



UNIVERSIDADE FEDERAL DO CEARÁ
CURSO DE GRADUAÇÃO EM CIÊNCIAS DA COMPUTAÇÃO

DANIEL DA SILVA MAIA

**UMA META-HEURÍSTICA MULTIPOPULACIONAL APLICADA AO PROBLEMA DE
ROTEIRIZAÇÃO DE VEÍCULOS COM MÚLTIPLOS DEPÓSITOS**

RUSSAS

2023

DANIEL DA SILVA MAIA

UMA META-HEURÍSTICA MULTIPULACIONAL APLICADA AO PROBLEMA DE
ROTEIRIZAÇÃO DE VEÍCULOS COM MÚLTIPLOS DEPÓSITOS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciências da Computação da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Ciências da Computação.

Orientador: Prof. Dr. Pablo Luiz Braga Soares

RUSSAS

2023

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

M185m Maia, Daniel da Silva.

Uma meta-heurística multipopulacional aplicada ao problema de roteirização de veículos com múltiplos depósitos / Daniel da Silva Maia. – 2023.
45 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Russas, Curso de Ciência da Computação, Russas, 2023.

Orientação: Prof. Dr. Pablo Luiz Braga Soares.

1. Golden Ball. 2. MDVRP. 3. K-Means. I. Título.

CDD 005

DANIEL DA SILVA MAIA

UMA META-HEURÍSTICA MULTIPOPULACIONAL APLICADA AO PROBLEMA DE
ROTEIRIZAÇÃO DE VEÍCULOS COM MÚLTIPLOS DEPÓSITOS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciências da Computação da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Ciências da Computação.

Aprovada em: 13 de Julho de 2023

BANCA EXAMINADORA

Prof. Dr. Pablo Luiz Braga Soares (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Bonfim Amaro Júnior
Universidade Estadual do Ceará (UECE)

Prof. Dr. Reuber Régis de Melo
Universidade Federal do Ceará (UFC)

À minha família, por sua capacidade de acreditar em mim e investir em mim. Mãe, seu cuidado e dedicação foi que deram, em alguns momentos, a esperança para seguir. Tia, sua presença significou segurança e certeza de que não estou sozinho nessa caminhada.

AGRADECIMENTOS

Agradeço a minha família, em especial a minha mãe e minha tia que me ajudaram e estiveram presente em todos os momentos difíceis, e fizeram de tudo possível para me permitir realizar esse sonho.

Aos meus amigos, ao qual compartilhamos muitos momentos felizes e tristes juntos, apoiando-se sempre uns nos outros.

Ao meus colegas e amigos de graduação, em especial ao Felipe e ao Carlos, que sempre estiveram presentes durante a graduação ao qual passamos por tantos momentos juntos, obrigado por cada vivência.

A minha namorada, Hernestiana, quem me apoiou, sempre acreditou e incentivou a concluir esse trabalho, te amo.

Ao meu orientador, Prof. Dr. Pablo Luiz Braga Soares, pelo incentivo, apoio e competência para orientar este trabalho no qual teve participação significativa.

Por fim, a todos que de alguma forma contribuíram positivamente na minha vida, meu muito obrigado.

“A conduta define o homem.”

(Kingsman)

RESUMO

Os gastos com logísticas de transporte são altamente dispendiosos para as empresas do ramo. Assim, otimizar rotas de transporte de veículos é um trabalho complexo. Nesse contexto, os algoritmos de roteirização de veículos vêm se mostrando uma opção pertinente para reduzir os custos sem, contudo, perder a qualidade do serviço. Ao longo dos anos, heurísticas e meta-heurísticas objetivam obter soluções logísticas eficientes, rápidas e econômicas. Neste trabalho, apresentamos o Problema de Roteirização de Veículos com Múltiplos Depósitos (Multi-depot Vehicle Routing Problem - MDVRP) e foi utilizado a meta-heurística Golden Ball (GB). Esse método recente da literatura aplica uma busca multi-populacional para encontrar soluções baseadas em conceitos de futebol. Além disso, foi realizado um comparativo da meta-heurística (GB) proposta com algumas variações de implementação da mesma.

Palavras-chave: MDVRP; golden ball; k-means; meta-heurística.

ABSTRACT

Transport logistics costs are costly for companies in the industry. Therefore, optimizing vehicle transport routes is a complex job. In this context, vehicle routing algorithms have proved to be an appropriate option to reduce costs without losing the quality of the service. Over the years, heuristics and meta-heuristics have sought efficient, fast, and economical logistics solutions. This work presents the Multi-depot Vehicle Routing Problem (MDVRP) and uses the Golden Ball (GB) metaheuristic. This recent method from the literature applies a multi-population search to find solutions based on football concepts. In addition, the meta-heuristic (GB) proposal is compared with some of its variations.

Keywords: MDVRP; golden ball; k-means; meta-heuristics.

LISTA DE FIGURAS

Figura 1 – Exemplo MDVRP	22
Figura 2 – Passo a passo de um algoritmo genético	25
Figura 3 – Fluxo do algoritmo	26
Figura 4 – Fluxograma da inicialização das soluções	28
Figura 5 – Representação de um <i>cluster</i> /jogador	28

LISTA DE TABELAS

Tabela 1 – Características do cenário	18
Tabela 2 – Características do problema físico	19
Tabela 3 – Artigos citados	22
Tabela 4 – Classificação dos algoritmos	33
Tabela 5 – Resultados do algoritmo principal	36
Tabela 6 – Resultados obtidos do algoritmo B	37
Tabela 7 – Resultados do algoritmo C	39
Tabela 8 – Resultados do Algoritmo D	41
Tabela 9 – Comparativo dos algoritmos	42

LISTA DE ABREVIATURAS E SIGLAS

<i>VRP</i>	Problema de roteirização de veículos
<i>CVRP</i>	Capacited Vehicle Routing Problem
<i>GA</i>	Genetic Algorithms
<i>GB</i>	Golden Ball
<i>HFFVRP</i>	Heterogeneous Fixed Fleet Vehicle Routing Problem
<i>LNS</i>	Large Neighborhood Search
<i>MDVRP</i>	Multi-depot Vehicle Routing Problem
<i>MDVRPTW</i>	Multi-Depot Vehicle Routing Problem with Time Windows
<i>PVRP</i>	Periodic Vehicle Routing Problem
<i>SA</i>	Simulated Annealing
<i>SDVRP</i>	Vehicle Routing Problem with Split Deliveries
<i>TS</i>	Tabu Search
<i>VRPTW</i>	Vehicle Routing Problem with Time Windows

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Justificativa	15
1.2	Objetivo Geral	15
1.3	Objetivo Específicos	15
1.4	Estrutura do trabalho	15
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	O problema básico de roteirização de veículos	17
2.2	Variantes do VRP	18
2.2.1	<i>Problema de roteirização de veículos capacitado (Capacitated VRP - CVRP)</i>	18
2.2.2	<i>Problema de roteirização de veículos com janela de tempo (VRP with Time Windows - VRPTW)</i>	19
2.2.3	<i>Problema de Roteirização de Veículos Multi-Período ou Periódico (Periodic Vehicle Routing Problem - PVRP)</i>	20
2.2.4	<i>Problema de Roteirização de Veículos com frota Heterogênea Fixa (Heterogeneous Fixed Fleet Vehicle Routing Problem - HFFVRP)</i>	20
2.2.5	<i>Problema de Roteirização de Veículos com Entregas Fracionadas (Vehicle Routing Problem with Split Deliveries - SDVRP)</i>	20
2.2.6	<i>Problema de Roteirização de Veículos com Múltiplos Depósitos - (Multi-Depot Vehicle Routing Problem - MDVRP)</i>	21
2.3	Meta-heurísticas	21
2.3.1	<i>Busca Tabu (Tabu Search - TS)</i>	23
2.3.2	<i>Recozimento Simulado (Simulated Annealing - SA)</i>	23
2.3.3	<i>Algoritmo Genético (Genetic Algorithms - GA)</i>	24
2.3.4	<i>Golden Ball (GB)</i>	25
3	METODOLOGIA	32
3.1	Caracterização da pesquisa	32
3.2	Coleta de dados	32
3.3	Procedimentos metodológicos	33
4	EXPERIMENTOS	34
5	RESULTADOS	35

5.1	Resultados do algoritmo principal	35
5.2	Resultados do Experimento B	36
5.3	Resultados do Experimento C	38
5.4	Resultados do Experimento D	40
5.5	Comparativo dos resultados	41
6	CONCLUSÕES E TRABALHOS FUTUROS	44
	REFERÊNCIAS	45

1 INTRODUÇÃO

A roteirização de veículos é um problema importante na gestão da logística em uma empresa. Neste contexto, a otimização de rotas é fundamental, possuindo, assim, um efeito direto na satisfação dos consumidores. Há diversas aplicações desse problema, como o roteamento de ônibus escolar, coleta municipal de lixo, correio e entrega de jornais, distribuição de varejo, etc, ver (THANGIAH *et al.*, 1991). Em suma, comumente evidenciado no cotidiano de empresas que disponibilizam *delivery* como formato de distribuição dos seus produtos ou serviços.

Multi-depot Vehicle Routing Problem (MDVRP) é uma variação do clássico Problema de roteirização de veículos (VRP) com a adição da restrição de possuir múltiplos depósitos. Trata-se de um problema em que uma frota de veículos, partindo de múltiplos depósitos, atende a um conjunto de clientes respeitando as restrições de demanda. No fim de todo o processo de entrega, após concluir sua rota, os veículos devem voltar para os depósitos de origem.

Neste contexto, os problemas podem conter um grande número de depósitos, tornando, dessa forma, uma decisão complexa: definir quais clientes serão servidos por quais depósitos, sem exceder as restrições de capacidade. Assim, na abordagem usada neste trabalho, será aplicado uma técnica de clusterização para agrupar os clientes com os depósitos de maneira eficiente, antes mesmo do processo de definição de rotas.

