas encontradas pela abordagem híbrida, mas foi inferior a esta nas três últimas instâncias. A heurística somente atingiu o ótimo em dois casos, apresentando resultados inferiores ao híbrido nas demais situações testadas. Desta forma, para este conjunto P de instâncias, o híbrido se mostrou superior aos demais métodos.

Tabela 7 – Resultados das instâncias do conjunto P (* são valores ótimos).

Instância	n	k	Solução Literatura	Híbrido			CPLEX		Colônia de Formigas		
				Solução Média	Desvio Padrão	GAP	Solução Média	GAP	Solução Média	Desvio Padrão	GAP
P-n16-k8	16	8	450	451,34	0,000	0,003	451,34	0,003	451,34	0,000	0,003
P-n19-k2	19	2	212	212,66*	0,000	0,003	212,66*	0,003	212,66*	0,000	0,003
P-n20-k2	20	2	216	217,42*	0,000	0,007	217,42*	0,007	217,42*	0,000	0,007
P-n21-k2	21	2	211	212,71*	0,000	0,008	212,71*	0,008	217,48	0,000	0,031
P-n22-k2	22	2	216	217,85*	0,000	0,009	217,85*	0,009	218,37	1,101	0,011
P-n23-k8	23	8	529	531,17	0,000	0,004	533,83	0,009	531,24	0,148	0,004
P-n40-k5	40	5	458	480,91	9,058	0,050	481,66	0,052	503,51	2,894	0,099
P-n45-k5	45	5	510	573,73	5,029	0,125	605,29	0,187	569,68	6,674	0,117
GAP Médio				0,026			0,035		0,034		
Desvio Padrão do GAP Médio				0,043			0,064		0,047		

Um dado observado nestes testes diz respeito ao período gasto pelo híbrido para encontrar as soluções ótimas. O tempo gasto por este método para encontrar o ótimo foi inferior ao tempo de processamento registrado pelo CPLEX quando executado individualmente, conforme ilustra a Tabela 8. Após análise, pode-se notar que os resultados da Colônia de Formigas aplicados como base inicial para o *solver* contribuíram para garantir o valor ótimo de forma mais rápida do que quando executado separadamente.

Tabela 8 – Tempo de execução do CPLEX executado individualmente e em conjunto com o ACO (método híbrido) para encontrar o valor ótimo

Instância	CPLEX (Híbrido)	CPLEX
P-n19-k2	2h 32 min	2h 35 min
P-n20-k2	1h 1 min	1h 20 min
P-n21-k2	25 min	28 min
P-n22-k2	1h 1 min	1h 35min

No conjunto B, caracterizado pela distribuição clusterizada de seus clientes, o híbrido apresentou resultados melhores do que o CPLEX, mas não demonstrou melhorias significativas quando comparados a heurística, conforme ilustra a Tabela 9. Observando o *GAP Médio*, a eficiência do método proposto neste trabalho foi praticamente a mesma da apresentada pela ACO, da mesma forma que os desvios padrões são ligeiramente parecidos.

Tabela 9 – Resultados das instâncias do conjunto B.

Instância	n	k	Solução Literatura	Híbrido			CPLEX		Colônia de Formigas		
				Solução Média	Desvio Padrão	GAP	Solução Média	GAP	Solução Média	Desvio Padrão	GAP
B-n31-k5	31	5	672	706,66	0,32	0,052	703,49	0,047	705,787	2,218	0,050
B-n34-k5	34	5	788	816,08	19,49	0,036	839,00	0,065	833,310	2,455	0,058
B-n35-k5	35	5	955	992,51	2,29	0,039	1022,52	0,071	989,882	3,164	0,037
B-n38-k6	38	6	805	855,12	4,29	0,062	883,10	0,097	851,555	3,526	0,058
B-n41-k6	41	6	829	845,00	1,15	0,019	912,20	0,100	843,528	1,447	0,018
B-n43-k6	43	6	742	764,28	1,03	0,030	809,17	0,091	761,155	1,763	0,026
B-n44-k7	44	7	909	953,74	5,25	0,049	1034,26	0,138	946,524	4,819	0,041
B-n45-k5	45	5	751	797,90	3,90	0,062	941,53	0,254	797,501	2,825	0,062
B-n45-k6	45	6	678	715,69	11,42	0,056	958,28	0,413	711,219	6,001	0,049
GAP Médio				0,045			0,142		0,044		
Desvio Padrão do GAP Médio				0,015			0,119		0,015		

Analisando as instâncias da literatura para as quais ainda não foram encontrados os valores ótimos, presentes na Tabela 10, percebe-se que nenhum dos métodos estudados conseguiu ser superior aos valores extraídos da CVRPLIB.

O híbrido apresentou resultados bastante superiores ao CPLEX e, assim como observado no grupo B, seus resultados estiveram bastante próximos aos produzidos pela ACO.

Tabela 10 – Resultados das instâncias do grupo CMT.

Instância	n	k	Solução Literatura	Híbrido			CPLEX		Colônia de Formigas		
				Solução Média	Desvio Padrão	GAP	Solução Média	GAP	Solução Média	Desvio Padrão	GAP
CMT6	50	6	555,43	595,98	1,653	0,073	739,11	0,331	590,32	4,312	0,063
CMT7	75	11	909,68	991,78	2,292	0,090	3257,66	2,581	988,80	6,122	0,087
CMT8	100	9	865,95	1054,15	14,341	0,217	4475,50	4,168	1051,29	11,312	0,214
CMT14	100	11	866,37	970,95	5,680	0,121	5131,49	4,923	959,22	14,155	0,107
GAP Médio				0,125			3,001		0,118		
Desvio Padrão do GAP Médio				0,064			2,030		0,067		

Uma análise dos resultados do conjunto CMT mostrou que as dificuldades de melhoria da solução ocorreram principalmente durante a execução do CPLEX. Este perdeu eficiência no processamento das instâncias maiores, fato já notado nos testes iniciais.

6.3 Análise dos Resultados

Em uma primeira análise, foram verificados os resultados dos *gaps* produzidos pelos algoritmos. Estes dados foram plotados em gráficos *boxplot* para identificar possíveis diferenças estatísticas entre o Híbrido, o CPLEX e a Colônia de Formigas (ACO).

Como pode-se notar nas Figuras 17, 18, 19, 20 e 21, em todos os grupos de instâncias existe a interseção entre as caixas. Esta característica não permite identificar diferenças estatísticas significativas entre os dados obtidos. Ressalta-se que, nas figuras apresentadas, o número (1) corresponde ao Híbrido, o (2) ao CPLEX e o (3) à ACO.

