



Universidade do Minho
Escola de Engenharia

José Miguel dos Santos Pereira

**Uma Abordagem à Resolução do Capacitated
Vehicle Routing Problem Utilizando uma
Meta-Heurística Baseada em Recozimento
Quântico**

Outubro de 2022



Universidade do Minho
Escola de Engenharia

José Miguel dos Santos Pereira

**Uma Abordagem à Resolução do Capacitated
Vehicle Routing Problem Utilizando uma
Meta-Heurística Baseada em Recozimento
Quântico**

Dissertação de Mestrado

Mestrado em Engenharia de Sistemas

Trabalho efetuado sob a orientação do

Professor Doutor José António Vasconcelos Oliveira

Outubro de 2022

DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

Este é um trabalho académico que pode ser utilizado por terceiros desde que respeitadas as regras e boas práticas internacionalmente aceites, no que concerne aos direitos de autor e direitos conexos.

Assim, o presente trabalho pode ser utilizado nos termos previstos na licença abaixo indicada.

Caso o utilizador necessite de permissão para poder fazer um uso do trabalho em condições não previstas no licenciamento indicado, deverá contactar o autor, através do RepositóriUM da Universidade do Minho.

Licença concedida aos utilizadores deste trabalho



Atribuição

CC BY

<https://creativecommons.org/licenses/by/4.0/>

AGRADECIMENTOS

O meu primeiro agradecimento é dedicado aos meus pais, que sempre puseram o meu percurso escolar e académico no topo das suas prioridades. Apesar das dificuldades, sempre me foi dada liberdade de escolha e todas as condições necessárias ao meu sucesso, e sinto-me privilegiado por isso.

À Vanda agradeço por todo o apoio e por me fazer sentir seguro e capaz de atingir todos os meus objetivos. Sem dúvida que não podia ter melhor pessoa ao meu lado.

Esta dissertação não seria possível sem a coordenação excecional do Professor Doutor José António Vasconcelos Oliveira, que me auxiliou em todas as minhas dúvidas com prontidão e que partilhou comigo muito do seu conhecimento. Obrigado por acreditar neste projeto e por me motivar em todas as fases do mesmo.

Não podia deixar de agradecer à minha irmã Dany por tudo o que fez por mim e por ter o melhor coração que conheço.

Aos meus amigos Tonecas agradeço por sempre me apoiarem no meu percurso académico e por todos os momentos que vivemos juntos. Agradeço especialmente ao António e ao Eduardo que me acompanharam neste mestrado e que sempre me auxiliaram em tudo.

Muito obrigado aos futuros engenheiros e grandes amigos Pedro e Diogo, ao Bananas, à Bárbara e ao Zinho por estarem sempre comigo em todas as ocasiões e por todo o companheirismo.

Esta dissertação foi inteiramente produzida no polo de Riba de Ave da Biblioteca Municipal Camilo Castelo Branco e no Café Latitude. Assim sendo, tenho de agradecer aos funcionários da biblioteca por serem tão atenciosos e prestáveis, e ao Sr. Pedro do Café Latitude por toda a motivação e por todas as conversas e ensinamentos que certamente levarei para a vida.

DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

RESUMO

Uma Abordagem à Resolução do Capacitated Vehicle Routing Problem Utilizando uma Meta-Heurística Baseada em Recozimento Quântico

A presente dissertação foi desenvolvida no âmbito da obtenção do grau de Mestre em Engenharia de Sistemas. Devido à sua atualidade e relevância prática, decidiu-se estudar o Problema de Encaminhamento de Veículos (*Vehicle Routing Problem* - VRP) e desenvolver e implementar uma meta-heurística baseada em recozimento quântico para a obtenção de soluções para o VRP.

A revisão de literatura incidiu, fundamentalmente, nas variantes e nos métodos de resolução do VRP, permitindo obter conhecimento do estado da arte a ser incluído no algoritmo a desenvolver. Também da literatura, selecionaram-se aleatoriamente 40 instâncias públicas, com o cuidado de representarem uma grande amplitude dimensional, incluindo desde instâncias fáceis até instâncias de muito difícil solução.

Previamente à implementação da meta-heurística baseada em recozimento quântico implementou-se em Python 3 o Algoritmo de Poupanças de Clarke and Wright. Esta implementação permitiu ter uma base de comparação para o algoritmo seguinte e permitiu um primeiro contacto com a implementação de algoritmos para a resolução de problemas de Otimização Combinatória. Relativamente aos resultados computacionais, obteve-se, em média, um desvio percentual face à solução ótima de 672.35%, e que foi alcançado em 7 horas de processamento. Apesar do algoritmo ter encontrado uma solução viável para todas as instâncias, destacam-se a fragilidade da eficácia do método e a maior velocidade do algoritmo.

O algoritmo baseado em recozimento quântico, implementado também em Python 3, é baseado nos conceitos de mecânica quântica e não necessita de qualquer hardware quântico para a sua resolução. O principal conceito utilizado é o tunelamento quântico. Foram realizados vários processamentos com diferentes combinações de valores para os parâmetros relativos ao recozimento quântico, sendo que o algoritmo apenas foi capaz de solucionar, em tempos computacionais razoáveis, 26 das 40 instâncias escolhidas. Analisaram-se os resultados do algoritmo comparando-os com a solução ótima e com o Algoritmo de Poupanças de Clarke and Wright. Obteve-se, no melhor dos processamentos, um desvio percentual face à solução ótima de 166.90% e um desvio percentual face ao Algoritmo de Poupanças de Clarke and Wright de -19.93% ao fim de 35 horas de processamento.

PALAVRAS-CHAVE

Heurísticas, Logística, Recozimento Quântico, Vehicle Routing Problem

ABSTRACT

An Approach for Solving Capacitated Vehicle Routing Problem Using a Quantum-Annealing-Based Meta-Heuristic

This dissertation was developed to achieve a master's degree in Systems Engineering. Due to its actuality and practical relevance, it was decided to study the Vehicle Routing Problem (VRP) and to develop and implement a quantum-annealing-based meta-heuristic for obtaining solutions for the VRP.