A literatura do problema apresenta profusas abordagens aplicando: modelos exatos, heurísticos e meta-heurísticos para resolver o MDVRP (BRAEKERS *et al.*, 2015). Os estudos indicam que em instâncias contendo uma quantidade pequena de clientes e depósitos, os algoritmos exatos podem ser utilizados, caso contrário, não é viável, pela demora de se obter uma solução (CACERES-CRUZ *et al.*, 2014). Neste cenário, o uso de algoritmos heurísticos e meta-heurísticos são muitas vezes mais adequados para aplicações práticas, dado que os problemas da vida real são significativamente maiores em escala. (BRAEKERS *et al.*, 2015)

Com isso em mente, esse trabalho propõe a aplicação e avaliação de uma meta-heurística multi-populacional chamada *Golden Ball* (GB), fundamentada na rotina de competições de futebol, para resolver o MDVRP. Essa técnica proposta por Osaba *et al.* (2013), remete a uma divisão de diferentes soluções em diversas equipes que treinam e enfrentam-se, criando uma competição que ocorre por um número preestabelecido de temporadas. Quando a temporada se encerra, a fase de transferências se inicia. Assim os jogadores podem trocar de equipes. Esse processo se repete até um critério de parada ser atingido. Além disso, será feito uma análise comparativa da *Golden Ball* proposta na seção 2.3 com algumas variações de implementação

também propostas neste trabalho que alteram a forma como a solução inicial é gerada e o fluxo do algoritmo para verificar a qualidade da metodologia aplicada.

1.1 Justificativa

A motivação para a realização desta pesquisa provém da possibilidade de contribuir com a literatura dos problemas de roteirização apresentando uma abordagem fundamentada em meta-heurísticas multi populacionais (*Golden Ball*).

1.2 Objetivo Geral

Esse trabalho tem como objetivo geral a utilização da *Golden Ball* aplicado ao Problema de Roteamento de Veículos com Múltiplos Depósitos, além de que, será feita uma análise comparativa dos resultados obtidos das quatro versões implementadas do *Golden Ball*.

1.3 Objetivo Específicos

- Adaptar o *Golden Ball* ao MDVRP;
- Mapear as instâncias do *dataset* de Cordeau e utilizar de entrada no *Golden Ball*;
- Comparar as versões propostas do *Golden Ball*;
- Aplicar conhecimentos práticos e teóricos obtidos no curso de Ciências da Computação para resolver um problema prático.

1.4 Estrutura do trabalho

A estrutura do trabalho está organizado em cinco capítulos, delimitada da seguinte forma: Após a introdução, temos, o capítulo 2 com a Fundamentação Teórica, na qual, são apresentados os conceitos relacionados ao trabalho com base nas pesquisas realizadas, com o intuito de criar um embasamento para o desenvolvimento do trabalho. Ademais, no capítulo 3 a metodologia do trabalho é especificada, explicando a caracterização das pesquisas realizadas para satisfazer os objetivos, como foi realizada a coleta dos dados utilizados na pesquisa e um detalhamento da resolução do trabalho em quatro fases, mapeamento do *dataset*, implementação dos algoritmos, execução dos algoritmos e uma análise comparativa dos resultados. Além disso, no capítulo 4 são explicados os detalhes na execução dos testes, como os parâmetros que foram

utilizados no algoritmo, quantas execuções foram realizadas, a máquina que foi usada e uma descrição quantitativa das instâncias escolhidas. Continuando, o capítulo 5 que estão contidos os resultados obtidos dos experimentos realizados, esses resultados estão tabelados e expõem de forma clara o que conseguimos com a pesquisa, para finalizar o capítulo 6 conclusão, é um resumo dos resultados obtidos e conclusões acerca da qualidade da metodologia aplicada.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção, são descritos os conceitos essenciais relacionados ao problema de roteirização clássico com seus parâmetros, bem como algumas das variações presentes na literatura, com ênfase na versão abordada neste trabalho. Além disso, são apresentadas profusas classificações presentes na literatura, algumas extensões e uma definição sucinta sobre meta-heurísticas, conteúdo aplicado para resolução em grande parcela de aplicações industriais no contexto dessa problemática.

2.1 O problema básico de roteirização de veículos

Os problemas de operação logística de transporte e roteirização de veículos recebem atenção em pesquisas científicas, pois o custo e planejamento envolvidos são complexos para a corporação. Os gastos com sistemas de distribuição e atendimento de serviços são uns dos maiores problemas que as empresas precisam lidar.

Nesse contexto, Dantzig e Ramser (1959) foram os pioneiros com a introdução do “Problema de despacho de caminhões”, modelando-o de maneira a considerar uma frota homogênea de caminhões para o atendimento de vários postos de gasolina, a partir de uma central e percorrendo o menor caminho possível. Ainda, segundo Christofides (1976), o problema básico de roteirização de veículos pode ser definido do modo que, veículos localizados em uma instalação central (depósito) são obrigados a visitar durante um determinado período, clientes, geograficamente, dispersos a fim de atender a uma demanda de pedidos.

Em tese, grande parcela dos problemas reais de roteirização possui algumas características e um conjunto de restrições que são ignoradas no problema clássico definido na literatura. As restrições mais usuais estão relacionadas às capacidades de carga dos veículos (CHRISTOFIDES, 1976). Dessa forma, essas vertentes podem ser classificadas em profusas categorias e tipos. Os inúmeros escopos divergem entre si em pontos alusivos ao tipo de carga, tipo de frota, número de veículos, quantidade de depósitos, a quantidade de pontos de origem, a localização dos clientes, ao tipo de função objetivo e várias outras particularidades (ELSHAER; AWAD, 2020).

A fio, são apresentadas as extensões do problema básico de roteirização de veículos, segundo os parâmetros mais relevantes e comuns. Algumas variações do problema de roteirização utilizam a combinação de características para criar novas variantes, para poderem ser aplicados a

problemas mais próximos vivenciados no cotidiano das empresas. Neste trabalho, os parâmetros foram adaptados de Braekers *et al.* (2015) sendo classificados em: características do cenário; e característica do problema físico. As Tabelas 1 e 2, a seguir, mostram as classificações:

Tabela 1 – Características do cenário

Características do cenário	Tipos
Número de paradas na rota	- Determinístico - Parcialmente Conhecido, Parcialmente probabilístico.
Divisão de carga	- Divisão Permitida - Divisão não permitida
Tipo de demanda	- Determinística - Estocástica - Desconhecida
Tempo em serviço/espera	- Determinístico - Dependente - Estocástico - Desconhecido
Janelas de tempo	- Janelas de tempo suaves - Janelas de tempo restritas - Mistura de ambos
Tipo da operação	- Coleta - Entrega - Coleta e entrega

Fonte: Elaborado pelo autor(2023)

2.2 Variantes do VRP

Nesta subseção são apresentados diversas extensões do problema clássico de roteirização de veículos, conhecido na literatura estrangeira com *vehicle routing problem (VRP)*. Foram adicionados ao conjunto de restrições inicial, aspectos apurados no cotidiano de aplicações industriais, resultando, assim, em um grande número de variantes encontradas por meio das referências. A seguir, há uma introdução das variações supracitadas, filtradas conforme a relevância para essa pesquisa.

2.2.1 Problema de roteirização de veículos capacitado (Capacitated VRP - CVRP)

O VRP clássico, também conhecido como VRP capacitado (CVRP), tem em sua natureza rotas de entrega projetadas e cada veículo percorre apenas uma rota. A frota contém as mesmas características, além da existência de somente um depósito. O objetivo é encontrar um

Tabela 2 – Características do problema físico

Características do problema físico	Tipos
Estrutura da rede	- Direcionada - Não direcionada
Localização dos clientes	- Clientes em nós - Instâncias de roteamento de arco
Número de depósitos	- Origem única - Múltipla Origem
Tipo de janela de tempo	- Restrição de clientes - Restrição de depósitos - Restrição de motoristas/veículos
Número de veículos	- Único veículo - Número limitado de veículos - Número ilimitado de veículos
Consideração de capacidade	- Veículos capacitados - Veículos não capacitados
Homogeneidade do veículo (Capacidade)	- Veículos semelhantes - Veículos específicos de carga - Veículos heterogêneos - Veículos específicos do cliente
Objetivo	- Depende do tempo de viagem - Dependente da distância - Dependente do veículo

Fonte: Elaborado pelo autor(2023)

conjunto de rotas com menor custo, com as seguintes restrições, cada cliente ser visitado apenas uma vez por veículo, cada veículo inicie e termine seu percurso na garagem, e a capacidade dos veículos não seja excedida.

2.2.2 Problema de roteirização de veículos com janela de tempo (VRP with Time Windows - VRPTW)

O VRPTW é uma extensão do problema clássico, onde é adicionado a restrição das janelas de tempo. Na pesquisa conduzida por Thangiah *et al.* (1991), os autores usaram um algoritmo genético para resolver o problema. O algoritmo elaborado consiste em dois módulos distintos: um módulo de agrupamento global que atribui clientes aos veículos por um processo denominado setorização genética e um módulo de otimização de rota local.

Já Gocken e Yaktubay (2019) efetuaram um comparativo de diferentes algoritmos de clusterização, aplicados na fase de inicialização da população de um algoritmo genético para resolver o VRPTW.

Outrossim, no trabalho de Schulz *et al.* (2019), os autores aplicaram meta-heurísticas

para a resolução do problema, que foi dividida em duas fases, para gerar a solução inicial foi utilizada a meta-heurística colônia de formigas, já para a segunda fase, aplicou-se a heurística Large Neighborhood Search(LNS) junto do algoritmo de 2-opt e o 2-opt entre rotas para aprimorar a solução obtida na primeira fase.