Figura 17 – [Grupo E] Boxplot

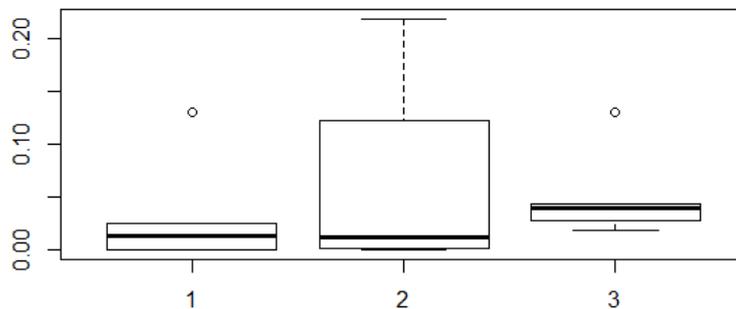


Figura 18 – [Grupo A] Boxplot

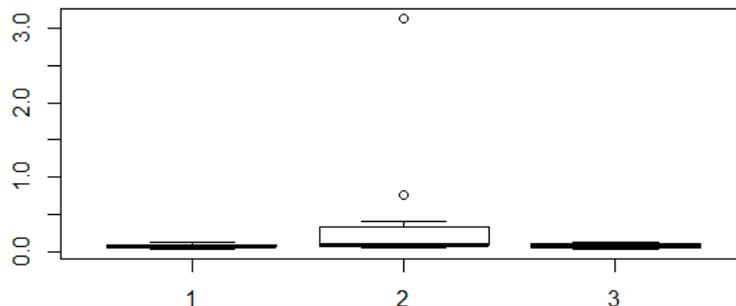


Figura 19 – [Grupo P] Boxplot

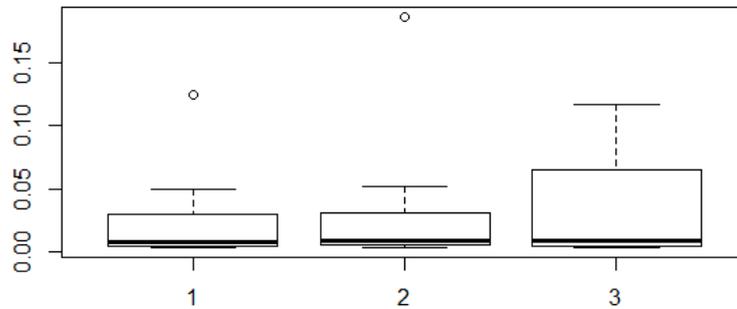


Figura 20 – [Grupo B] Boxplot

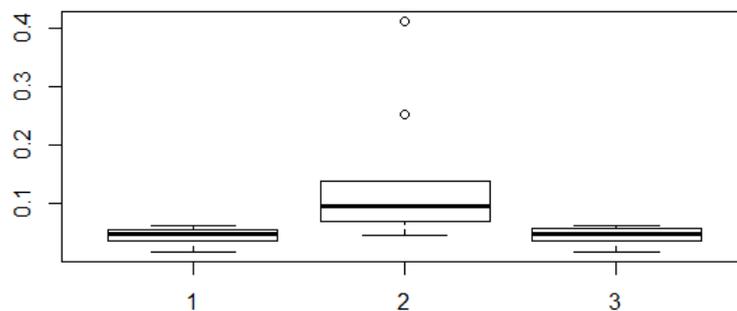
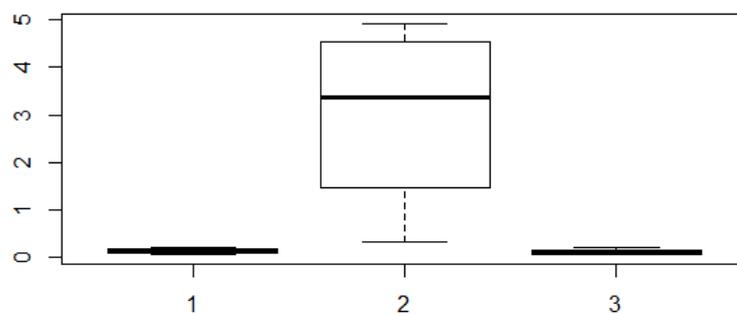


Figura 21 – [Grupo CMT] Boxplot



Desta forma é necessário realizar uma análise estatística mais profunda. Neste trabalho o teste estatístico escolhido para analisar os resultados dos *gaps* produzidos foi, inicialmente,

a Análise de Varância (ANOVA). Seu intuito é verificar se existem diferenças entre as médias de uma determinada variável em relação a um tratamento com dois ou mais níveis categóricos.

Para realizar tal análise é necessário validar três principais pressupostos, sendo eles a normalidade, a homocedasticidade e a independência sobre os erros.

Para trabalhar com o princípio da normalidade, deve-se verificar se os resultados apresentados seguem uma distribuição normal, o que inicialmente não era característica dos dados encontrados neste trabalho. Para solucionar este problema, aplicou-se a técnica de transformação *Box-Cox* para a normalização deles.

Em casos nos quais não foi possível aplicar a análise ANOVA, utilizou-se do teste não paramétrico de *Kruskal-Wallis*, que possui o mesmo objetivo da análise ANOVA.

6.3.1 Instâncias do Grupo E

Após a normalização, o teste de *Shapiro-Wilk* foi adotado para verificar a premissa de normalidade. Como resultado do teste obteve-se o valor de 0.001629, o qual é menor que o nível de significância $\alpha = 0.05$ e, portanto rejeita-se a hipótese nula de que a distribuição é normal.

Como os dados não fazem parte de uma distribuição normal, não será possível aplicar a análise ANOVA. Desta forma, foi adotado o teste não paramétrico de *Kruskal-Wallis* para melhor estudo dos resultados obtidos.

Este teste não paramétrico, apresentado na Figura 22, revelou um valor de *p-value* igual a 0.3903, que é superior ao nível de significância de 0.05.

Figura 22 – [Grupo E] Teste de *Kruskal-Wallis*

```
kruskal-wallis rank sum test
data: Metodos by Instancias
kruskal-wallis chi-squared = 1.8818, df = 2, p-value = 0.3903
```

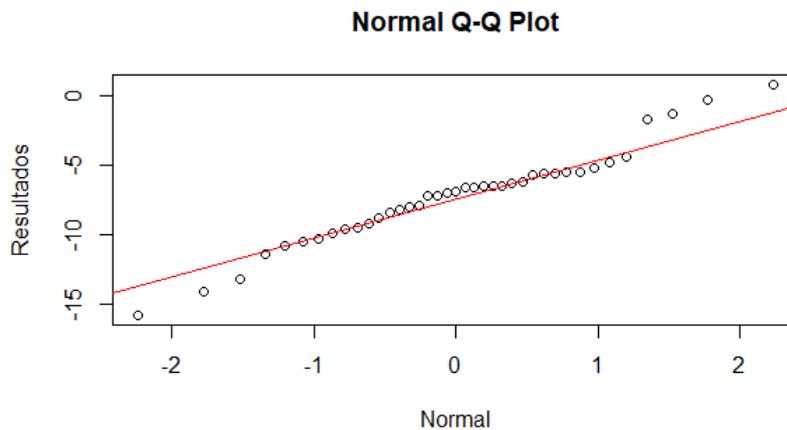
Este resultado indica que não há provas suficientes de que os dados deste conjunto de instâncias possuam diferenças estatísticas significativas.

6.3.2 Instâncias do Grupo A

O teste de *Shapiro-Wilk* foi adotado para verificar a premissa de normalidade, e teve como resultado o valor de 0.3181. Este é maior que o nível de significância $\alpha = 0.05$, levando a

conclusão de que os dados fazem parte de uma distribuição normal, conforme ilustra o gráfico *Qqplot* presente na Figura 23.