The bibliographical review addressed essentially the VRP's variants and solution methods, and it provided knowledge about the state of the art about the topic to be included in the algorithm to be developed. Additionally, 40 public instances were selected, with the caution that they represent a big dimensional amplitude, including easy-solving instances and very hard-solving ones.

Previously to the implementation of the quantum-annealing-based meta-heuristic, Clarke & Wright's Savings Algorithm was implemented. This first implementation gave a comparative base for the following algorithm and allowed a first contact with the implementation of Combinatorial Optimization solving algorithms. On the computational results, it was obtained a mean percentual deviation facing the optimal solution of 672.35% by the end of 7 hours of processing runs. Although the algorithm found a viable solution to all 40 selected instances, the weakness of the method stands out, and also the faster velocity of the algorithm which was implemented using Python 3.

The quantum-annealing-based algorithm is based on quantum mechanical concepts and doesn't need any quantum hardware for its resolution. The main concept used was quantum-tunnelling. The implementation of the algorithm was made using the language Python 3. Various processing runs were made, using different values to the quantum-annealing parameters. The algorithm could only solve 26 of the 40 instances in reasonable computing times. The results of the algorithm were analysed in confront with the optimal solution and Clarke & Wright's Savings Algorithm results. In the best processing run a percentual deviation facing the optimal solution of 166.90% was obtained, and a percentual deviation facing Clarke & Wright's Savings Algorithm of -19.93%, being needed 35 hours to process all instances.

KEYWORDS

Heuristics, Logistics, Quantum-Annealing, Vehicle Routing Problem

ÍNDICE

Agradecimentos.....	iii
Resumo.....	v
Abstract.....	vi
Índice.....	vii
Índice de Figuras.....	x
Índice de Tabelas.....	xi
Índice de Equações.....	xii
Lista de Abreviaturas, Siglas e Acrónimos.....	xiii
1. Introdução.....	1
1.1 Enquadramento.....	1
1.2 Objetivos.....	2
1.3 Estrutura da Dissertação.....	3
2. Revisão de Literatura.....	4
2.1 Conceitos Básicos.....	4
2.2 Variantes do VRP.....	4
2.2.1 VRP com Veículos Capacitados (<i>Capacitated Vehicle Routing Problem – CVRP</i>).....	5
2.2.2 VRP com Janelas Temporais (<i>Vehicle Routing Problem with Time Windows – VRPTW</i>) ..	5
2.2.3 Problema das Entregas e Recolhas (<i>Pickup and Delivery Problem - PDP</i>).....	6
2.2.4 VRP com frota heterogênea (<i>Heterogeneous Fleet VRP – HF-VRP</i>).....	6
2.2.5 VRP com Múltiplos Depósitos (<i>Multiple Depots VRP – MDVRP</i>).....	6
2.2.6 VRP Periódico (<i>Periodic VRP – PVRP</i>).....	7
2.3 Métodos de resolução.....	7
2.3.1 Métodos de solução exata.....	10
2.3.1.1 Branch-and-Bound.....	10
2.3.1.2 Branch-and-Cut.....	10
2.3.2 Heurísticas.....	11
2.3.2.1 Heurísticas construtivas.....	11

2.3.2.1.1	Algoritmo de Poupanças de Clarke and Wright	12
2.3.2.1.2	Heurísticas de Inserção Sequencial	13
2.3.2.2	Heurísticas de melhoria baseadas em pesquisa local	13
2.3.2.2.1	Melhorias intra-rota.....	13
2.3.2.2.2	Melhorias inter-rotas	14
2.3.2.3	Heurísticas de duas fases	16
2.3.2.3.1	<i>Cluster-First Route-Second</i>	16
2.3.2.3.2	<i>Route-First Cluster-Second</i>	20
2.3.3	Meta-heurísticas	21
2.3.3.1	Meta-heurísticas Baseadas em Pesquisa Local	21
2.3.3.1.1	Pesquisa Tabu.....	22
2.3.3.1.2	Recozimento Simulado	22
2.3.3.2	Meta-heurísticas de Pesquisa Populacional.....	23
2.3.3.2.1	Algoritmos Genéticos	23
2.3.3.2.2	Colônia de Formigas.....	23
3.	Implementação de Metodologias de Solução	24
3.1	Algoritmo de Poupanças de Clarke and Wright.....	24
3.1.1	Introdução.....	24
3.1.2	O algoritmo	24
3.2	Algoritmo baseado em recozimento quântico para resolução do CVRP	25
3.2.1	Introdução.....	25
3.2.2	Recozimento Quântico	26
3.2.3	Configuração inicial	27
3.2.4	Cálculo de distâncias.....	27
3.2.5	Solução inicial	27
3.2.6	Recozimento Simulado	28
3.2.6.1	Tunelamento	28
3.2.6.2	Soluções de pior qualidade	29

3.2.7	Dados gerados pelo algoritmo	31
4.	Resultados experimentais.....	33
4.1	Instâncias utilizadas	33
4.2	Soluções obtidas pelo Algoritmo de Poupanças de Clarke and Wright.....	35
4.3	Soluções obtidas pelo Algoritmo Baseado em Recozimento Quântico	38
4.3.1	Limitações Computacionais e de Tempo de Processamento	38
4.3.2	Primeiro Processamento e Observações Iniciais	42
4.3.3	Segundo Processamento	47
4.3.4	Terceiro Processamento	49
4.3.5	Quarto Processamento	51
4.3.6	Avaliação da Prestação do Algoritmo	53
5.	Conclusão	55
5.1	Considerações finais	55
5.2	Limitações e Trabalho Futuro	56
	Referências Bibliográficas	59
	Apêndice 1 – Fluxograma: Algoritmo de Poupanças de Clarke and Wright	65
	Apêndice 2 – Fluxograma: Algoritmo Baseado em Recozimento Quântico	66
	Apêndice 3 – Pseudocódigo: Algoritmo Baseado em Recozimento Quântico	67
	Apêndice 4 – Resultados Obtidos pelo Primeiro Processamento	68
	Apêndice 5 – Resultados Obtidos pelo Segundo Processamento.....	69
	Apêndice 6 – Resultados Obtidos pelo Terceiro Processamento.....	70
	Apêndice 7 – Resultados Obtidos pelo Quarto Processamento.....	71