2.2.3 Problema de Roteirização de Veículos Multi-Período ou Periódico (Periodic Vehicle Routing Problem - PVRP)

Baseado no clássico *VRP*, no qual o horizonte de tempo do problema passa a ter N dias ao invés de ser diário. Cacchiani *et al.* (2014) apresentaram um algoritmo de otimização, incorporando componentes heurísticos e exatos, o algoritmo é baseado na relaxação de programação linear, e aplica técnicas de fixação de liberação de colunas como uma busca local, fazendo uso de uma lista tabu para evitar ciclos.

2.2.4 Problema de Roteirização de Veículos com frota Heterogênea Fixa (Heterogeneous Fixed Fleet Vehicle Routing Problem - HFFVRP)

O problema de roteirização de veículos com frota heterogênea fixa, é uma variação do *VRP* clássico, onde a frota de veículos é fixa e é composta por diversos tipos de veículos. No trabalho de MirHassani e Saadati (2014), foi utilizado uma meta-heurística de memória adaptativa reativa que utilizou a busca tabu como procedimento de melhoria para resolver o HFFVRP.

2.2.5 Problema de Roteirização de Veículos com Entregas Fracionadas (Vehicle Routing Problem with Split Deliveries - SDVRP)

Diferente do problema clássico do *VRP*, a restrição que cada cliente deve ser servido exatamente uma vez é removida, ou seja, entregas fracionadas são permitidas. Lim *et al.* (2021) efetuaram um trabalho onde foi aplicado um algoritmo genético (GA) para resolver um *VRP* com entregas fracionadas e múltiplos depósitos.

2.2.6 *Problema de Roteirização de Veículos com Múltiplos Depósitos - (Multi-Depot Vehicle Routing Problem - MDVRP)*

O problema estudado, é uma combinação do problema de roteirização de veículos clássico (*VRP*) com a adição da restrição de possuir múltiplos depósitos. Dessa forma, é um **Problema de Roteirização de Veículos com Múltiplos Depósitos (MDVRP)**.

Adianta, será elucidado detalhadamente a modelagem do problema.

Primeiramente, é assumido que a quantidade de clientes, depósitos e veículos é conhecida, no qual, cada depósito dispõe de um veículo para atender as entregas, ademais, todos os depósitos são grandes o suficiente para armazenar a demanda de todos clientes a ele associados, e todo depósito possui a mesma quantidade de clientes que precisam ser atendidos, além disso, a demanda dos clientes e a demanda que o veículo pode transportar também já é previamente conhecida. Outro ponto, é que cada veículo precisa iniciar e terminar a rota de entregas no seu depósito associado. Também vale ressaltar que um cliente só pode ser visitado uma vez.

Podemos definir o problema da seguinte forma:

- N : Número de depósitos e veículos;
- QC : Quantidade de cliente por depósito;
- M_{ij} : Distância entre um cliente i e um ponto j (cliente ou depósito);
- D_i : Demanda de um cliente i ;
- C_i : Capacidade do veículo i .

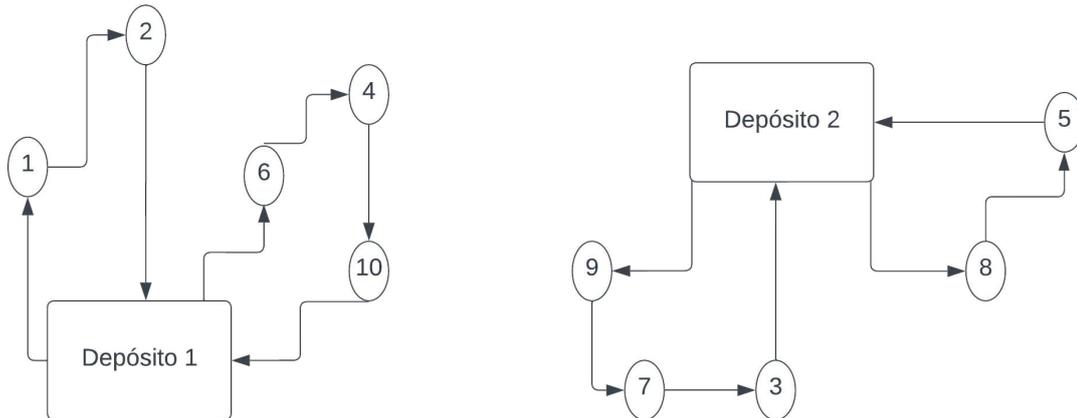
A figura 1, é um exemplo do MDVRP contendo 2 depósitos e 10 clientes, na qual, cada um dos depósitos possui 5 clientes que precisam ser atendidos.

As vertentes do problema de roteirização supracitadas não abrange todas as vertentes conhecidas da literatura. A Tabela 3, resume, alguns trabalhos da literatura e em quais extensões do VRP eles se aplicam. Destaca-se que todos os trabalhos mencionados na Tabela 3 são aplicados ao CVRP.

2.3 Meta-heurísticas

O termo meta-heurística denota um conjunto de métodos para solução que pode ser aplicado a uma ampla gama de problemas diversos, especialmente de otimização (GLOVER, 1986). Há uma coordenação de procedimentos de busca locais com estratégias de mais alto nível,

Figura 1 – Exemplo MDVRP



Fonte: Elaborado pelo autor(2022).

Tabela 3 – Artigos citados

Autores	VRPTW	MDVRP	PVRP	HFFVRP	SDVRP
(CHRISTOFIDES, 1976)					
(FISHER; JAIKUMAR, 1981)					
(THANGIAH <i>et al.</i> , 1991)	X				
(CACCHIANI <i>et al.</i> , 2014)			X		
(CORDEAU <i>et al.</i> , 1997)		X	X		
(MIRHASSANI; SAADATI, 2014)				X	
(LIM <i>et al.</i> , 2021)		X			X
(GOCKEN; YAKTUBAY, 2019)	X				
(SCHULZ <i>et al.</i> , 2019)	X				

Fonte: Elaborado pelo autor (2022).

de modo a criar um processo capaz de escapar de mínimos locais e realizar uma busca robusta no espaço de soluções de um problema (SÖRENSEN; GLOVER, 2013).

Há uma profusão de classificações para delinear estratégias de codificação por parte do conceito de meta-heurística. Desde aspectos voltados para a parametrização, até o limite da busca sobre o espaço de soluções.

Assim, para demarcar o escopo técnico utilizado, são apresentados a Busca Tabu e Recozimento Simulado, considerando uma vertente de análise de solução única, além do do algoritmo genético que utiliza busca populacional, todas essas citadas são meta-heurísticas que são grandemente exploradas na área e com inúmeros trabalhos publicados. Por fim, será explicado o algoritmo *Golden Ball*, uma meta-heurística de busca multi-populacional.

2.3.1 *Busca Tabu (Tabu Search - TS)*

É uma meta-heurística que foi proposto por Glover (1986), a Busca Tabu é um método de busca local, onde partindo de uma solução inicial, a mesma sai explorando o espaço de soluções movendo-se para outra solução que seja o seu melhor vizinho. Possui uma estrutura de memória chamada lista tabu que armazena as soluções visitadas, ou seja, não aceitando movimentos que levem a soluções já encontradas. A lista fica na memória durante um determinado espaço de tempo ou certo número de iterações, assumindo que é computacionalmente inviável armazenar todas as soluções geradas.

Como resultado final é esperado que se encontre um ótimo global ou uma solução eficiente próxima disso. Com o intuito de interromper o procedimento, duas regras são utilizadas comumente: encerrar ao atingir um número máximo de iterações ou quando o valor da melhor solução chega a um limite inferior conhecido, evitando, assim, a execução desnecessária do algoritmo.

O método também possui estratégias de intensificação que tem por objetivo concentrar a busca em regiões promissoras, podendo retornar a uma solução visitada para explorar sua vizinhança de forma mais efetiva. Cordeau *et al.* (2001) possui um trabalho com destacadas citações na literatura. Os autores utilizam a busca tabu para resolver um *VRP* com janelas de tempo.

2.3.2 *Recozimento Simulado (Simulated Annealing - SA)*

Fundamentada em técnicas da metalurgia de resfriamento. Quimicamente, os átomos tendem a se mover agitadamente (grau de entropia das moléculas) e então, gradualmente, eles se fixam em estados de energia mais baixos. Analogamente, o algoritmo inicia com uma solução S , escolhida a partir da função objetivo ou de forma aleatória e uma variável T que representa a temperatura. T , normalmente, começa com um valor alto e vai diminuindo ao decorrer da execução. Neste intervalo, o algoritmo faz um movimento até um dos seus vizinhos S' e avalia esse novo ponto. Se os resultados melhorarem no ponto S' , então movemos até ele e repetimos o processo, caso contrário se o ponto S' for pior, só iremos nos mover para ele caso nossa *probabilidade* de ir para um ponto negativo seja excedente a um número aleatório. A probabilidade de aceitar um valor negativo (solução pior) é definida por:

$$probabilidade(p) = Exp(S' - S/T)$$

A função exponencial, calcula a diferença do ponto vizinho S' subtraída pelo ponto em que estávamos anteriormente S , dividindo por T (temperatura). Deste modo, as primeiras iterações do algoritmo tem uma propriedade mais exploratória, pois temos uma probabilidade maior de aceitar valores negativos e conforme a execução, o mesmo tende a só aceitar valores positivos, diminuindo a probabilidade do algoritmo aceitar movimentos ruins. Esse procedimento é realizado até que a temperatura atinja uma temperatura final ou de congelamento. (PINHEIRO *et al.*, 2020) utilizou o SA em um problema de roteirização de veículos e obteve uma melhora de 10,71% sobre as soluções iniciais .