Figura 23 – [Grupo A] Teste de Normalidade



Para verificar a premissa de homocedasticidade foi utilizado o teste de *Fligner-Killeen*, para o qual este conjunto de instâncias teve como resultado 0.5662. Tal valor é superior a 0.05 e, portanto, não pode-se rejeitar a hipótese nula de que as variâncias deste grupo são iguais.

Após aceitar a segunda premissa, parte-se para a terceira e última validação, correspondente à independência dos dados do conjunto A de instâncias. Para este caso foi realizado o teste de *Durbin-Watson*, cujo resultado encontrado foi 0.004886, que é inferior ao nível de significância $\alpha = 0.05$. Desta forma rejeita-se a hipótese nula de que os dados são independentes.

Devido a falha no teste de Independência, os dados não foram aceitos em todos pressupostos e, portanto, não será possível aplicar a análise ANOVA. Desta forma, foi adotado o teste não paramétrico de *Kruskal-Wallis* para melhor estudo dos resultados obtidos.

Este teste não paramétrico revelou um valor de *p-value* igual a 0.1213, observado na Figura 24, o qual é superior ao nível de significância de 0.05, indicando que não há provas suficientes de que os dados deste conjunto de instâncias possuam diferenças estatísticas significativas.

Figura 24 – [Grupo A] Teste de *Kruskal-Wallis*

```
kruskal-wallis rank sum test
data: Metodos by Instancias
kruskal-wallis chi-squared = 4.2195, df = 2, p-value = 0.1213
```

6.3.3 Instâncias do Grupo P

Após a normalização, o teste de *Shapiro-Wilk* foi adotado para verificar a premissa de normalidade. Como resultado do teste obteve-se o valor de 0.01132, o qual é menor que o nível de significância $\alpha = 0.05$ e, portanto, conclui-se que os dados não fazem parte de uma distribuição normal.

Por não atender a premissa de Normalidade, será aplicado o teste não paramétrico de *Kruskal-Wallis*. Este teve como resultado um valor de *p-value* igual a 0.9329, que é superior ao nível de significância de 0.05, podendo ser observado na Figura 25.

Figura 25 – [Grupo P] Teste de *Kruskal-Wallis*

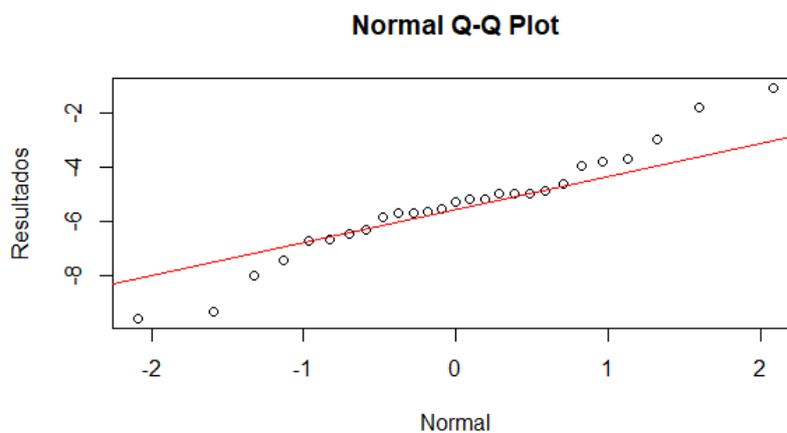
```
kruskal-wallis rank sum test
data: Metodos by Instancias
kruskal-wallis chi-squared = 0.13897, df = 2, p-value = 0.9329
```

Com este resultado podemos concluir que não há provas suficientes de que os dados deste conjunto de instâncias possuam diferenças estatísticas significativas.

6.3.4 Instâncias do Grupo B

O teste de *Shapiro-Wilk* para o grupo B de instâncias teve como resultado 0.4834, que é maior que o nível de significância $\alpha = 0.05$. Desta forma, aceita-se a hipótese nula de que a distribuição normal, conforme ilustra o gráfico *Qqplot* presente na Figura 26.

Figura 26 – [Grupo B] Teste de Normalidade



Para verificar a premissa de homocedasticidade foi utilizado o teste de *Fligner-Killeen*, para o qual este conjunto de instâncias teve como resultado 0.9296. Tal valor é superior a 0.05 e, portanto, não pode-se rejeitar a hipótese nula de que as variâncias deste grupo são iguais.

Após aceitar a segunda premissa, parte-se para a terceira e última validação, correspondente à independência dos dados do conjunto B de instâncias. Para este caso foi realizado o teste de *Durbin-Watson*, cujo resultado encontrado foi 0.04938, que é inferior ao nível de significância $\alpha = 0.05$. Logo, rejeita-se a hipótese nula de que os dados são independentes.

Como os dados não foram aceitos no pressuposto de Independência, não será possível aplicar a análise ANOVA. Desta forma, foi adotado o teste não paramétrico de *Kruskal-Wallis* para melhor estudo dos resultados obtidos.

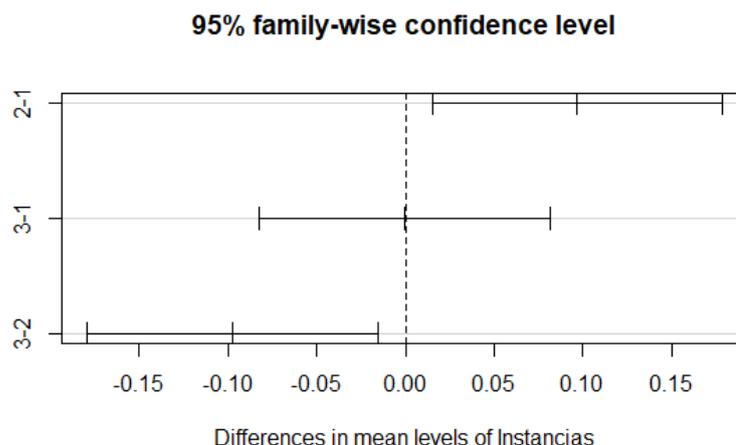
Este teste não paramétrico revelou um valor de *p-value* igual a 0.001253, que é inferior ao nível de significância de 0.05, demonstrando que os dados deste conjunto de instâncias apresentam diferenças estatísticas significativas. Tal resultado pode ser observado na Figura 27.

Figura 27 – [Grupo B] Teste de *Kruskal-Wallis*

```
kruskal-wallis rank sum test
data: Metodos by Instancias
kruskal-wallis chi-squared = 13.364, df = 2, p-value = 0.001253
```

O teste de Tukey foi utilizado para verificar onde estas diferenças se encontram. Os resultados demonstram que estas estão no CPLEX com o Híbrido, e na ACO com o CPLEX, conforme ilustram as Figuras 28 e 29. Neste caso, o número (1) corresponde ao Híbrido, o (2) ao CPLEX e o (3) à ACO.

Figura 28 – Representação gráfica do teste de Tukey para o grupo de instâncias B



Na Figura 29, pode-se comprovar através do $p_{adj} < 0,05$, que os pares [2-1] e [3-2] possuem diferenças estatísticas entre eles.