ÍNDICE DE FIGURAS

Figura 1 - Métodos de resolução do VRP	9
Figura 2 - Algoritmo de Poupanças de Clarke and Wright.....	12
Figura 3 - Mecanismo λ -opt de Lin	14
Figura 4 - String cross.....	15
Figura 5 - String Exchange	15
Figura 6 - String relocation.....	15
Figura 7 - Algoritmo Sweep: Situação inicial	17
Figura 8 - Algoritmo Sweep: Clustering.....	18
Figura 9 - Algoritmo de Fisher e Jaikumar: Situação inicial.....	19
Figura 10 - Algoritmo de Fisher e Jaikumar: Clustering	20
Figura 11 - Tunelamento Quântico	26
Figura 12 - Fluxograma: Solução inicial	28
Figura 13 - Fluxograma: Tunelamento.....	29
Figura 14 - Processo de aceitação de soluções com pior qualidade	31
Figura 15 - Gráfico de Dispersão: Relação entre o n° de clientes e o desvio percentual face à solução ótima	37
Figura 16 - Gráfico de Dispersão: Relação entre o n° de clientes e o tempo de processamento	38
Figura 17 - Gráfico de dispersão: Relação entre a percentagem de túneis aceites com soluções de pior qualidade e o desvio percentual face à solução ótima.....	45
Figura 18 - Gráfico de dispersão: Relação entre a temperatura final e o desvio percentual face à solução ótima	45
Figura 19 - Gráfico de dispersão: Relação entre a percentagem de iterações em que existiu um tunelamento e o desvio percentual face à solução ótima	46
Figura 20 - Fluxograma: APCW	65
Figura 21 - Fluxograma: ABRQ.....	66

ÍNDICE DE TABELAS

Tabela 1 - Número de trocas por tunelamento (exemplo).....	29
Tabela 2 - Evolução da probabilidade de aceitação (cenário 1)	30
Tabela 3 - Evolução da probabilidade de aceitação (cenário 2)	31
Tabela 4 - Dados das instâncias utilizadas.....	34
Tabela 5 - Soluções obtidas pelo APCW.....	36
Tabela 6 - Diferenças no Desvio Médio e no Tempo de Processamento com e sem a instância X-n655-k131	37
Tabela 7 - Características das instâncias que afetam o processamento.....	40
Tabela 8 - Resultados do primeiro processamento.....	43
Tabela 9 - Resultados do segundo processamento	48
Tabela 10 - Valores de alpha ajustados ao número de iterações	49
Tabela 11 - Resultados do terceiro processamento.....	50
Tabela 12 - Resultados do quarto processamento	52
Tabela 13 - Resultados completos do primeiro processamento	68
Tabela 14 - Resultados completos do segundo processamento.....	69
Tabela 15 - Resultados completos do terceiro processamento	70
Tabela 16 - Resultados completos do quarto processamento.....	71

ÍNDICE DE EQUAÇÕES

Equação 1 - Fórmula de cálculo de poupanças.....	12
Equação 2 - Fórmula de cálculo de distâncias no plano cartesiano	27
Equação 3 – Fórmula de cálculo do número de trocas por tunelamento	28
Equação 4 - Fórmula de cálculo da probabilidade de aceitação	30
Equação 5 - Fórmula de cálculo do desvio percentual face à solução ótima	43
Equação 6 - Fórmula de cálculo do desvio percentual face à solução obtida no APCW	43

LISTA DE ABREVIATURAS, SIGLAS E ACRÓNIMOS

ABRQ – Algoritmo Baseado em Recozimento Quântico

APCW – Algoritmo de Poupanças de Clarke & Wright

CVRP – *Capacitated Vehicle Routing Problem*

HF-VRP – *Heterogeneous Fleet VRP*

MDVRP – *Multiple Depots VRP*

PDP – *Pickup and Delivery Problem*

PVRP – *Periodic VRP*

TSP – *Traveling-Salesman Problem*

VRP – *Vehicle Routing Problem*

VRPTW – *Vehicle Routing Problem with Time Windows*

1. INTRODUÇÃO

Neste primeiro capítulo introduz-se o tema em estudo, fazendo-se um breve enquadramento do mesmo. De seguida, apresentam-se os principais objetivos desta dissertação e, por fim, indica-se como a mesma está estruturada.

1.1 Enquadramento

Atualmente, e devido à crescente globalização, a concorrência entre empresas dos mais variados setores torna-se cada vez mais acentuada. A redução de custos torna-se um fator decisivo para a sobrevivência e competitividade de um negócio. Uma das abordagens que as empresas utilizam para reduzir o preço dos produtos é reconfigurar a sua cadeia de abastecimento, nomeadamente a sua política de transportes. Fazer mudanças na cadeia de abastecimento pode ajudar na redução de custos e permitir que a empresa compita mais facilmente com base no preço (Rao Tummala et al., 2006). Custos relacionados com a cadeia de abastecimento rondam os 55% do custo final do produto (Lapinskaitė & Kuckailytė, 2014), sendo os custos de transporte os que têm maior representatividade, chegando a um ou dois terços dos custos logísticos totais (Ballou, 2008). Tendo ainda em consideração o progressivo aumento do preço dos combustíveis, torna-se fulcral para qualquer empresa que realize a distribuição dos seus produtos diminuir os seus encargos com transporte. Por todas estas razões, a otimização de rotas têm-se tornado um problema central já que permite uma redução de custos numa operação quotidiana de muitas empresas - a distribuição.

O problema de encaminhamento de veículos (*Vehicle Routing Problem* - VRP) procura determinar um conjunto ótimo de rotas a ser percorrido por uma frota de veículos servindo vários clientes, sendo um dos mais importantes e mais estudados problemas de otimização combinatória (Toth & Vigo, 2014).

Dantzig & Ramser (1959) desenvolveram uma primeira formulação matemática para o VRP sob o nome "*Truck Dispatching Problem*" (Goel & Maini, 2017). O problema foi apresentado como uma generalização do problema do caixeiro-viajante (*Traveling-Salesman Problem* - TSP), em que o caixeiro tem de voltar ao ponto inicial (Dantzig & Ramser, 1959). Ainda assim, a designação "VRP" apenas foi utilizada pela primeira vez anos mais tarde por Golden & Stewart, (1975). Desde então, centenas de artigos têm-se dedicado a encontrar soluções exatas ou aproximadas para as várias variantes deste problema (Baldacci et al., 2008).