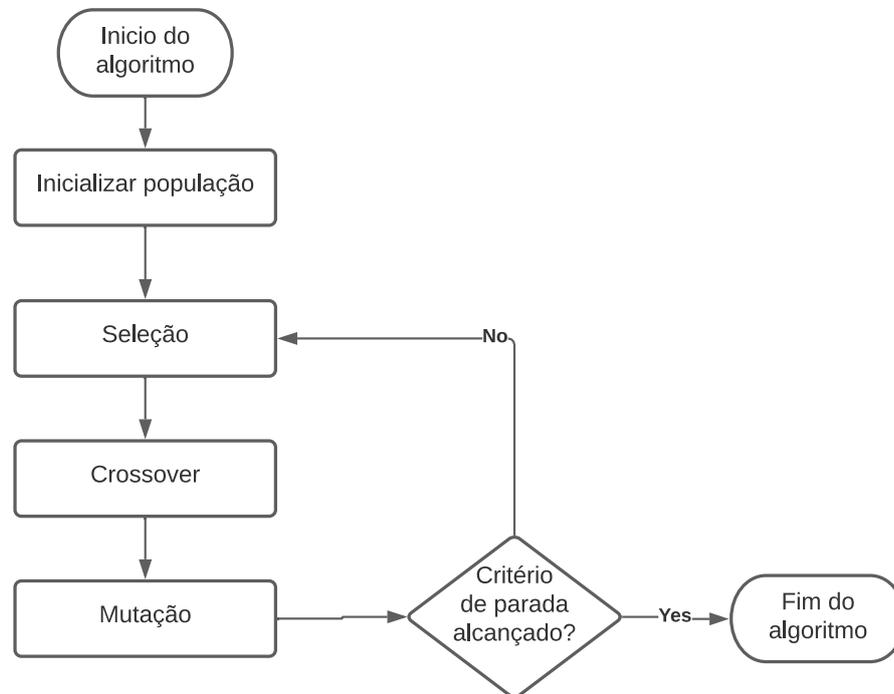
2.3.3 Algoritmo Genético (Genetic Algorithms - GA)

É uma meta-heurística para otimização baseada em técnicas existentes na teoria da evolução natural de Darwin e na reprodução genética. Segundo a teoria de C. Darwin, o princípio de seleção privilegia os indivíduos mais aptos ao ambiente com maior longevidade e, dessa maneira, com maior probabilidade de deixar seus descendentes, ou seja, os descendentes desses indivíduos perpetuam seus códigos genéticos para as próximas gerações.

De maneira semelhante, o algoritmo opera um conjunto de indivíduos chamado população que evolui aplicando técnicas que modificam os cromossomos dos mesmos. O método parte de um grupo de soluções iniciais (população), em seguida, é feita a avaliação dos indivíduos onde é medido a aptidão dos mesmos. Os mais aptos são selecionados para reproduzir em processo chamado *Crossover*, onde é re combinado o cromossomo dos que foram selecionados e novos indivíduos são criados. Por fim, a partir de uma probabilidade definida no algoritmo, ocorre uma mutação nos indivíduos que altera o cromossomo dos mesmos. Dessa forma, temos uma gama de indivíduos novos, ou seja, um novo grupo de soluções é gerado. O procedimento se repete até atingir um critério de parada, como por exemplo o número de gerações, chamado ciclos do algoritmo ver (THANGIAH *et al.*, 1991).

A figura 2, apresenta um fluxograma dos processos de um algoritmo genético:

Figura 2 – Passo a passo de um algoritmo genético



Fonte: Elaborado pelo autor(2022).

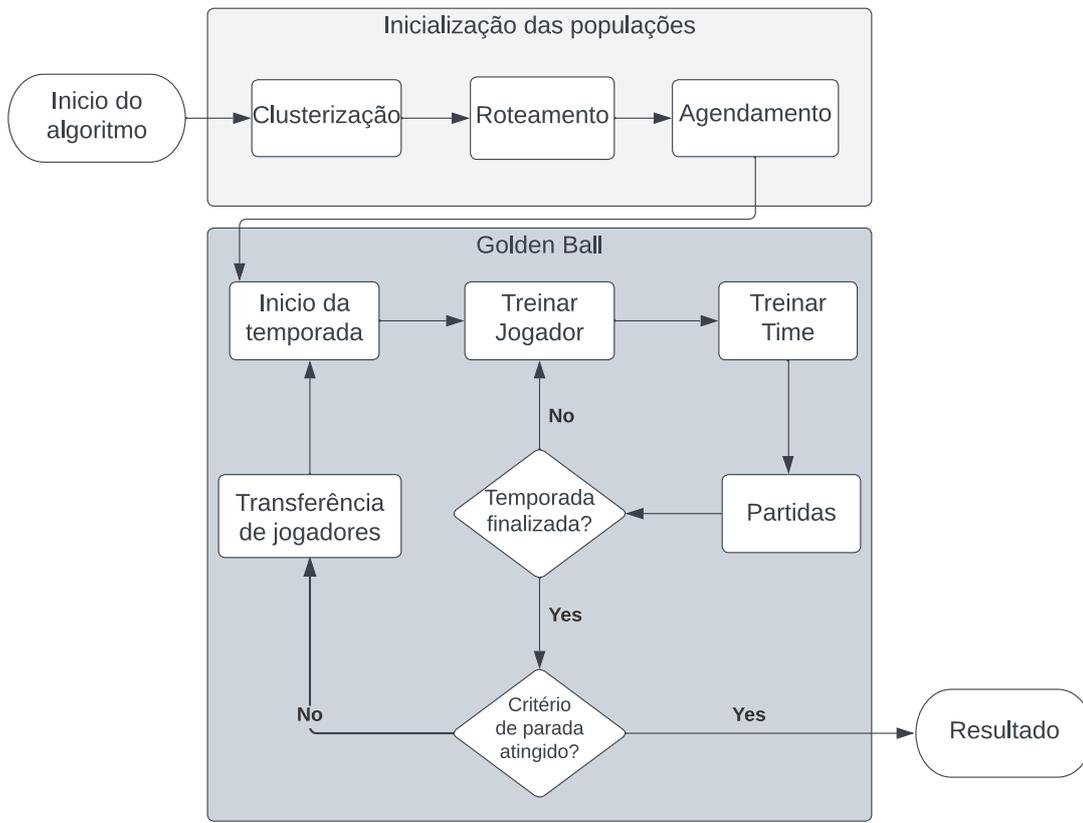
2.3.4 *Golden Ball (GB)*

A *Golden Ball* é uma meta-heurística proposta por Osaba *et al.* (2013), com características de uma busca multi-populacional utilizando conceitos oriundos do futebol. Em suma, será criada uma competição em que as possíveis soluções, competirão e o time vencedor, será a nossa solução.

De modo geral, o algoritmo é dividido em diversas etapas. Desde a criação das populações iniciais, onde serão gerados os times que competirão, como também, uma fase de treinamento, no qual, os times melhoram individualmente e coletivamente, além disso, acontecerão partidas entre os times, que gerarão uma pontuação que serão utilizados de base para o período de transferência de jogadores entre os times, enfim, todo esse processo se repete até que um dos critérios de parada seja alcançado.

Adiante, serão elucidados detalhadamente cada uma das etapas da implementação da *Golden Ball* proposta neste trabalho. Neste contexto, a figura 3 exhibe todos os processos supracitados e o fluxo do algoritmo.

Figura 3 – Fluxo do algoritmo



Fonte: Adaptado de (OSABA *et al.*, 2013).

Na **fase de inicialização das populações** (fig.3), é onde serão criados as primeiras soluções base que serão utilizadas de entrada na meta-heurística, ela pode ser dividida em três fases: **agrupamento (clusterização); roteamento e agendamento**.

No agrupamento uma vez que há muitos depósitos é necessário definir quais clientes serão servidos por quais depósitos. Neste âmbito, será utilizado a técnica de *k-means* proposto por (MACQUEEN *et al.*, 1967). O valor do parâmetro *k* que indica a quantidade de *clusters* será a quantidade de depósitos existentes na instância, sendo também o depósito utilizado como centroide durante o agrupamento. Após esta etapa, é realizado um cálculo para descobrir centroide mais próximo de cada cliente. Para finalizar, o centroide é reposicionado pela média da posição dos clientes do *cluster*, porém, nas implementações deste trabalho optou-se por não mover a localização dos depósitos por motivos do contexto do problema, após esse procedimentos, os *clusters* são criados.

Após todos os clientes serem alocados em um *cluster*, inicia-se a fase de roteamento, na qual, é criada uma rota inicial que seleciona de forma aleatória um cliente de um *cluster* e a

partir dele, os próximos são selecionados pegando os clientes mais próximos do anterior, até que todos os clientes do *cluster* estejam alocados na rota.

Na etapa de agendamento, um veículo saindo do depósito, precisa visitar todos os clientes da rota e ao final retornar ao armazém, respeitando a capacidade de demanda do veículo, dessa forma, é percorrida a rota gerada no roteamento, verificando a demanda de cada cliente e caso o veículo não tenha demanda o suficiente para atender ao cliente, é necessário ir ao depósito reabastecer o veículo.

Após as três etapas iniciais, temos uma solução gerada, ou seja, a solução é o conjunto de rotas que foram gerados para cada um dos *clusters*, e a qualidade dessa solução pode ser calculada pela média da distância percorrida dessas rotas, quanto menor a média, melhor a qualidade da solução. Durante a execução do algoritmo, serão executadas operações que tem o objetivo de diminuir a distância percorrida, mantendo a solução válida.

De maneira análoga, trazendo para o contexto da meta-heurística, cada jogador seria equivalente a um *cluster*, que por sua vez, cada jogador possui um valor de qualidade associado que seria a distância da rota do *cluster*. De forma similar, a solução é equivalente a uma equipe, ou seja, um conjunto de jogadores e o valor da equipe é a média da qualidade de seus jogadores.

As fases de roteamento e agendamento se repetem até que todas as rotas sejam criadas, ou seja, cada solução possuem os mesmos *clusters*, porém, com rotas diferentes, gerando soluções diferentes que serão utilizadas de entrada no algoritmo. A figura 4 mostra o processo de clusterização, roteamento e agendamento para um exemplo de 10 clientes e 2 depósitos. Já a figura 5 apresenta a um exemplo de um *cluster* com sua rota definida.