Figura 29 – Resultado do teste de Tukey para o grupo de instâncias B

	diff	lwr	upr	p adj
2-1	0.0967777778	0.01493347	0.17862209	0.0183452
3-1	-0.0006666667	-0.08251098	0.08117765	0.9997719
3-2	-0.0974444444	-0.17928876	-0.01560013	0.0175114

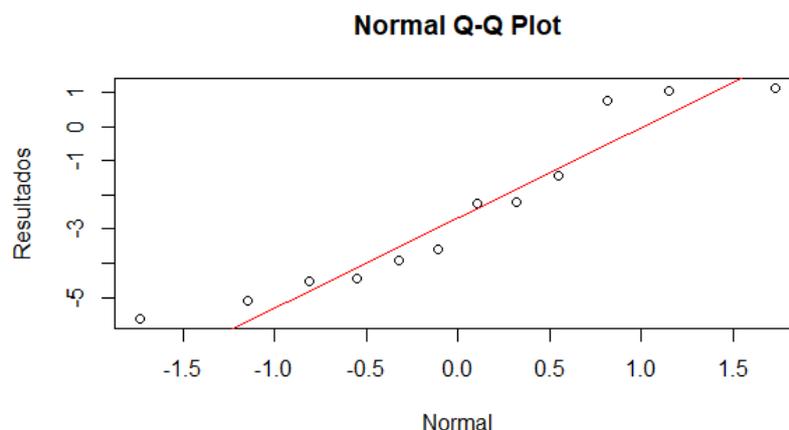
O teste de Tukey informa que as diferenças significativas estão entre o CPLEX e a ACO, e entre o CPLEX e o Híbrido. Ao analisar os seus valores do *diff* podemos definir qual estratégia é superior à outra. No par [2-1], que corresponde ao [CPLEX - Híbrido], temos um *diff* positivo de 0.0967777778, o que significa que o Híbrido foi melhor do que o CPLEX. No par [3 - 2], que diz respeito à [ACO - CPLEX], o valor do *diff* é negativo, levando a conclusão de que a heurística Colônia de Formigas é superior ao CPLEX.

Desta forma, pode-se concluir que os resultados do CPLEX são inferiores aos encontrados pela ACO e pelo Híbrido. Também pode-se afirmar que o Híbrido e a heurística Colônia de Formigas não apresentaram resultados estaticamente melhores em relação um ao outro, uma vez que $op\ adj > 0,05$.

6.3.5 Instâncias do Grupo CMT

O teste de *Shapiro-Wilk* para o grupo CMT de instâncias teve como resultado 0.1495, que é maior que o nível de significância $\alpha = 0.05$. Desta forma, aceita-se a hipótese nula de que a distribuição normal, conforme ilustra o gráfico *Qqplot* presente na Figura 30.

Figura 30 – [Grupo CMT] Teste de Normalidade



Para verificar a premissa de homocedasticidade foi utilizado o teste de *Fligner-Killeen*, para o qual este conjunto de instâncias teve como resultado 0.719. Tal valor é superior a 0.05 e, portanto, não pode-se rejeitar a hipótese nula de que as variâncias deste grupo são iguais.

Após aceitar a segunda premissa, parte-se para a terceira e última validação, correspon-

dente à independência dos dados do conjunto CMT de instâncias. Para este caso foi realizado o teste de *Durbin-Watson*, cujo resultado encontrado foi 0.06811, que é superior ao nível de significância $\alpha = 0.05$. Desta forma, não rejeita-se a hipótese nula de que os dados são independentes.

Neste caso, todos os dados foram aceitos em todas as premissas testadas e, portanto, poderão ser submetidos a análise ANOVA. Esta indicou um *p-value* < 0.05 , conforme ilustra a tabela da Figura 31. Resalta-se que: SQ = soma dos quadrados, *gl* = graus de liberdade, MQ = média dos quadrados, F = distribuição F. O valor de *p-value* é menor do que $\alpha = 0.05$, indicando que os resultados apresentados possuem diferenças estatísticas.

Figura 31 – Resultado ANOVA

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Instancias	2	50.26	25.129	15.01	0.00136 **
Residuals	9	15.07	1.675		

O teste de Tukey foi utilizado para verificar onde estas diferenças se encontram. Seus resultados demonstram que estas se encontram entre o CPLEX e o Híbrido, e entre a ACO e o CPLEX, conforme ilustra a Figura 32.

Figura 32 – Resultado do teste de Tukey para o grupo de instâncias CMT

	diff	lwr	upr	p adj
2-1	4.212866	1.658084	6.767649	0.0032941
3-1	-0.246278	-2.801061	2.308505	0.9610253
3-2	-4.459144	-7.013927	-1.904361	0.0022703

O teste de Tukey informa que as diferenças significativas estão entre o CPLEX e a ACO, e entre o CPLEX e o Híbrido. Ao analisar os seus valores do *diff* podemos definir qual estratégia é superior à outra. No par [2-1], que corresponde ao [CPLEX - Híbrido], temos um *diff* positivo de 4.212866, o que significa que o Híbrido foi melhor do que o CPLEX. No par [3 - 2], que diz respeito à [ACO - CPLEX], o valor do *diff* é negativo, levando a conclusão de que a heurística Colônia de Formigas é superior ao CPLEX.

Desta forma, pode-se concluir que os resultados do CPLEX são inferiores aos encontrados pela ACO e pelo Híbrido. Também pode-se afirmar que o Híbrido e a heurística Colônia de Formigas não apresentaram resultados estaticamente melhores em relação um ao outro, uma vez que $op\ adj > 0,05$.

Capítulo 7

Conclusão

O presente trabalho apresentou uma abordagem híbrida para resolver o Problema de Roteamento de Veículos Capacitados. Este compreende a obtenção de um conjunto de rotas, que devem ser percorridas por uma frota de veículos homogêneos, atendendo assim as demandas de um conjunto de clientes. Seu objetivo é a minimização do custo total das rotas, sabendo-se que elas devem iniciar e terminar no depósito central, cada cliente somente pode ser atendido uma única vez por um único veículo, respeitando sua capacidade.

A metodologia apresentada combinou duas técnicas, com características bastantes distintas, com objetivo desenvolver uma estratégia eficiente para resolver o problema, tendo em vista que o custo a ser minimizado compreendia a distância total percorrida.

Neste trabalho foi possível identificar as falhas e os pontos fortes de cada metodologia. A heurística possui um desempenho melhor do que o CPLEX à medida que o tamanho do problema aumenta, mas, em contrapartida, seus resultados tendem a estagnar rapidamente. O *solver* da IBM mostrou resultados satisfatórios nas instâncias menores, mas perdeu eficiência, considerando a qualidade da solução no mesmo tempo computacional, nos casos em que o número de clientes cresceram.

O método proposto procurou abstrair as deficiências do CPLEX e da ACO para assim, gerar resultados melhores dos que encontrados quando estes foram executados separadamente.

Os testes mostraram que a heurística interferiu positivamente no desempenho do CPLEX, que, a partir das soluções factíveis geradas por esta, produziu melhores resultados do que quando executado individualmente.

Pelas análises estatísticas pode-se inferir que o CPLEX, limitando seu tempo de execução em 10 horas, tem desempenho inferior à heurística Colônia de Formigas, assim como apresenta resultados piores quando comparado ao Híbrido. Conclui-se também que não

houve diferenças estatísticas significativas nos resultados produzidos entre o híbrido e a heurística ACO.

7.1 Trabalhos Publicados

Os resultados apresentados neste trabalho foram publicados no:

- XIX ENMC - Encontro Nacional de Modelagem Computacional (Artigo completo).
- XLVIII Simpósio Brasileiro de Pesquisa Operacional (Pôster).