O grande interesse da comunidade científica internacional nas diferentes variantes do VRP é motivado não só pela sua notória dificuldade enquanto problema de otimização combinatória, mas também pela sua relevância prática (Toth & Vigo, 2014). Recentemente, e com o advento das tecnologias de computação quântica, surgiram alguns métodos de resolução baseados nestas novas ferramentas. O recozimento quântico é um desses métodos que, com base em princípios da mecânica quântica, é utilizado na resolução de problemas de otimização combinatória (Kadowaki & Nishimori, 1998).

Como as tecnologias quânticas estão indisponíveis para a maior parte dos investigadores, surgiram nos últimos anos métodos de resolução do VRP que, apesar de não utilizarem qualquer tipo de ferramenta quântica, baseiam-se nos seus princípios com vista a obter soluções de melhor qualidade.

Por se tratar de um problema bastante complexo, mas ao mesmo tempo atual e atrativo, esta dissertação surge com o intuito de estudar o VRP e de investigar implementação prática de um algoritmo baseado em recozimento quântico.

1.2 Objetivos

O presente projeto tem como objetivo uma compreensão profunda do tema em análise, conhecendo todas as suas particularidades, projetos já desenvolvidos e desafios que se colocam para o futuro. É também objetivo modelar e implementar algoritmos de resolução do VRP.

Assim sendo, têm-se como principais objetivos desta dissertação:

- Conhecer de forma profunda o VRP e os estudos já existentes na área;
- Compreender o processo de recozimento quântico e as suas implicações no estudo do VRP;
- Construir um algoritmo construtivo de resolução do VRP;
- Construir um algoritmo de resolução do VRP que se baseie em recozimento quântico;
- Analisar os resultados obtidos pelo algoritmo.

1.3 Estrutura da Dissertação

A presente dissertação encontra-se dividida em cinco capítulos. Neste primeiro capítulo fez-se um breve enquadramento do problema, referindo ainda os principais objetivos deste projeto.

No segundo capítulo é feita uma revisão da literatura dos temas a abordar, de forma a melhor se compreender esta área de estudo. Têm-se como focos centrais as principais variantes e os métodos de resolução do VRP.

No capítulo seguinte, descreve-se a implementação de dois algoritmos, indicando-se pormenorizadamente como os mesmos foram desenvolvidos.

No quarto capítulo são analisados os dados obtidos pelos algoritmos elaborados, descrevendo-se os processamentos efetuados e as suas consequências nos resultados.

Por fim, no quinto e último capítulo apresentam-se as principais conclusões desta dissertação, indicando ainda as suas principais limitações e recomendações de trabalho futuro.

2. REVISÃO DE LITERATURA

Neste capítulo apresenta-se uma revisão bibliográfica dos principais temas a abordar. Inicialmente, apresentam-se os conceitos básicos relativos ao VRP para um melhor entendimento dos tópicos seguintes. A seguir, na revisão de literatura propriamente dita, referem-se as principais variantes do VRP bem como os seus diferentes métodos de resolução.

2.1 Conceitos Básicos

Num problema típico de VRP existe um conjunto de clientes que deve ser servido pela frota de veículos disponível. Cada cliente é servido precisamente uma única vez e por apenas um veículo. Cada veículo começa e acaba a sua rota no depósito.

O objetivo do problema é encontrar, de acordo com as características dos clientes e da frota, a solução que minimize os custos totais de transporte.

A solução a obter é uma lista das rotas a percorrer por cada veículo, indicando especificamente quais são os clientes servidos em cada rota e em que ordem. Cada rota é percorrida por um e um só veículo que visita os clientes de forma sequencial.

O número de soluções possíveis para um VRP aumenta exponencialmente à medida que aumenta o número de clientes, tornando instâncias de grande dimensão muito difíceis de resolver.

Em termos de complexidade computacional, o VRP é classificado como NP-difícil. Este conjunto de problemas são reduzíveis num problema NP, que por sua vez são problemas verificáveis em tempo polinomial, isto é, a sua solução pode ser confirmada num tempo polinomial. Por outras palavras, um problema NP-difícil é tão ou mais difícil que um problema NP.

2.2 Variantes do VRP

Devido às diferentes características e particularidades que se encontram nos problemas do mundo real, um grande número de variantes do VRP tem sido definidas e estudadas pela comunidade científica. Assim, antes de se estudar ou modelar um problema real deste tipo, é essencial que se identifiquem as características específicas do mesmo e, por consequência, a variante que melhor se adequa ao problema em causa.

Elshaer & Awad, (2020) apuraram quais as variantes mais abordadas pela comunidade científica no período de 2009 a 2017. Retratar-se-ão, de seguida, as variantes com maior relevância segundo esse estudo.

2.2.1 VRP com Veículos Capacitados (*Capacitated Vehicle Routing Problem – CVRP*)

O CVRP é a variante mais estudada do VRP apesar da sua relevância ser, normalmente, puramente académica (Toth & Vigo, 2014) devido à sua simplicidade.

A principal característica do CVRP é a existência de uma capacidade limitada em cada veículo. Para além disso, toda a frota é idêntica e existe apenas um único depósito. A procura dos clientes é conhecida antecipadamente, ou seja, é determinística. O objetivo fundamental do CVRP é a minimização do custo operacional dos veículos (Zulfiqar et al., 2018).

A frota deve visitar os clientes um por um satisfazendo a sua procura na totalidade e retornar ao depósito assim que entregar toda a sua carga (Ardon, 2022).

2.2.2 VRP com Janelas Temporais (*Vehicle Routing Problem with Time Windows – VRPTW*)

O VRPTW é uma variante do VRP em que o serviço a cada cliente começa num determinado intervalo de tempo, chamado de janela temporal (Kallehauge et al., 2006). Este problema é adotado quando os clientes apresentam restrições de horário para o seu atendimento.

As principais utilizações do VRPTW prendem-se com entregas em supermercados, bancos, serviços de correio, recolha de lixo, rotas de autocarros escolares, rondas de segurança e distribuição urbana de jornais (Kallehauge et al., 2006).