Cada solução criada é denominada de equipe (e), na qual, cada *cluster* da solução é um jogador (j), e todos os *clusters* equivalentes dessas soluções possuem os mesmos clientes e depósitos, vale ressaltar que essa é uma importante restrição, pois durante as operações realizadas nessas soluções, é preciso manter todas com *clusters* equivalentes, para que a solução final seja válida.

$$\text{Equipe } e_1 : \{j_1, j_2, j_3, \dots, j_{(QC)}\}$$

$$\text{Equipe } e_2 : \{j_1, j_2, j_3, \dots, j_{(QC)}\}$$

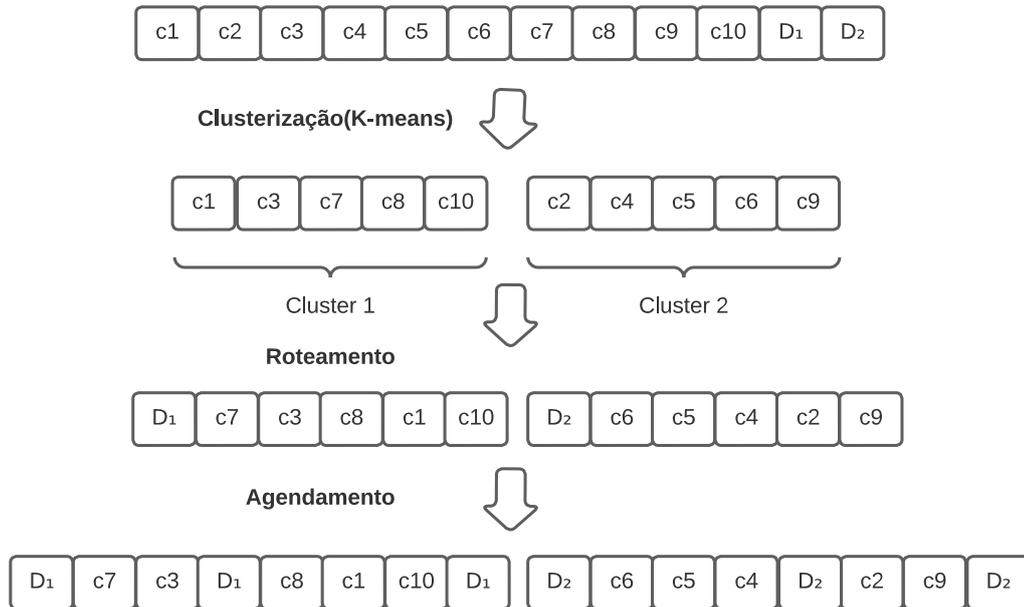
$$\vdots$$

$$\text{Equipe } e_{TE} : \{j_1, j_2, j_3 \dots, j_{(QC)}\}$$

QC : Quantidade de *clusters*.

TE : Total de equipes.

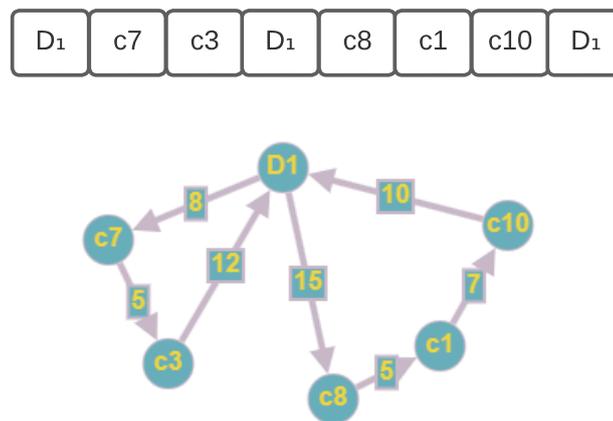
Figura 4 – Fluxograma da inicialização das soluções



Fonte: Elaborado pelo autor (2022).

Figura 5 – Representação de um *cluster/jogador*

Cluster/Jogador



Fonte: Elaborado pelo autor (2022).

Concluído esses procedimentos, tem início a fase de competição, onde as equipes treinam seus jogadores de forma individualizada e coletivamente, para que melhorem a sua aptidão. Após a fase de treino, as equipes se enfrentam por meio de um campeonato, considerando,

ainda, uma janela de transferências de jogadores ao final.

O processo é dividido em temporadas (T_i). Para que uma T_i seja finalizada, todas as equipes se enfrentam duas vezes, dessa forma toda T_i é dividida em duas partes com a mesma duração. Em cada uma dessas partes, as equipes jogam uma vez contra as outras.

Na **fase de treino** figura 3, é quando todos os jogadores de cada equipe, recebem um treinamento para que evoluam. São realizados dois tipos de treino, o individualizado onde apenas um jogador recebe o treinamento e o treino do time, um treino coletivo.

Trazendo para o contexto do algoritmo, o treinamento individualizado sucede ao selecionarmos um cliente aleatório de um *cluster* e trocarmos sua posição na rota com seu sucessor, ao fazermos isso, a distância da rota é recalculada, levando em consideração as restrições de capacidade, é possível haver uma melhora na distância do *cluster* (jogador).

De forma análoga, no treinamento coletivo, é realizado um treinamento em dois jogadores de uma equipe, nesse processo, dois clientes selecionados de forma aleatória de *clusters* também escolhidos de forma aleatória são trocados (*Crossover*), realizado a troca, o treinamento individualizado é realizado nos novos conjuntos de clientes, buscando uma rota que possua uma aptidão melhor que a anterior.

$$J_i : [c_0, c_1, c_2, c_3, c_4]$$

$$J_k : [c_9, c_8, c_7, c_6, c_5]$$

Onde c_i e c_k são as características (clientes) dos jogadores (clusters), uma das combinações possíveis dos atributos dos mesmos, resulta em dois novos jogadores, que é descrito a seguir:

$$J'_i : [c_0, c_5, c_2, c_3, c_4]$$

$$J'_k : [c_9, c_8, c_7, c_6, c_1]$$

No exemplo acima, o cliente c_1 do jogador J_i e o cliente c_5 do jogador J_k foram trocados, resultando em dois novos jogadores J'_i e J'_k .

Esse processo todo se repete para todas as soluções trocando os mesmos clientes dos mesmos *clusters*, verificado se houve melhora em todas as soluções, os novos times são criados com o conjunto de cliente de cada *cluster* e rotas alteradas.

Finalizado os treinamentos as **Partidas** se iniciam, figura 3, cada partida consiste marcar o máximo de gols. A definição do placar de uma partida sucede da seguinte forma, dois

jogadores se enfrentam pela posição em que estão na equipe. Assim sendo, o jogador com maior aptidão daquela posição marca um gol para sua equipe. Naturalmente, a equipe que marcar mais gols é a vencedora da partida. Além disso, a equipe vencedora ganha 3 pontos enquanto que a perdedora não marca pontos 0. Em caso de empate, cada equipe ganha 1 ponto. Em resumo, os pontos são utilizados para definir a classificação no campeonato.

O **período de transferências** figura 3, é o processo em que as equipes trocam os jogadores entre si. No mundo do futebol, esse processo ocorre todo ano, toda temporada possui dois períodos de transferências. Da mesma forma, esse processo é implementado pela GB, porém, com apenas um período de transferência. Nesse contexto, a pontuação e a posição no campeonato de cada equipe (e) são os fatores que definem quais transferências serão realizadas.

Desse modo, no fim da temporada, as equipes (e) que estão no topo da classificação são aperfeiçoadas com os melhores jogadores j_i das equipes que estão logo abaixo na tabela, analogamente, os piores jogadores destas equipes que estão no topo da tabela, são trocados por jogadores equivalentes (que possuem os mesmo conjunto de clientes) das equipes de baixo. Em tese, o processo de transferências é mostrado a seguir, em um campeonato com 4 equipes, os mesmos estão ranqueados de acordo a classificação da liga e os jogadores j_i estão ordenados de acordo com a aptidão (Apt).

Equipe e_1 : $\{j_4, j_1, j_3, j_2\}$

Equipe e_2 : $\{j_3, j_4, j_2, j_1\}$

Equipe e_3 : $\{j_3, j_1, j_4, j_2\}$

Equipe e_4 : $\{j_1, j_2, j_4, j_3\}$

Equipe e_1 : Apt $j_4 = 180$; Apt $j_1 = 200$; Apt $j_3 = 220$; Apt $j_2 = 230$; Total: 830

Equipe e_2 : Apt $j_3 = 190$; Apt $j_4 = 200$; Apt $j_2 = 230$; Apt $j_1 = 225$; Total: 845

Equipe e_3 : Apt $j_3 = 190$; Apt $j_1 = 220$; Apt $j_4 = 230$; Apt $j_2 = 250$; Total: 890

Equipe e_4 : Apt $j_1 = 220$; Apt $j_2 = 225$; Apt $j_4 = 230$; Apt $j_3 = 240$; Total: 915

Encerrado o período de transferências, as equipes são formadas da seguinte forma:

Equipe e_1 : Apt $j_4 = 180$; Apt $j_1 = 200$; Apt $j_3 = 190$; Apt $j_2 = 230$; Total: 800

Equipe e_2 : Apt $j_3 = 190$; Apt $j_4 = 200$; Apt $j_2 = 230$; Apt $j_1 = 220$; Total: 840

Equipe e_3 : Apt $j_3 = 220$; Apt $j_1 = 220$; Apt $j_4 = 230$; Apt $j_2 = 225$; Total: 895

Equipe e_4 : Apt $j_1 = 225$; Apt $j_2 = 250$; Apt $j_4 = 230$; Apt $j_3 = 240$; Total: 945

Esse processo de transferências de jogadores, favorece a técnica de busca do algoritmo, evitando cair em ótimos locais além de aumentar a capacidade de busca da meta-heurística.

Em suma, esse processo ajuda a aprimorar a capacidade exploratória do algoritmo.