7.2 Trabalhos Futuros

Como trabalhos futuros, deseja-se desenvolver as seguintes etapas:

- Trabalhar com outra heurística no algoritmo híbrido.
- Trabalhar outras opções do otimizador CPLEX para que instâncias maiores possam ser testadas.
- Realizar uma análise estatística mais completa dos resultados encontrados.

Apêndices

APÊNDICE A – O Processo de Híbridização

Para que a comunicação entre os métodos ocorra de forma eficiente é necessário alimentar o software com dois diferentes arquivos, sendo um deles, com extensão *.lp*, composto pelo problema e seus dados mapeados de acordo com o modelo matemático com o qual está trabalhando. O segundo, com formato *.mst*, possui os dados da melhor solução transcritos de uma forma que esta possa ser interpretada pelo CPLEX.

O arquivo com extensão *.lp* possui as mesmas características presentes em qualquer problema de programação linear, com as variáveis, o objetivo e as suas restrições (Figura 33).

Fonte: IBM Knowledge Center

```
Maximize
obj: x1 + x2 + x3 + x4
Subject To
c1: - x1 + x2 + x3 + 10 x4 <= 20
c2: x1 - 3 x2 + x3 <= 30
c3: x2 - 3.5 x4 = 0
Bounds
0 <= x1 <= 40
2 <= x4 <= 3
General
x4
End
```

Figura 33 – Estrutura de um arquivo com extensão *.lp*

O *.mst* é um arquivo com declarações *xml* composto pela enumeração de todas as variáveis do problema e suas respectivas atribuições, sendo 0, para aquelas que não estão ativas, ou 1, para aquelas que fazem parte da solução (Figura 34).

```

<?xml version = "1.0" encoding="UTF-8" standalone="yes"?>
<CPLEXSolutions version="1.2">
<CPLEXSolution version="1.2">
  <header
    problemName="IloCplex"
    solutionName="m1"
    solutionIndex="0"
    MIPStartEffortLevel="0"
    writeLevel="2"/>
  <variables>
    <variable index="0" value="0"/>
    <variable index="1" value="0"/>
    <variable index="2" value="0"/>
    <variable index="3" value="0"/>
    <variable index="4" value="0"/>
    <variable index="5" value="0"/>
    <variable index="6" value="0"/>
    <variable index="7" value="0"/>
    <variable index="8" value="0"/>
    <variable index="9" value="0"/>
    <variable index="10" value="0"/>
  </variables>
</CPLEXSolution>
</CPLEXSolutions>

```

Figura 34 – Estrutura de um arquivo com extensão *.mst*

O software da IBM possui um funcionamento padrão, constituído das seguintes etapas:

- (i) Realizar um pré-processamento para simplificar ou reduzir o modelo;
- (ii) Procurar uma solução factível;
- (iii) Buscar a solução ótima.

Contudo, ao utilizar a estratégia *MIP start* - adotada neste trabalho -, uma base factível já é repassada ao problema e, desta forma, a etapa (ii) não é realizada. Ao realizar a leitura do arquivo *.lp*, deve-se, logo em seguida, executar a leitura do arquivo *.mst*. Neste momento o CPLEX irá parear as informações de variável/valor e realizar uma interpretação linha a linha, na ordem em que estas aparecerem. Na prática, cada linha do arquivo de solução será associada a uma variável presente no arquivo do modelo, conforme ilustra a Figura 35.

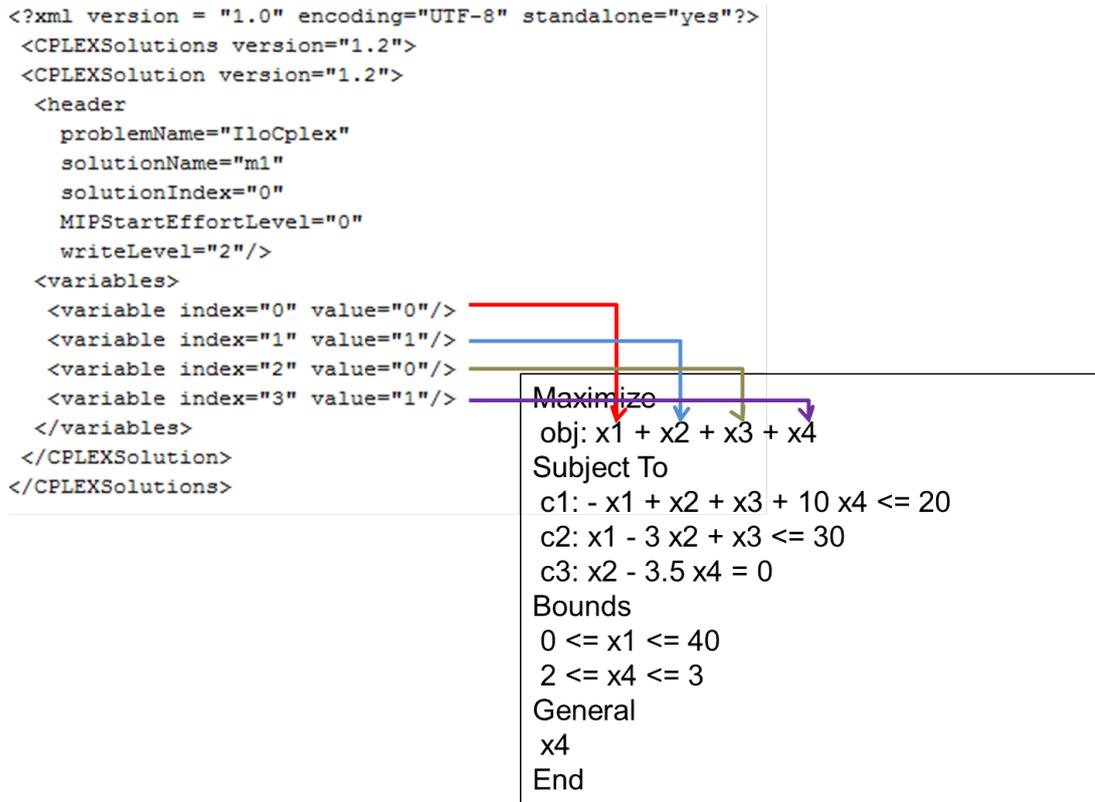


Figura 35 – Interpretação dos arquivos *.lp* e *.mst*

A principal intenção ao utilizar esta estratégia é que o software da IBM consiga progredir satisfatoriamente na solução, gerando resultados melhores dos que os produzidos pelos métodos separadamente, assim como pretende-se que eles estejam o mais próximo possível dos encontrados na literatura.

O arquivo *.mst*, utilizado na estratégia *MIP Start*, é produzido com base na modelagem de Junqueira (2013), sendo sua estrutura observada na Figura 36. Para que ele seja construído são necessárias algumas informações disponíveis em um arquivo *.txt*, o qual é gerado pela ACO.

```

for (k=1;k<=n;k++){
  for (l=1;l<=n;l++){
    for (t=1;t<=n;t++){
      for (v=0;v<numVeic;v++){
        printf("<variable index="0" value="1">
      }
    }
  }
}

```

// Saindo do cliente **k**
 // Para o cliente **l**
 // No instante **t**
 // Usando o veículo **v**

1 => Verdadeiro
 0 => Falso

Figura 36 – Parte do processo de criação do arquivo *.mst*

Este arquivo *.txt*, exibido na Figura 37, conterá os dados da melhor formiga, sendo indispensável para que o CPLEX dê continuidade ao processo de otimização. As informações nele presentes correspondem ao número de clientes a serem atendidos, o número de veículos usados na rota, e a ordem de visitação do roteiro.