As janelas temporais são designadas como *hard* se não puderem ser violadas. Se um veículo chegar demasiado cedo a um cliente, deve esperar até que a janela temporal inicie e, da mesma forma, não é permitido que o veículo chegue depois do fecho da janela temporal. Por outro lado, as janelas temporais são classificadas como *soft* se puderem ser violadas tendo um custo como penalidade (Kallehauge et al., 2006).

2.2.3 Problema das Entregas e Recolhas (*Pickup and Delivery Problem - PDP*)

No PDP os itens a transportar não estão originalmente concentrados em depósitos, mas sim distribuídos pelos nodos da rede (Rizzoli et al., 2007). Assim, um certo número de bens deve ser transportado de uma certa localização de recolha até uma localização de entrega (Elshaer & Awad, 2020), não existindo por isso um depósito comum.

Este tipo de problema é utilizado, por exemplo, na reposição de stocks entre lojas de retalho, na gestão de sistemas de partilha de carros ou bicicletas, na distribuição de bebidas e recolha de garrafas vazias ou em serviços urbanos de correio (Toth & Vigo, 2014).

A maioria das instâncias do PDP são bastante mais difíceis de resolver que instâncias do mesmo tamanho do VRP clássico (Toth & Vigo, 2014), já que todos os nodos podem ser de recolha ou de entrega.

Este tipo de problemas inclui, habitualmente, janelas temporais quer para a entrega quer para a recolha (Rizzoli et al., 2007).

2.2.4 VRP com frota heterogênea (*Heterogeneous Fleet VRP – HF-VRP*)

Normalmente, as empresas dispõem de veículos com diferentes características para realizar as suas entregas, podendo a frota ser ampliada ou reduzida progressivamente ao longo do tempo. O HF-VRP considera que os veículos da frota podem divergir em termos de capacidade, custos fixos e variáveis, velocidade e clientes a que os mesmos podem aceder (Toth & Vigo, 2014).

Prins (2009) sugere uma divisão para o HF-VRP: quando existe um número ilimitado de veículos de cada tipo, estamos perante um VRP com frota de tamanho misto (*Fleet Size and Mix Vehicle Routing Problem*); por outro lado, o VRP com frota heterogênea (*Heterogeneous Vehicle Routing Problem*) considera uma frota fixa e conhecida previamente.

2.2.5 VRP com Múltiplos Depósitos (*Multiple Depots VRP – MDVRP*)

No MDVRP os veículos começam e terminam as suas rotas a partir de um dos diversos depósitos disponíveis (Ramos et al., 2020), tornando-se uma variante de extrema relevância para empresas de grande dimensão e com vários centros de distribuição.

Salienta-se que cada rota começa e termina no mesmo depósito e que nem todos os depósitos são obrigatoriamente utilizados (Renaud et al., 1996). Esta variante tem aplicações práticas na entrega de refeições, produtos químicos, bebidas, máquinas, gases industriais, produtos petrolíferos, alimentos pré-embalados, entre outros (Renaud et al., 1996).

2.2.6 VRP Periódico (*Periodic VRP* – PVRP)

O PVRP é uma variante em que o serviço aos clientes deve ser prestado em múltiplos períodos. Assume-se que os clientes devem ser servidos numa certa frequência e que recebem uma quantidade fixa de produtos em cada visita (Archetti et al., 2017).

O PVRP deve ser dividido em dois níveis de planeamento (Toth & Vigo, 2014): no primeiro nível, deverá ser selecionada a periodicidade das entregas para cada cliente, de acordo com as suas necessidades; no segundo nível, deverá ser resolvido um VRP onde constam os clientes que devem ser servidos numa determinada data, resultantes do planeamento do nível anterior.

Esta variante foi estudada, por exemplo, para a manutenção periódica de elevadores, recolha de lixo e entrega de sangue em hospitais (Coene et al., 2010).

2.3 Métodos de resolução

O VRP é um dos tópicos mais estudados no campo da Investigação Operacional. Desta forma, está disponível na literatura uma grande variedade de estratégias de resolução do VRP (Goel & Maini, 2017).

Os métodos de resolução para o VRP podem ser divididos em métodos exatos, métodos heurísticos e meta-heurísticas (Tan & Yeh, 2021).

Os **métodos exatos** fornecem uma solução ótima para o problema. Contudo, atualmente apenas instâncias de pequena dimensão podem ser resolvidas de forma ótima (Toth & Vigo, 2014). Como o VRP é considerado NP-Difícil, isto é, a dificuldade para encontrar a solução ótima cresce exponencialmente à medida que o número de clientes aumenta (Kramer et al., 2015), resolver de forma exata instâncias de grande dimensão torna-se uma tarefa bastante morosa (Tan & Yeh, 2021).

Todavia, as instâncias utilizadas na vida real são, por vezes, de grande dimensão e devem ser resolvidas rapidamente (Toth & Vigo, 2014). Assim sendo, são frequentemente utilizados **métodos heurísticos**, que procuram encontrar uma solução aceitável para o problema num número limitado de iterações (Goel & Maini, 2017), sendo por isso um método de solução aproximada, isto é, não garante a otimalidade da solução.

Já as **meta-heurísticas** adaptam soluções genéricas e utilizam-nas em problemas específicos. Para além de requererem muito menos trabalho a desenvolver do que uma heurística, as meta-heurísticas podem encontrar soluções quase ótimas em tempos computacionais razoáveis. Golden et al., (2008) constataram que o VRP em particular registava um recorde de implementações bem-sucedidas de meta-heurísticas.

A Figura 1 - Métodos de resolução do VRP esquematiza os métodos de resolução que serão retratados nos pontos seguintes.