Neste contexto, chegamos ao **Critério de parada** que, por sua vez, deve permitir que a busca do algoritmo percorra uma enorme área do espaço de solução. O critério desenvolvido é composto por duas cláusulas:

$$\sum_{i=1}^{TE} AE'_i \leq \sum_{i=1}^{TE} AE_i$$

$$MelhorSolucao' \leq MelhorSolucao$$

A execução do GB se encerra, quando um dos critérios de parada é alcançado, o primeiro critério é atingido quando não há uma melhora na soma da aptidão de todas as equipes, comparadas com a temporada anterior. O segundo é quando a qualidade melhor solução encontrada não evoluiu, em contraste com a temporada antecedente. Uma vez que o critério de parada é alcançado, o algoritmo retorna a equipe com maior pontuação do campeonato do campeonato, em conclusão, a solução do problema.

(GUEZOULI *et al.*, 2018) propuseram a aplicar a GB no problema de roteirização de veículos com múltiplos depósitos e janelas de tempo (MDVRPTW), onde na fase de inicialização da população, foi realizado um agrupamento dos depósitos com os clientes, em que os clientes eram associados aos depósitos mais próximos de acordo com a distância euclidiana, além disso, cada jogador era representado por um *cluster* (depósito + clientes).

3 METODOLOGIA

Nesta seção é apresentada a metodologia a ser aplicada. Portanto, serão descritos quais ações e decisões serão seguidos para alcançar o objetivo principal do trabalho.

3.1 Caracterização da pesquisa

Do ponto de vista de seus objetivos, essa pesquisa caracteriza-se como exploratória, objetivando adquirir familiaridade com o tema. Em relação à forma da abordagem, trata-se de uma pesquisa quantitativa, pois os dados, resultados e comparações serão ilustrados em tabelas e discutidos ao decorrer da escrita do texto. Por último, no que se refere a procedimentos técnicos, caracteriza-se como pesquisa bibliográfica, uma vez que foi desenvolvida a partir de material já publicado (ALMEIDA, 2014).

Para a realização da pesquisa descrita neste trabalho, foi realizada uma exploração bibliográfica na base de dados CAPES e o *Google Acadêmico*, utilizando como palavras-chave “Problema de roteamento de veículos”, “*Golden Ball*”, “Algoritmos Genéticos”, “Recozimento Simulado” em português e inglês. Em seguida, foram selecionados trabalhos com muitas citações que aplicaram as meta-heurísticas citadas acima na resolução do VRP e de suas vertentes, onde foi analisado a abordagem de resolução dos mesmos.

3.2 Coleta de dados

Em relação aos procedimentos usadas para a coleta de dados, que consistem em um conjunto de instâncias do problema chamados *Datasets*, foi realizada uma pesquisa, utilizando de palavras-chave “MDVRP” e “*Datasets*” no *Google*, o qual encontrou um repositório online para o VRP, chamado vrp-rep, que possui instancias que foram utilizados em trabalhos anteriores. Utilizando como filtro de pesquisa “MDVRP” no repositório, encontrou-se 7 conjuntos de *datasets* utilizados na literatura, foi utilizado exclusivamente o de Cordeau *et al.* (1997) que possui uma excelente organização dos dados o que permitiu um fácil entendimento, este possui 33 instâncias que serão utilizadas de entrada no nosso algoritmo para validar a sua eficiência.

Após a realização da coleta, os dados serão analisados e os resultados obtidos desses *datasets*, serão examinados e mostrados em forma de tabelas. Com base nisso, será possível adquirir uma conclusão acerca da qualidade do algoritmo.

3.3 Procedimentos metodológicos

O procedimento proposto para a resolução do problema e validação do algoritmo pode ser dividida em três fases principais: mapeamento dos dados do *dataset*; execução dos algoritmos; e a análise dos resultados.

Neste contexto, antes de tudo, foi necessário realizar um mapeamento dos dados das instâncias do *dataset*, no qual, após, uma análise para entendimento de como os dados estavam dispostos, foi criado um método que recebe os dados das instâncias e mapeava para suas devidas entidades no código (Cliente, Depósito, Veículo).

Após essa fase inicial, com as entidades já devidamente mapeadas, foram executados quatro variações da *Golden Ball* explicada na seção 2.3. Cada uma das variações diferem na forma de como a fase de clusterização é realizada e se a troca de clientes entre os *clusters* é realizada, verificando se a qualidade das soluções melhoraram ou não. As duas primeiras variações serão com o agrupamento baseado nos clientes mais próximos dos depósitos, uma verificando se a troca de clientes é positiva para a solução e só realizando a troca nesses casos, e a outra fazendo a troca de clientes sem verificar a qualidade das trocas. As duas ultimas serão com a fase de agrupamento alocando os clientes para um dos depósitos disponível de forma aleatória, onde, de forma semelhante às variações anteriores, uma verifica a qualidade da troca de clientes, e só troca aquelas que melhoram a solução, e a outra realiza trocas que nem sempre melhoram a qualidade. Ver a tabela 4 referente a essas variações.

Tabela 4 – Classificação dos algoritmos

Algoritmos	Clusterização K-Means	Clusterização Aleatório	Com verificação	Sem verificação
Algoritmo Principal	X		X	
Algoritmo B	X			X
Algoritmo C		X	X	
Algoritmo D		X		X

Fonte: Elaborado pelo autor (2023).

Para finalizar, é realizado uma análise comparativa dos resultados obtidos das execuções dos algoritmos, mencionados a cima, no qual, os resultados serão tabelados no capítulo 5, será verificado a qualidade dos algoritmos executados conforme o valor da solução obtida, tempo de execução, quais algoritmos foram melhores em certo cenários, etc.

4 EXPERIMENTOS

Neste capítulo, será apresentado as configurações de parâmetros utilizados nos testes, a linguagem utilizada, configuração da máquina utilizada e um detalhamento dos dados das instâncias do *dataset* escolhido.

Neste contexto, os parâmetros para a execução do GB serão apresentados a seguir.

- Quantidade de equipes = 8;
- Quantidade de jogadores por equipe (depende da quantidade de depósitos para cada *dataset*);
- Quantidade de treinos individuais = 100;
- Quantidade de treinos coletivos = 100;
- Quantidade de temporadas = 4;
- Quantidade de partidas (depende da quantidade de equipes).

Além dos parâmetros definidos, é importante mencionar que cada versão do algoritmo foi executado uma única vez utilizando e foi desenvolvido na linguagem de programação C#, os parâmetros definidos acima e os resultados estão dispostos em tabelas no capítulo 5.

Acerca da configuração da máquina utilizada nos experimentos, foi usado o meu computador pessoal para realizar os experimentos dos algoritmos, contendo as seguintes configurações:

- Processador Ryzen 7 5700x CPU @ 3.40GHz 4.20GHz;
- Contendo 32 GB de memória RAM 3600Mhz;
- Com o Sistema operacional *Windows 10*.

O *dataset* foi proposto por (CORDEAU *et al.*, 1997) e encontra-se disponível para download no repositório online VRP - REP, suas instâncias podem conter e variar de 48 até 360 clientes, no qual, esses clientes estão dispostos em um grande espaço geométrico, e possuem entre 1 e 100 quantidade de demanda, o número de depósitos pode variar de 2 até 9 depósitos, onde a capacidade de transporte dos veículos desses depósitos variam de 60 até 500 de capacidade. Vale mencionar que as instâncias p01, p02, p06, p08, p10 e p11 não serão executados, pois, não há uma quantidade igual de clientes para cada *cluster*, nossa implementação não trata esse cenário

5 RESULTADOS

Neste capítulo, apresentamos os resultados obtidos por meio da realização de experimentos e análises. Os resultados são apresentados utilizando tabelas para facilitar a visualização e compreensão dos dados obtidos.

A seguir, apresentamos as principais tabelas que contêm os resultados dos experimentos realizados:

As tabelas estão assim organizadas, a coluna 1 indica o nome da instância, a coluna 2 indica o valor da solução inicial gerada, a coluna 3 o valor da solução final obtida após a execução do *Golden Ball*, a coluna 4 indica o tempo de execução em segundos da instância e por ultimo a coluna 5 indica o GAP que é o percentual de diferença da solução inicial para a solução final.

5.1 Resultados do algoritmo principal

Foram obtidos resultados altamente promissores, nos quais foi alcançada uma média de melhoria de 11,31% por instância, ou seja, foi diminuída a distância total percorrida em 11,31% por instância. Essa melhoria variou entre 5,97% nos casos em que apenas o treino interno foi realizado e 25,94% nas instâncias em que o algoritmo conseguiu realizar trocas significativamente boas de clientes entre os clusters.

Além do aprimoramento das soluções, é importante destacar o tempo de execução, que variou entre 2,29 e 62,20 segundos no caso mais demorado. Esse tempo está relacionado à quantidade de tentativas que o algoritmo fez para trocar clientes entre os clusters. Como o algoritmo aceita apenas trocas positivas nas soluções e essas trocas são selecionadas de forma aleatória, a quantidade de tentativas por instância pode variar consideravelmente, resultando nessa diferença de tempo. Nos piores cenários, como o da instância p09, p05 e pr06, são as instâncias que mais possuem clientes por depósito, havendo uma quantidade maior de possíveis trocas para o algoritmo realizar, o que levou a uma maior exploração do espaço de soluções, aumentando consideravelmente o tempo de execução.

É importante ressaltar que, no algoritmo testado, as soluções iniciais já possuem uma qualidade consideravelmente boa. Isso ocorre porque suas rotas foram criadas a partir dos clusters, nos quais os depósitos foram agrupados com os clientes com base na proximidade. Nesse sentido, as melhorias nas soluções foram extremamente positivas.