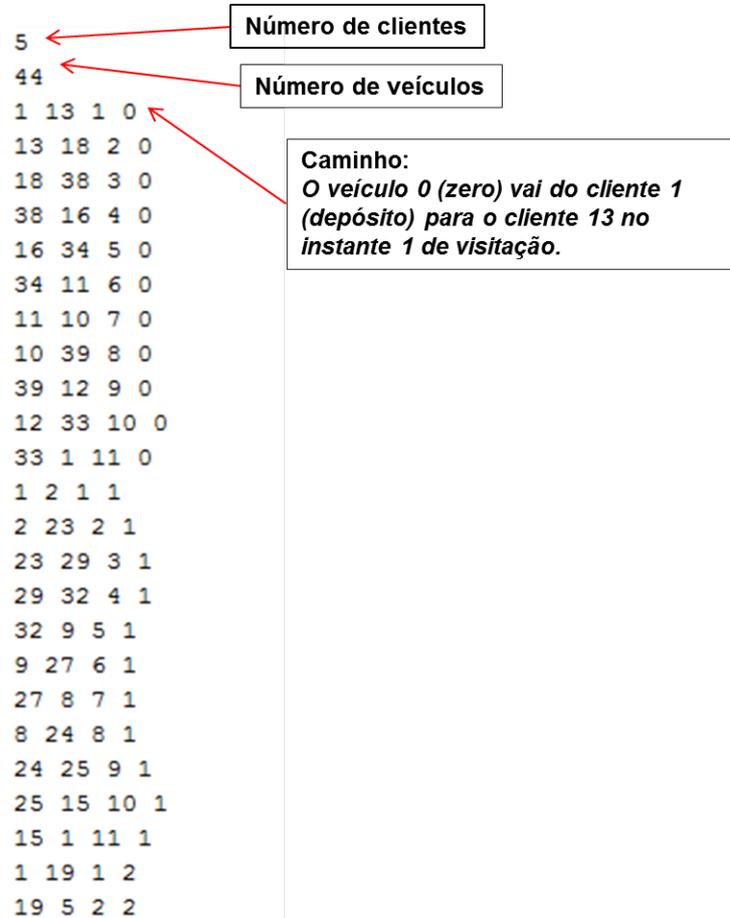


Figura 37 – Arquivo *.txt* gerado pela ACO contendo os dados da melhor formiga

O arquivo *.lp* também é criado com base no modelo matemático de Junqueira (2013), e utiliza dos mesmos recursos de programação do arquivo *.mst*, contudo, os dados necessários para sua geração são retirados da instância obtida da literatura.

Com estes arquivos contruídos, uma chamada de sistema é realizada para inicialização do CPLEX, o qual executará os comandos de abertura e interpretação do *.lp* e *.mst*, para, posteriormente, começar a otimizar a solução repassada.

A Figura38 demonstra como este processo é perceptível ao usuário, apresentando o comportamento do CPLEX após a leitura dos arquivos *.lp* e *.mst*. Após a heurística repassar os dados da melhor rota (1), os arquivos de integração são gerados de maneira transparente ao usuário, o qual é posteriormente apresentado a tela do CPLEX. Neste momento (2), de forma automática, o arquivo com o problema modelado, seguido do arquivo com a solução, são lidos e seus dados associados. Seguindo seu comportamento padrão, o software da

IBM realiza o pré-processamento (3) para simplificar o modelo inserido, e então parte para a otimização da solução repassada pela heurística (4).

```

CPLEX> Problem 'modelo.lp' read.
Read time = 0.16 sec. (4.45 ticks)
CPLEX> MIP start file 'solucao.mst' read.
CPLEX> New value for time limit in seconds: 3600
CPLEX> 1 of 1 MIP starts provided solutions.
MIP start 'm1' defined initial solution with objective 402.8338.
Tried aggregator 1 time.
MIP Presolve eliminated 0 rows and 3679 columns.
MIP Presolve modified 67036 coefficients.
Reduced MIP has 1814 rows, 38913 columns, and 484065 nonzeros.
Reduced MIP has 38913 binaries, 0
Presolve time = 0.59 sec. (303.46
Tried aggregator 1 time.
Reduced MIP has 1814 rows, 38913 c
Reduced MIP has 38913 binaries, 0
Presolve time = 0.16 sec. (75.58 ticks)
Probing time = 0.16 sec. (27.36 ticks)
Clique table members: 112404.
MIP emphasis: balance optimality and feasibility.
MIP search method: dynamic search.
Parallel mode: deterministic, using up to 4 threads.
Root relaxation solution time = 1.75 sec. (1200.40 ticks)

```

Nodes		Objective	IInf	Best Integer	Cuts/ Best Bound	ItCnt	Gap
Node	Left						
*	0+	0		402.8338	0.0000		100.00%
	0	0	231.4489	435	402.8338	231.4489	20 42.54%

Figura 38 – Uma solução repassada pela ACO sendo interpretada pelo CPLEX

Ao final do seu processamento, o CPLEX exporta os dados da solução por ele encontrada em um arquivo .txt, conforme ilustrado na Figura 39.

Variable Name	Solution Value
d1_7_1_0	1.000000
d1_9_1_1	1.000000
d1_17_1_2	1.000000
d1_19_1_3	1.000000
d1_22_1_4	1.000000
d1_24_1_5	1.000000
d1_37_1_6	1.000000
d2_1_8_3	1.000000
d3_42_4_2	1.000000
d4_38_7_0	1.000000
d5_25_8_1	1.000000
d6_2_7_3	1.000000
d7_23_2_0	1.000000
d8_1_9_0	1.000000
d9_40_2_1	1.000000
d10_16_7_6	1.000000
d11_29_7_4	1.000000
d12_43_6_1	1.000000
d13_1_3_5	1.000000
d14_47_4_4	1.000000
d15_1_7_2	1.000000
d16_28_8_6	1.000000
d17_48_2_2	1.000000
d18_15_6_2	1.000000

Caminho:
 O veículo 0 sai do cliente 1 (depósito) para o cliente 7 no instante 1 de visitação.

Figura 39 – Arquivo .txt produzido pelo CPLEX

No tipo de hibridização caracterizada pelo processo de realimentação, os dados deste arquivo *.txt* serão usados pela heurística para dar continuidade ao processo de melhoria da solução. Estes dados são lidos, interpretados e inseridos em uma formiga, a qual é inserida na colônia, conforme foi mostrado anteriormente na Figura 10b. Após isso, o funcionamento do algoritmo híbrido procederá da mesma forma que ocorre no processo sem a realimentação, com a heurística exportando seus dados ao CPLEX.