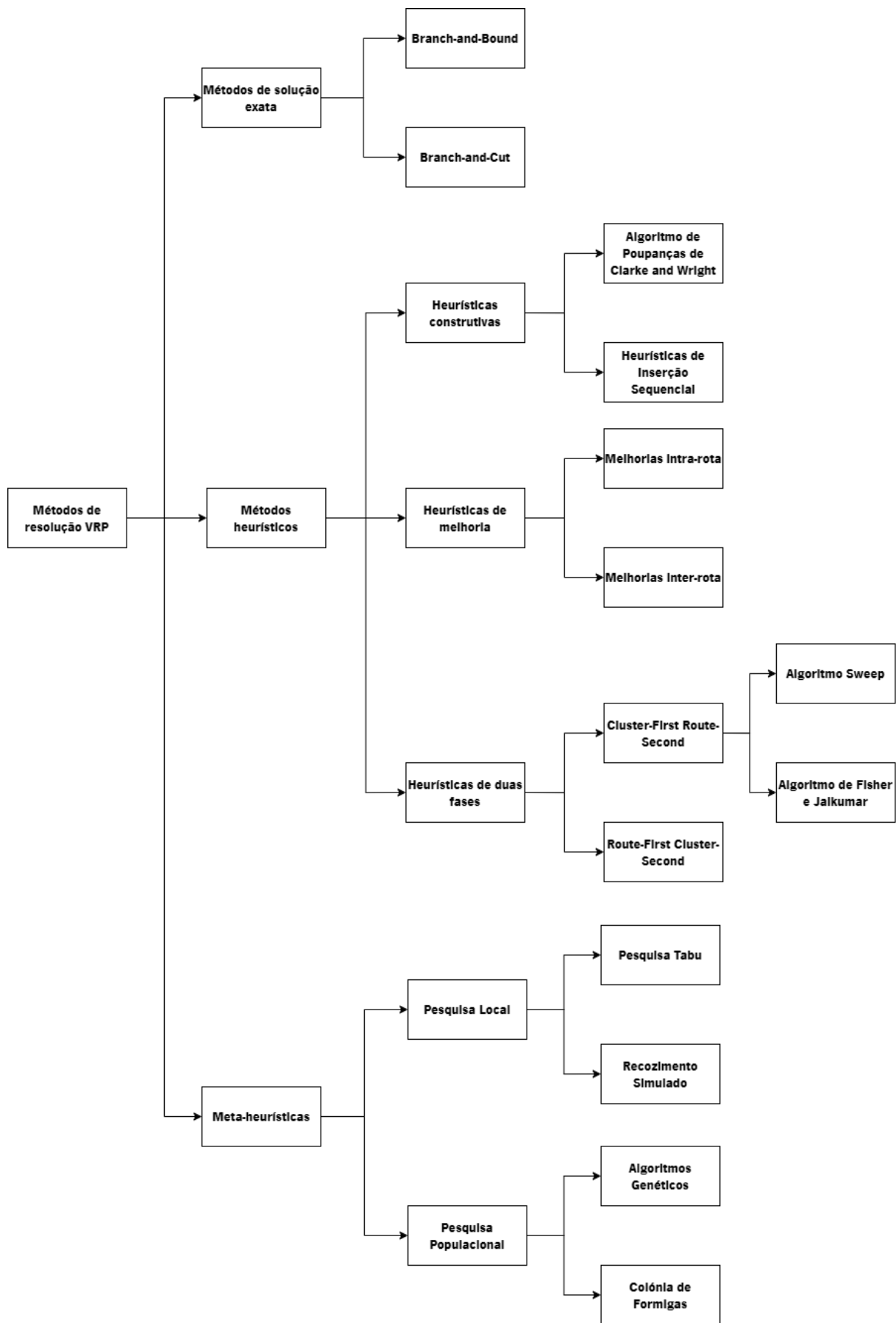


Figura 1 - Métodos de resolução do VRP

2.3.1 Métodos de solução exata

Os métodos exatos de resolução do VRP são algoritmos que procuram encontrar a solução ótima para o problema. Contudo, estes métodos apenas são eficientes para instâncias relativamente pequenas, devido à grande variância do seu tempo computacional. Segundo Toth & Vigo (2014), apenas são resolvíveis otimamente instâncias a rondar os 100 clientes.

Nos últimos anos, com o desenvolvimento dos tempos de processamento e da capacidade de memória dos computadores, a investigação de métodos exatos está em rápido crescimento. É expectável que no futuro cada vez mais métodos exatos sejam utilizados para a resolução do VRP e que o número de nodos possa aumentar consideravelmente (Tan & Yeh, 2021).

2.3.1.1 *Branch-and-Bound*

O *Branch-and-Bound* (Método de Partição e Avaliações Sucessivas) é uma abordagem baseada no princípio de que o conjunto das soluções de um problema pode ser dividido em subconjuntos mais pequenos, excluindo regiões de solução fracionárias e garantindo a não exclusão da solução ótima. Esses subconjuntos podem ser avaliados sistematicamente até que a solução ótima seja encontrada (Taylor, 2013).

Este método organiza as possíveis soluções numa árvore, onde cada nodo representa um dos subconjuntos. Para cada nodo são calculados os limites superior e inferior da função objetivo, indicando o espaço em que as soluções advindas desse ramo se encontram. Caso as margens calculadas não excluam a possibilidade desse subconjunto conter a solução ótima, esse nodo é dividido novamente (He et al., 2014).

Apesar de ser normalmente referido no contexto da programação inteira, o *Branch-and-Bound* pode ser utilizado para resolver muitos problemas de otimização combinatória (Kianfar, 2011).

2.3.1.2 *Branch-and-Cut*

O *Branch-and-Cut* pode ser visto como uma extensão do *Branch-and-Bound*. Sinteticamente, este método utiliza o mesmo *framework* que o *Branch-and-Bound*, mas em cada nodo são efetuados cortes adicionais ao espaço de soluções (Chen et al., 2011)

O intuito dos cortes é obter limites o mais estreitos possível de forma a reduzir o número de ramos da árvore. Em contrapartida, se muitos cortes forem aplicados, o processo de otimização pode tornar-se mais demorado (Roos, 2003).

Para além disso, e à semelhança do *Branch-and-Bound*, o algoritmo poderá ser interrompido a qualquer momento, produzindo uma solução viável em que os limites indicam a distância a que a solução atual poderá estar de ser ótima (Hoffman & Ralphs, 2013).

2.3.2 Heurísticas

Sendo o VRP um problema NP-hard e que não pode ser resolvido otimamente em tempo polinomial, vários estudos têm se dedicado à construção de heurísticas (Mirshekarian et al., 2015). As heurísticas não exploram todo o espaço de soluções, mas, ainda assim, conseguem produzir soluções bastante razoáveis e, acima de tudo, fazem-no em tempos computacionais bastante modestos. Para além disso, a maior parte das heurísticas pode ser adaptada à diversidade de fatores que se encontram em problemas da vida real (Toth & Vigo, 2002).