Tabela 5 – Resultados do algoritmo principal

Instâncias	Solução Inicial	Solução final	Tempo de Execução(s)	GAP (%)
p03	281,92	208,78	5,26	25,94
p04	744,98	678,81	25,54	8,88
p05	664,22	541,04	25,83	18,55
p07	348,93	321,21	9,77	7,92
p09	2036,59	1873,01	62,20	8,03
p12	1012,64	888,96	18,01	12,21
p13	994,17	890,38	17,92	10,44
p14	960,29	900,95	17,93	6,18
p15	995,91	916,71	19,32	7,95
p16	950,46	863,74	19,25	9,12
p17	1007,55	914,73	19,25	9,21
p18	987,45	888,46	20,49	10,02
p19	977,04	918,67	20,52	5,97
p20	1036,77	927,62	20,56	10,53
p21	1006,70	918,32	22,64	8,78
p22	1018,34	919,59	22,65	9,70
p23	1006,19	930,55	22,64	7,52
pr01	511967,60	392080,64	2,29	23,42
pr02	603086,64	499923,35	8,05	17,11
pr03	878454,43	800229,62	16,91	8,90
pr04	924192,51	789855,89	26,04	14,54
pr05	1084918,00	956084,18	37,43	11,87
pr06	1219759,72	1109388,48	50,31	9,05
pr07	409000,05	367636,81	3,96	10,11
pr08	567428,74	483787,92	10,30	14,74
pr09	737877,92	683709,30	18,10	7,34
pr10	994650,80	880518,75	27,93	11,47
			21,15	11,31

Fonte: Elaborado pelo autor (2023).

5.2 Resultados do Experimento B

Ao contrário do primeiro algoritmo testado, esse algoritmo realiza trocas entre clientes de clusters diferentes sem levar em consideração a qualidade dessas trocas. Devido à qualidade inicial já consideravelmente boa da solução, foram obtidos 9 resultados positivos, resultando em uma otimização média de quase 4% em relação à solução inicial. No entanto, em 18 instâncias, os resultados foram negativos, especialmente as instâncias p12, p13 e p14, que

Tabela 6 – Resultados obtidos do algoritmo B

Instâncias	Solução Inicial	Solução final	Tempo de Execução(s)	GAP (%)
p03	250,94	242,28	0,40	3,44
p04	743,48	804,78	0,37	-8,24
p05	613,76	672,99	0,35	-9,65
p07	379,60	386,13	0,20	-1,72
p09	1984,33	1971,01	0,87	0,67
p12	1061,06	1465,16	0,24	-38,08
p13	1079,96	1333,73	0,24	-23,49
p14	973,73	1252,84	0,24	-28,66
p15	1051,45	1140,68	0,34	-8,48
p16	1025,80	1045,40	0,36	-1,91
p17	994,27	1105,28	0,35	-11,16
p18	1033,82	1113,42	0,49	-7,69
p19	1066,51	1131,98	0,51	-6,13
p20	1026,93	1046,66	0,46	-1,92
p21	1033,93	1113,56	0,67	-7,70
p22	1031,57	1038,31	0,64	-0,65
p23	1024,59	1068,96	0,65	-4,33
pr01	531881,77	571480,13	0,07	-7,44
pr02	662486,89	653916,99	0,19	1,29
pr03	925235,01	921958,31	0,30	0,35
pr04	965762,51	938238,04	0,47	2,85
pr05	1082690,62	966741,02	0,65	10,70
pr06	1209845,51	1169056,74	0,87	3,37
pr07	393447,94	442940,49	0,10	-12,57
pr08	530025,11	585400,07	0,22	-10,44
pr09	772790,06	752969,07	0,39	2,56
pr10	1065722,36	955278,88	0,61	10,36
			0,41	3,95

Fonte: Elaborado pelo autor (2023).

possuem uma capacidade de carga do veículo bastante limitada, fazendo com que cada alteração negativa, tivesse um impacto significativo no valor da solução.

É relevante mencionar que o tempo de execução dessas instâncias foi extremamente baixo, não excedendo 1 segundo por instância, isso se deve ao fato de que o critério de parada do algoritmo era rapidamente alcançado ao obter uma solução pior do que a temporada anterior, resultando no término imediato da execução do algoritmo e no retorno da solução.

5.3 Resultados do Experimento C

De maneira semelhante ao que foi mencionado na seção 5.1, os resultados obtidos com este algoritmo foram altamente favoráveis. O tempo médio de execução por instância foi de 21 segundos, variando entre 3,86 e 61,59 segundos, esses valores estão associados à instância com o menor número de clientes por depósito e à instância com o maior número de clientes por depósito, respectivamente. Quanto maior o número de clientes, maior é o escopo de possíveis trocas.

O algoritmo demonstrou resultados positivos em todas as instâncias, com uma média de otimização de rota de 10,41%. Houve uma variação de 4,04% a 22,74% em relação à melhoria da solução inicial. Essa diferença decorre da capacidade do algoritmo em realizar ou não a troca de clientes. No pior cenário, houve apenas um ajuste na rota existente, sem nenhuma troca de clientes. Por outro lado, nos cenários com os melhores resultados, a troca de clientes foi possível, o que teve um impacto significativo na qualidade da solução.

Em resumo, o algoritmo apresentou um tempo de execução e um índice de otimização comparáveis ao algoritmo apresentado na seção 5.1. No entanto, sua solução inicial foi pior, uma vez que a clusterização dos clientes e depósitos foi realizada de forma aleatória neste algoritmo.

Tabela 7 – Resultados do algoritmo C

Instâncias	Solução Inicial	Solução final	Tempo de Execução(s)	GAP (%)
p03	399,59	372,49	5,55	6,78
p04	941,27	875,49	25,50	6,98
p05	763,09	711,87	25,69	6,71
p07	588,70	519,39	9,60	11,77
p09	3475,06	3319,44	61,59	4,47
p12	1722,80	1330,99	17,78	22,74
p13	1595,79	1483,21	17,76	7,05
p14	1784,75	1398,01	17,92	21,66
p15	1876,72	1800,89	19,08	4,04
p16	2135,22	1850,65	19,11	13,32
p17	2183,72	2002,66	19,13	8,29
p18	2740,94	2572,42	20,41	6,14
p19	2844,30	2510,22	20,46	11,74
p20	2800,25	2496,24	20,83	10,85
p21	3004,69	2828,69	23,38	5,85
p22	3043,12	2865,24	23,29	5,84
p23	2930,77	2804,35	22,73	4,31
pr01	642970,03	499725,39	3,86	22,27
pr02	937292,90	814025,23	9,29	13,15
pr03	1458138,51	1278221,92	16,67	12,33
pr04	1499741,16	1321130,96	25,83	11,90
pr05	1550842,17	1364237,20	37,09	12,03
pr06	1923844,24	1775713,50	50,08	7,69
pr07	713892,99	620535,13	4,40	13,07
pr08	1019944,51	916217,80	10,25	10,16
pr09	1171920,26	1055278,53	18,06	9,95
pr10	1641016,72	1476003,73	27,79	10,05
			21,22	10,41

5.4 Resultados do Experimento D

Neste último experimento realizado, as substituições de clientes entre os *clusters* foram realizadas independentemente do resultado ser positivo ou negativo. Por essa razão e de maneira similar ao experimento descrito na seção 5.2, o tempo de execução foi extremamente baixo, não excedendo 1 segundo por instância. Isso ocorreu porque, assim que uma troca piorava a solução, o critério de parada era alcançado e a solução obtida era retornada ao final do processo.

Foram registrados 22 resultados positivos, com uma média de aproximadamente 8% de melhoria em relação à solução inicial. Esses ganhos variaram de 1,08% a 16,50% de otimização por instância. Além dos resultados positivos mencionados, também foram obtidos 5 resultados negativos. Esses casos ocorreram em instâncias que possuíam veículos com capacidade de carga reduzida, o que fazia com que qualquer alteração desfavorável resultasse em um resultado insatisfatório. Esses resultados negativos são semelhantes aos explicados anteriormente na seção 5.2.

Em suma, o algoritmo apresentou resultados satisfatórios com um tempo de execução muito baixo. Isso era esperado, uma vez que a solução inicial é gerada aleatoriamente, permitindo muitas possibilidades de melhoria. Além disso, o processo é concluído e a solução é retornada rapidamente devido ao critério de parada ser alcançado quando uma troca piora a solução.

Tabela 8 – Resultados do Algoritmo D

Instâncias	Solução Inicial	Solução final	Tempo de Execução(s)	GAP (%)
p03	407,05	367,17	0,38	9,79
p04	960,96	1009,59	0,36	-5,06
p05	810,10	796,87	0,35	1,63
p07	610,12	565,41	0,19	7,32
p09	3692,71	3578,34	0,96	3,09
p12	1920,13	1716,74	0,23	10,59
p13	1726,12	1844,65	0,22	-6,86
p14	1777,10	2092,81	0,25	-17,76
p15	2143,14	2201,64	0,37	-2,72
p16	2215,90	2080,92	0,33	6,09
p17	2031,22	2152,81	0,36	-5,98
p18	2910,00	2429,71	0,48	16,50
p19	2761,31	2569,87	0,47	6,93
p20	2546,65	2375,80	0,54	6,70
p21	3174,23	2887,24	0,71	9,04
p22	3156,81	2853,10	0,67	9,62
p23	3162,49	2749,69	0,70	13,05
pr01	613932,78	568072,39	0,09	7,46
pr02	883878,76	856500,00	0,20	3,09
pr03	1520782,26	1326794,06	0,33	12,75
pr04	1600101,57	1495947,88	0,49	6,50
pr05	1477059,64	1461023,94	0,68	1,08
pr06	1990530,34	1835736,10	0,93	7,77
pr07	773726,91	682862,62	0,11	11,74
pr08	1037959,15	934343,44	0,26	9,98
pr09	1310520,87	1188161,56	0,44	9,33
pr10	1643750,14	1554056,81	0,63	5,45
			0,43	7,97

5.5 Comparativo dos resultados

Na tabela 9, a seguir está um comparativo dos resultados alcançados com os 4 algoritmos propostos.