Referências

- ARAÚJO, H. A. **Algoritmo Simulated Annealing: Uma Nova Abordagem**. 2001. 102 p. Dissertação (Pós Graduação em Ciência da Computação) — Universidade Federal de Santa Catarina, Florianópolis, (2001). Citado na página 11.
- AUGERAT, P. et al. Computational results with a branch and cut code for the capacitated vehicle routing problem. **Tech. Rep. 949-M**, Université Joseph Fourier, (1995). Citado na página 33.
- BEASLEY, J. E. Route-first cluster-second methods for vehicle routing. **Omega**, v. 11, p. 403–408, (1983). Citado na página 12.
- BELL, J. E.; MCMULLEN, P. An ant colony optimization techniques for the vehicle routing problem. **Advanced Engineering Informatics**, v. 18, p. 41–48, (2004). Citado na página 13.
- BITTENCOURT, G. C. **Modelagem E Implementação De Um Sistema Computacional Para A Solução De Um Problema De Roteamento De Veículos (PRV) Com O Uso Da Metaheurística Busca Dispersa (Scatter Search)**. Novembro 2010. 50 p. Monografia (Graduação em Engenharia de Produção) — Universidade Federal de Juiz de Fora, Juiz de Fora, (2010). Citado na página 7.
- BITTENCOURT, G. C. et al. Problema de roteamento de veículos capacitados (PVRC): Solução manual x Busca dispersa. In: CONGRESO LATINO-IBEROAMERICANO DE INVESTIGACIÓN OPERATIVA. Rio de Janeiro, (2012). Citado na página 12.
- CHAVES, A. A. **Modelagens Exata e Heurística para Resolução do Problema do Caixaieiro Viajante com Coleta de Prêmios**. 2003. 49 p. Monografia (Graduação em Ciência da Computação) — Departamento de Ciência da Computação, Universidade Federal de Ouro Preto, Ouro Preto, (2003). Citado na página 2.
- CHEN, S.; GOLDEN, B.; WASIL, E. The split delivery vehicle routing problem: Applications, algorithms, test problems, and computational results. **Networks**, v. 49, p. 318–329, (2007). Citado na página 12.
- CHRISTOFIDES, N.; EILON, S. An algorithm for the vehicle-dispatching problem. **OR**, Operational Research Society, v. 20, n. 3, p. 309–318, (1969). Citado na página 33.
- CHRISTOFIDES, N.; MINGOZZI, A.; TOTH, P. Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. **Mathematical Programming**, v. 20, p. 255–282, (1981). Citado 2 vezes nas páginas 2 e 10.
- CLARKE, G.; WRIGHT, J. W. Scheduling of vehicles from a central depot to a number of delivery points. **Operations Research**, v. 12, p. 568–581, (1964). Citado 3 vezes nas páginas 2, 12 e 13.
- CORREA, R. S.; NETO, M. S. Cadeia de suprimentos - a importância da escolha do método de custeio na gestão dos custos logísticos. In: **Anais do IV Congresso de Pós Graduação da Unimep**. São Paulo: Editora Unimep, (2006). p. 1–10. Citado na página 1.

CROES, G. A. A method for solving traveling salesman problems. **Operations Research**, v. 6, p. 791–812, novembro/dezembro (1958). Citado na página 12.

DANTZIG, G. B.; RAMSER, J. H. The truck dispatching problem. **Management Science**, v. 6, p. 80–91, out. (1959). Citado 2 vezes nas páginas 7 e 8.

DILSON, P. L. **Heurísticas E Algoritmo Exato Para o Problema De Roteamento De Veículos Com Coleta E Entrega Simultâneas**. Fevereiro 2010. 83 p. Dissertação (Mestrado em Ciência da Computação) — Universidade Federal de Minas Gerais, Belo Horizonte, (2010). Citado na página 8.

DORIGO, M. et al. **Positive Feedback as a Search Strategy**. [S.l.], (1991). Citado na página 17.

DORIGO, M.; STÜTZLE, T. **Ant Colony Optimization**. [S.l.]: Bradford Company, (2004). ISBN 0262042193. Citado na página 11.

DORNER, K. F.; HARTL, R. F.; LUCKA, M. A parallel version of the D-Ant algorithm for the Vehicle Routing Problem. In: VAJTERISC, M. et al. (Ed.). **Parallel Numerics**. [S.l.]: Springer, (2005). p. 109–118. Citado na página 13.

ESTEVAM, J. B. **Heurísticas para o problema de roteamento de veículos capacitados - PRVC Visando aplicação no Gerenciamento da Cadeia de Suprimentos**. 2003. 117 p. Monografia (Graduação em Ciência da Computação) — Departamento de Ciência da Computação, Universidade Federal de Lavras, Lavras, (2003). Citado 2 vezes nas páginas 9 e 11.

FERREIRA, F. M.; BACHEGA, S. J. Programação linear: Um estudo de caso sobre os custos de transportes em uma empresa de setor de confecções de catalão-GO. In: XXXI ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO. Belo Horizonte, (2011). Citado 2 vezes nas páginas 5 e 19.

FERREIRA, F. S.; MONTEIRO, G. D.; TEIXEIRA, O. N. Colônia evolucionária de formigas: Uma proposta inicial aplicada ao problema do caixeiro viajante. **Revista Hífen**, v. 32, p. 286–292, (2008). Citado na página 17.

FUKASAWA, R. et al. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. **Mathematical Programming**, v. 106, p. 491–511, (2006). Citado 2 vezes nas páginas 2 e 10.

GALVÃO, R. D. et al. Roteamento de veículos com base em sistemas de informação geográfica. **Gestão e Produção**, v. 4, p. 159–174, (1997). Citado na página 12.

GASPARIN, E. **Combinação de Métodos Heurísticos na Resolução de um Problema de Roteamento de Veículos Capacitados**. 2014. 58 p. Monografia (Graduação em Engenharia de Produção) — Universidade Tecnológica Federal do Paraná, Medianeira, (2014). Citado na página 13.

GILLET, B. E.; JOHNSON, J. G. Multi-terminal vehicle-dispatch algorithm. **Elsevier**, v. 4, p. 711–718, (1976). Citado na página 13.

GILLET, B. E.; MILLER, L. R. A heuristic algorithm for the vehicle dispatch problem. **Operations Research**, v. 22, p. 341–349, (1974). Citado 3 vezes nas páginas 2, 12 e 13.