As heurísticas são comumente divididas em três tipos (Toth & Vigo, 2002):

- **Heurísticas construtivas**, que constroem gradualmente uma solução viável tendo em conta o custo dessa mesma solução;
- **Heurísticas de duas fases**, em que o problema é subdividido no *clustering* dos vértices em soluções viáveis e na construção propriamente dita das rotas;
- **Heurísticas de melhoria baseadas em pesquisa local**, que procuram melhorar uma solução viável através da troca sequencial de vértices.

2.3.2.1 Heurísticas construtivas

As heurísticas construtivas determinam uma solução de acordo com um conjunto de regras de construção definido previamente. O processo de construção é realizado passo a passo e as partes já construídas da solução mantêm-se geralmente inalteradas no decorrer do algoritmo (Reinelt, 1994).

Este tipo de heurísticas é geralmente utilizado de forma a obter uma solução inicial que posteriormente possa ser utilizada numa heurística de melhoria ou numa metaheurística (Avdoshin & Beresneva, 2019).

2.3.2.1.1 Algoritmo de Poupanças de Clarke and Wright

O Algoritmo de Poupanças de Clarke & Wright (APCW) é provavelmente a mais conhecida heurística para o VRP (Toth & Vigo, 2002). Este algoritmo foi apresentado em 1964 num artigo onde se pretendia obter as rotas ótimas para um conjunto de camiões de diferentes capacidades utilizados na entrega de produtos de um depósito para vários clientes.

O APCW constrói, inicialmente, rotas de ida e volta entre o depósito e todos os clientes. De seguida, duas rotas são fundidas de acordo com as poupanças que podem ser obtidas. Esta é uma heurística bastante simples e que tem as vantagens de ser rápida, intuitiva, e fácil de implementar (Toth & Vigo, 2002).

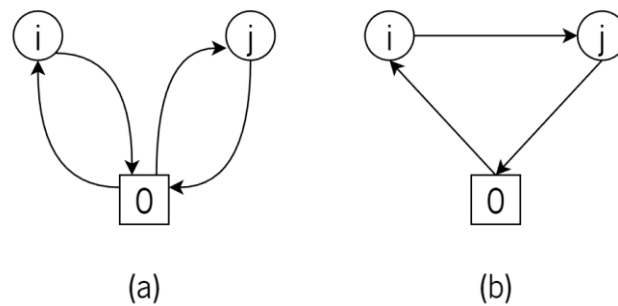


Figura 2 - Algoritmo de Poupanças de Clarke and Wright

A Figura 2 (a) representa a situação inicial do algoritmo, onde cada rota inclui apenas um cliente e o depósito. A distância total a ser percorrida seria duas vezes a distância entre o depósito e o cliente i mais duas vezes a distância entre o depósito e o cliente j . Esta situação é impraticável pois gera um elevado número de rotas e, conseqüentemente, um grande número de veículos necessários para as percorrer.

A Figura 2 (b) representa o resultado do APCW, onde os dois clientes i e j são atendidos numa mesma rota. A poupança z_{ij} obtida pela junção de duas rotas é obtida pela expressão:

Equação 1 - Fórmula de cálculo de poupanças

$$z_{ij} = d_{0i} + d_{0j} - d_{ij}$$

onde d_{0i} representa a distância entre o depósito e o cliente i , d_{0j} a distância entre o depósito e o cliente j e d_{ij} a distância entre o cliente i e o cliente j (Jeřábek et al., 2016). Caso o valor de z_{ij} seja negativo, então não existe vantagem em unir as duas rotas. É ainda de salientar que a junção das rotas apenas é realizada se a procura total não ultrapassar a capacidade do veículo e se não forem violadas quaisquer outras restrições (Pichpibul & Kawtummachai, 2013).

2.3.2.1.2 Heurísticas de Inserção Sequencial

Os algoritmos de inserção sequencial constroem uma solução tentando repetidamente inserir os clientes que ainda não pertencem a nenhuma rota numa rota em construção (Arvianto et al., 2019). O algoritmo constrói uma rota de cada vez até que todos os clientes sejam inseridos (Jawarneh & Abdullah, 2015).

O primeiro cliente a ser inserido numa rota é referido como o cliente *seed*. Assim que o cliente *seed* é identificado, o algoritmo considera, para os nodos que ainda não foram inclusos em nenhuma rota, qual o custo de incluir esses nodos na rota atual (Joubert & Claasen, 2006). Quando o custo de inserir um cliente numa nova rota for menor do que o inserir na já existente, o cliente será afetado a uma nova rota (Campbell & Savelsbergh, 2004).

Os vários algoritmos de inserção sequencial diferem, principalmente, na forma como é calculado o custo de inserção. Sendo esta uma heurística bastante utilizada para a resolução do VRPTW, é comum este custo ter em conta não só a distância a percorrer, mas também o tempo gasto nas deslocações e no tempo de serviço. Para detalhes adicionais sobre as heurísticas de inserção sequencial, recomenda-se a consulta de Jawarneh & Abdullah, (2015).

2.3.2.2 Heurísticas de melhoria baseadas em pesquisa local

Nas heurísticas de melhoria são aplicadas, em cada iteração, melhorias à solução inicial até que um determinado critério seja atingido. A solução inicial pode ser gerada aleatoriamente ou através de uma heurística de construção. Já o critério de paragem pode ser baseado no tempo de pesquisa, no número de iterações ou na estagnação do algoritmo ao fim de um certo número de iterações (Tavares et al., 2009).

Podem ser distinguidos dois tipos de heurísticas de melhoria, sendo eles as melhorias intra-rota e as melhorias inter-rotas (Breedam, 1995).

2.3.2.2.1 Melhorias intra-rota

Os algoritmos de melhoria intra-rota procuram diminuir a distância de uma rota alterando a sequência dos seus nodos (Breedam, 1995). Segundo Toth & Vigo, (2002), todas as heurísticas de melhoria intra-rota existentes para o problema do caixeiro-viajante podem também ser utilizadas para a resolução do VRP.