Tabela 9 – Comparativo dos algoritmos

Instâncias	Algoritmo Principal	Algoritmo B	Algoritmo C	Algoritmo D
p03	208,78	242,28	372,49	367,17
p04	678,81	804,78	875,49	1009,59
p05	541,04	672,99	711,87	796,87
p07	321,21	386,13	519,39	565,41
p09	1873,01	1971,01	3319,44	3578,34
p12	888,96	1465,16	1330,99	1716,74
p13	890,38	1333,73	1483,21	1844,65
p14	900,95	1252,84	1398,01	2092,81
p15	916,71	1140,68	1800,89	2201,64
p16	863,74	1045,40	1850,65	2080,92
p17	914,73	1105,28	2002,66	2152,81
p18	888,46	1113,42	2572,42	2429,71
p19	918,67	1131,98	2510,22	2569,87
p20	927,62	1046,66	2496,24	2375,80
p21	918,32	1113,56	2828,69	2887,24
p22	919,59	1038,31	2865,24	2853,10
p23	930,55	1068,96	2804,35	2749,69
pr01	392080,64	571480,13	499725,39	568072,39
pr02	499923,35	653916,99	814025,23	856500,00
pr03	800229,62	921958,31	1278221,92	1326794,06
pr04	789855,89	938238,04	1321130,96	1495947,88
pr05	956084,18	966741,02	1364237,20	1461023,94
pr06	1109388,48	1169056,74	1775713,50	1835736,10
pr07	367636,81	442940,49	620535,13	682862,62
pr08	483787,92	585400,07	916217,80	934343,44
pr09	683709,30	752969,07	1055278,53	1188161,56
pr10	880518,75	955278,88	1476003,73	1554056,81
27 instâncias				

Com base nos resultados apresentados, era esperado que o algoritmo principal, inicialmente proposto, obtivesse resultados superiores em todas as instâncias. No entanto, ao analisar os dados, observou-se que nas instâncias pr05 e pr06, o algoritmo B, que realizava trocas sem verificar a qualidade das mesmas, obteve resultados quase tão bons quanto o algoritmo principal. Isso sugere que, em certos casos, a abordagem menos restritiva do algoritmo B pode ser eficaz.

Além disso, um ponto relevante a se destacar é o tempo de execução. Verificou-se que o algoritmo B possui uma média de tempo de execução cerca de 50 vezes mais rápida do que o algoritmo A (principal). Essa diferença significativa no tempo de processamento pode ser um fator importante a ser considerado ao escolher entre os algoritmos.

Outro aspecto a ser mencionado são os algoritmos C e D, que utilizaram uma solução inicial criada a partir de uma clusterização aleatória. Surpreendentemente, o algoritmo D, que não verificava as trocas, obteve resultados superiores em 6 instâncias em comparação com o algoritmo C. Essa descoberta sugere que a abordagem do algoritmo D, apesar de sua simplicidade, pode ser vantajosa em determinadas situações.

Em resumo, o algoritmo principal demonstrou sua superioridade na maioria das instâncias, conforme esperado. No entanto, os resultados também revelaram que o algoritmo B obteve resultados próximos e com um tempo de execução significativamente menor. Além disso, os algoritmos C e D mostraram-se competitivos entre si em algumas instâncias, apesar de suas limitações.

6 CONCLUSÕES E TRABALHOS FUTUROS

A metodologia implementada demonstrou ser uma alternativa promissora na resolução do MDVRP. O algoritmo principal proposto neste estudo exibiu um desempenho eficiente, com uma média de otimização por instância de 11,31% em relação aos outros algoritmos *B*, *C* e *D*. Vale destacar que em algumas instâncias, o algoritmo principal foi capaz de obter uma otimização de quase 27% por meio de excelentes trocas entre os clientes dos *clusters*. Adicionalmente, é importante mencionar o tempo de execução do algoritmo principal, o qual mostrou melhor resultado em relações aos demais. Somente nos casos em que havia um grande número de clientes por depósito é que o tempo de execução ultrapassou 1 minuto.

As variações da implementação que foram apresentadas nas seções 5.2 e 5.4, revelaram *insights* interessantes sobre o algoritmo como o baixíssimo tempo de execução, tendo em média, 0.42s por instância, e que em cenários onde a capacidade de carga do veículo é bastante elevada, as trocas realizadas sem melhora da qualidade garantida, retornou resultados próximos do algoritmo principal. No entanto, constatou-se a necessidade de aprimorar ainda mais a implementação, a fim de permitir que o algoritmo explore de forma mais eficiente o espaço de soluções, evitando a interrupção prematura da execução devido ao critério de parada.

Para pesquisas futuras, seria interessante adaptar o algoritmo de forma a suportar as restrições que não foram abordadas neste trabalho, tais como a duração máxima da rota por veículo, o tempo de serviço do cliente e a utilização da quantidade de veículos definida nas instâncias, outra sugestão, será a implementação de um algoritmo híbrido do *golden ball* com a busca tabu, para salvar as soluções ruins já visitadas evitando cair nelas novamente, além de fazer um comparativo com os trabalhos disponíveis na literatura.

REFERÊNCIAS

- ALMEIDA, M. de S. **Elaboração de Projeto, TCC, Dissertação e Tese: uma abordagem simples, prática e objetiva**. [S.l.]: Atlas, 2014. v. 2.
- BRAEKERS, K.; RAMAEKERS, K.; NIEUWENHUYSE, I. The vehicle routing problem: State of the art classification and review. **Computers Industrial Engineering**, v. 99, 12 2015.
- CACCHIANI, V.; HEMMELMAYR, V.; TRICOIRE, F. A set-covering based heuristic algorithm for the periodic vehicle routing problem. **Discrete Applied Mathematics**, Elsevier B.V, Netherlands, v. 163, n. Pt 1, p. 53–64, 2014. ISSN 0166-218X.
- CACERES-CRUZ, J.; ARIAS, P.; GUIMARANS, D.; RIERA, D.; JUAN, A. A. Rich vehicle routing problem: Survey. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 47, n. 2, dec 2014. ISSN 0360-0300. Disponível em: <<https://doi.org/10.1145/2666003>>.
- CHRISTOFIDES, N. The vehicle routing problem. **RAIRO - Operations Research - Recherche Opérationnelle**, EDP-Sciences, v. 10, n. V1, p. 55–70, 1976. Disponível em: <http://www.numdam.org/item/RO_1976__10_1_55_0/>.
- CORDEAU, J.-F.; GENDREAU, M.; LAPORTE, G. A tabu search heuristic for periodic and multi-depot vehicle routing problems. **Networks**, v. 30, n. 2, p. 105–119, 1997. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291097-0037%28199709%2930%3A2%3C105%3A%3AAID-NET5%3E3.0.CO%3B2-G>>.
- CORDEAU, J.-F.; LAPORTE, G.; MERCIER, A. A unified tabu search heuristic for vehicle routing problems with time windows. **Journal of the Operational Research Society**, Taylor Francis, v. 52, n. 8, p. 928–936, 2001. Disponível em: <<https://doi.org/10.1057/palgrave.jors.2601163>>.
- DANTZIG, G.; RAMSER, R. The truck dispatching problem. **Management Science**, v. 6, p. 80–91, 10 1959.
- ELSHAER, R.; AWAD, H. A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants. **Computers Industrial Engineering**, v. 140, p. 106242, 2020. ISSN 0360-8352. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0360835219307119>>.
- FISHER, M. L.; JAIKUMAR, R. A generalized assignment heuristic for vehicle routing. **Networks**, v. 11, n. 2, p. 109–124, 1981. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/net.3230110205>>.
- GLOVER, F. Future paths for integer programming and links to artificial intelligence. **Computers Operations Research**, v. 13, n. 5, p. 533–549, 1986. ISSN 0305-0548. Applications of Integer Programming. Disponível em: <<https://www.sciencedirect.com/science/article/pii/0305054886900481>>.
- GOCKEN, T.; YAKTUBAY, M. Comparison of different clustering algorithms via genetic algorithm for vrptw. **INTERNATIONAL JOURNAL OF SIMULATION MODELLING/DAAAM INTERNATIONAL VIENNA**, 2019.

GUEZOULI, L.; BENSACKHRIA, M.; ABDELHAMID, S. Efficient golden-ball algorithm based clustering to solve the multi-depot vrp with time windows. **International Journal of Applied Evolutionary Computation (IJAEC)**, IGI Global, v. 9, n. 1, p. 1–16, 2018.

LIM, H.; LEE, G. M.; SINGGIH, I. K. Multi-depot split-delivery vehicle routing problem. **IEEE Access**, v. 9, p. 112206–112220, 2021.

MACQUEEN, J. *et al.* Some methods for classification and analysis of multivariate observations. In: OAKLAND, CA, USA. **Proceedings of the fifth Berkeley symposium on mathematical statistics and probability**. [S.l.], 1967. v. 1, n. 14, p. 281–297.

MIRHASSANI, S.; SAADATI, Z. A reactive adaptive memory metaheuristic for solving hffvrp. **Production manufacturing research**, Taylor Francis, v. 2, n. 1, p. 805–830, 2014. ISSN 2169-3277.

OSABA, E.; DÍAZ, F.; ONIEVA, E. Golden ball: A novel meta-heuristic to solve combinatorial optimization problems based on soccer concepts. **Applied Intelligence**, v. 41, p. 145–166, 12 2013.

PINHEIRO, C.; FERNANDES, C.; TAGLIALENHA, S. Lopes de S. Algoritmo simulated annealing para roteirização de veículos em uma empresa de outsourcing simulated annealing algorithm for vehicle routing in a outsourcing company. **Colloquium Exactarum**, v. 11, p. 1–16, 01 2020.

SCHULZ, A. A. H.; SCARPIN, C.; SOUZA, A. S. de. Uma abordagem meta heurística para o problema do vrptw. 2019.

SÖRENSEN, K.; GLOVER, F. Metaheuristics. In: _____. [S.l.: s.n.], 2013. p. 960–970. ISBN 978-1-4419-1137-7.

THANGIAH, S.; NYGARD, K.; JUELL, P. Gideon: a genetic algorithm system for vehicle routing with time windows. i, p. 322–328, 1991.