- GLOVER, F. Future paths for integer programming and links to artificial intelligence. **Decision Sciences** 8, p. 156–166, (1977). Citado na página 11.
- GLOVER, F. Future paths for integer programming and links to artificial intelligence. **Computers and Operations Research - Special issue: Applications of integer programming**, v. 13, p. 533–549, maio (1986). Citado na página 12.
- GOLDBARG, M. C.; LUNA, H. P. L. **Otimização Combinatória e Programação Linear**. São Paulo: Elsevier, (2000). Citado na página 6.
- GOLIATT, P. V. Z.; ANGELO, J. S.; BARBOSA, H. J. C. **Manual de Computação Evolutiva**. Minas Gerais: Editora UFMG, (2013). Citado na página 15.
- GOSS, S. et al. Self-organized shortcuts in the argentine ant. **Naturwissenschaften**, v. 76, p. 579–581, (1989). Citado 2 vezes nas páginas 15 e 16.
- GUIMARÃES, T. H. **Uma nova abordagem heurística para a resolução do problema do roteamento de veículos capacitados com restrições tridimensionais de carregamento**. 2011. 108 p. Dissertação (Pós Graduação em Métodos Numéricos em Engenharia) — Universidade Federal do Paraná, Curitiba, (2011). Citado na página 1.
- JUNQUEIRA, L. **Modelos e algoritmos para problemas integrados de roteamento e carregamento de veículos**. Maio 2013. 228 p. Tese (Doutorado em Engenharia de Produção) — Universidade Federal de São Carlos, São Paulo, (2013). Citado 5 vezes nas páginas 1, 8, 22, 61 e 62.
- KIRKPATRICK, S.; GELATT, C. D.; VECCH, M. P. Optimization by simulated annealing. **Science**, v. 220, p. 671–680, (1983). Citado 2 vezes nas páginas 11 e 12.
- LI, X.; TIAN, P. An ant colony system for the open vehicle routing problem. In: VAJTERISC, M. et al. (Ed.). **Ant Colony Optimization and Swarm Intelligence**. [S.l.]: Springer, (2006). v. 4150, p. 356–363. Citado na página 13.
- LIMA, S. J.; SANTOS, R. A. R.; ARAÚJO, S. A. Otimização do problema de roteamento de veículos capacitado usando algoritmos genéticos e as heurísticas de gillett e miller e descida de encosta. **XXXV Encontro Nacional De Engenharia De Produção**, (2015). Citado 2 vezes nas páginas 2 e 13.
- LIN, S. Computer solutions of the traveling salesman problem. **The Bell System Technical Journal**, v. 44, p. 2245–2269, (1965). Citado na página 13.
- MARINS, F. A. S. **Introdução a Pesquisa Operacional**. São Paulo: Cultura Acadêmica, (2011). Citado na página 19.
- MARTEL, A.; VIEIRA, D. R. **Análise e projeto de redes logísticas**. [S.l.]: Editora Saraiva, (2008). Citado na página 6.
- MAZZEO, S.; LOISEAU, I. An ant colony algorithm for the capacitated vehicle routing. **Electronic Notes in Discrete Mathematics**, v. 18, p. 181–186, (2004). Citado na página 13.

MELQUIADES, J. A. R. **Modelagem para a roteirização do processo de coleta e transporte de resíduos sólidos urbanos**. Fevereiro 2011. 95 p. Tese (Doutorado em Métodos Numéricos em Engenharia – Área de concentração: Programação Matemática, Setores de Tecnologia e Ciências Exatas) — Universidade Federal do Paraná, Curitiba, (2011). Citado na página 6.

MIURA, M. **Resolução de um problema de Roteamento de veículos em uma empresa transportadora**. (2003). 89 p. Monografia (Graduação em Engenharia de Produção) — Escola Politécnica, Universidade de São Paulo, São Paulo, 2003. Citado 2 vezes nas páginas 7 e 8.

MOREIRA, D. A. **Pesquisa Operacional: Curso Introductório**. São Paulo: Cengage Learning, (2007). Citado na página 5.

MULLER, A. et al. Caracterização da rede de transportes intermunicipal em Guarapuava/PR. In: II SIMPÓSIO DE ESTUDOS URBANOS, 2013. Paraná, (2013). Citado na página 6.

NETO, F. F.; JÚNIOR, M. K. Logística empresarial. **Revista Gestão Empresarial**, v. 1, p. 39–49, (2002). Citado na página 1.

NETO, R. F. T.; FILHO, M. G. Otimização por colônia de formigas para o problema de sequenciamento de tarefas em uma única máquina com terceirização permitida. **SciELO**, v. 20, p. 76–86, (2013). Citado 2 vezes nas páginas 16 e 17.

NÉIA, S. S. et al. Roteamento de veículos utilizando otimização por colônia de formigas e algoritmo genético. In: LOPES, H. S.; RODRIGUES, L. C. A.; STEINER, M. T. A. (Ed.). **Meta-Heurísticas em Pesquisa Operacional**. 1. ed. [S.l.]: Omnipax, (2013). p. 219–236. Citado na página 13.

OLIVEIRA, R. A. C.; DELGADO, K. V. Sistema para roteamento de veículos capacitados aplicando métodos de monte carlo. In: XI SIMPÓSIO BRASILEIRO DE SISTEMAS DE INFORMAÇÃO. Goiânia, (2015). Citado 3 vezes nas páginas 2, 13 e 14.

OSMAN, I. H. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. In: OPERATION RESEARCH. [S.l.], (1993). p. 421–451. Citado na página 12.

PEARL, J. **Heuristics: Intelligent Search Strategies for Computer Problem Solving**. Boston: Addison-Wesley Longman Publishing Co., Inc., (1984). Citado na página 13.

PIZZOLATO, N.; GANDOLPHO, A. **Técnicas de otimização**. [S.l.]: LTC, (2013). ISBN 9788521617235. Citado na página 19.

REIMANN, M.; DOERNER, K.; HARTL, R. D-ants: Savings based ants divide and conquer the vehicle routing problem. **Computers & Operations Research**, v. 31, p. 563–591, (2004). Citado na página 13.

REIS, M. L. **Um algoritmo de geração de colunas e cortes para o problema de roteamento de veículos**. 2004. 85 p. Dissertação (Mestrado em Informática) — Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, (2004). Citado 2 vezes nas páginas 9 e 10.

RESENDE, M. G. C.; OCHI, L. S. O. Greedy randomized adaptative search procedures. **Journal of Global Optimization**, v. 6, p. 109–133, (2009). Citado na página 12.

SANTOS, F. V. **A Utilização da Pesquisa Operacional como Ferramenta para Redução de Custos na Logística de Distribuição: Problema de Roteamento de Veículos Capacitados (PRVC)**. Novembro 2014. 127 p. Monografia (Graduação em Engenharia de Produção) — Centro Universitário de Formiga, Formiga, (2014). Citado na página 10.

SOSA, N. G. M.; GALVÃO, R. D.; GANDELMAN, D. A. Algoritmo de busca dispersa aplicado ao problema clássico de roteamento de veículos. **Pesquisa Operacional**, v. 27, p. 293–310, maio/agosto (2007). Citado 2 vezes nas páginas 11 e 12.

TAN, W. F. et al. Routing vehicles with ants. **World Scientific**, v. 9, p. 495–502, (2012). Citado na página 13.

TEITZ, M. B.; BART, P. Heuristics methods for estimating the generalized vertex median of a weighted graph. **Operations Research**, v. 16, p. 955–961, (1968). Citado na página 13.

TOTH, P.; VIGO, D. **The Vehicle Routing Problem**. [S.l.]: Society for Industrial and Applied Mathematics, (2002). Citado na página 10.

UCHOA, E. et al. New benchmark instances for the capacitated vehicle routing problem. *Optimization Online*, (2014). Citado na página 32.

ZANNETI, M. C. V.; MAROCCO, A. P.; JUNIOR, H. S. C. Adaptação da metaheurística grasp como alternativa para melhorar o desempenho da logística aplicada ao e-commerce. **Holos**, v. 5, p. 320–332, (2014). Citado na página 12.

ZENG, I.; ONG, H. L.; ONG, K. M. An assignment-based local search method for solving vehicle routing problems. **Asia-Pacific Journal of Operation Research**, v. 22, p. 85–104, (2005). Citado na página 12.