Grande parte dos procedimentos de melhoria intra-rota podem ser descritos nos termos do mecanismo λ -opt de Lin (1965). Um número λ de arcos é removido da rota e os λ segmentos resultantes são reconectados de todas as formas possíveis. Se existir alguma reconexão vantajosa, então a mesma é implementada. O procedimento termina num mínimo local onde não é possível obter mais melhorias (Toth & Vigo, 2002).

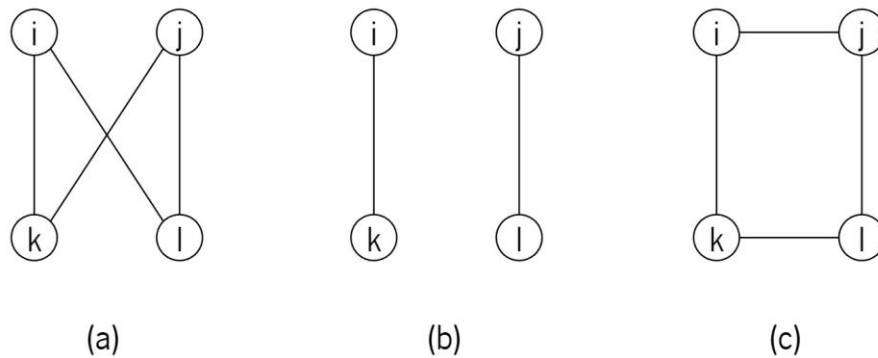


Figura 3 - Mecanismo λ -opt de Lin

Na Figura 3 está retratado um possível exemplo para o mecanismo λ -opt de Lin. Neste caso, λ foi definido como 2, ou seja, serão removidos 2 arcos da solução inicial, representada na Figura 3 (a). Na Figura 3 (b) é possível verificar que os dois arcos removidos foram o arco $i - l$ e o arco $k - j$. Os dois segmentos de rota restantes são, então, reconectados, tal como é possível verificar em Figura 3 (c). No exemplo em análise, a única solução alternativa é conectar os nodos i e j e os nodos k e l , contudo, caso existissem outras hipóteses de conexão, as mesmas seriam também consideradas. A nova rota está finalmente construída e será mantida caso seja mais curta que a rota anterior.

2.3.2.2 Melhorias inter-rotas

As heurísticas de melhoria inter-rotas consideram possíveis trocas ou realocações de nodos entre duas rotas de uma solução inicial (Breedam, 1995).

Vários tipos de trocas têm sido estudados por inúmeros autores, destacando-se a classificação proposta por (Breedam, 1995). À luz desta classificação, consideram-se quatro operações (Toth & Vigo, 2002):

- **String cross:** duas cadeias de vértices são trocadas, existindo o cruzamento de duas arestas de diferentes rotas (Figura 4);
- **String exchange:** duas cadeias de vértices são trocadas entre duas rotas (Figura 5);

- **String relocation.** uma cadeia de vértices é transferida de uma rota para outra (Figura 6);
- **String mix.** é selecionado o melhor movimento entre o string exchange e o string relocation.

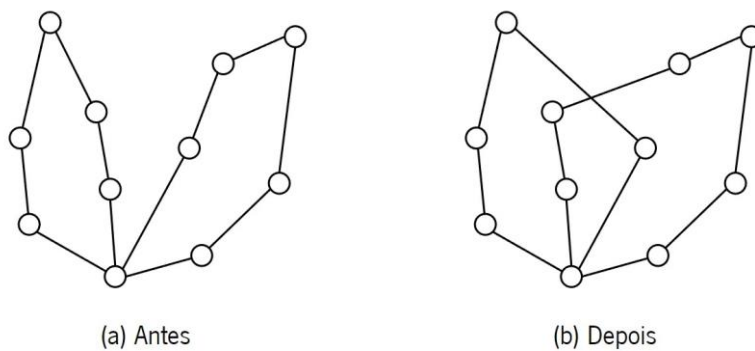


Figura 4 - String cross
(Adaptado de Toth & Vigo (2002))

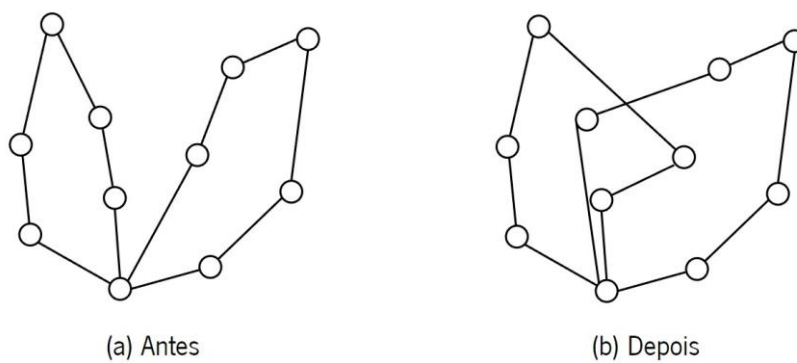


Figura 5 - String Exchange
(Adaptado de Toth & Vigo (2002))

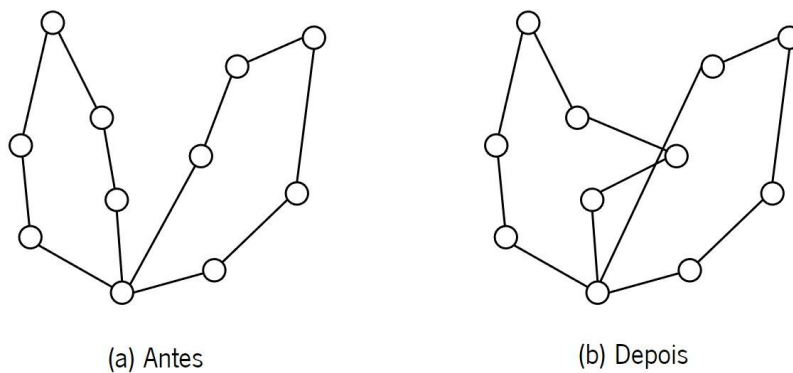


Figura 6 - String relocation
(Adaptado de Toth & Vigo (2002))

2.3.2.3 Heurísticas de duas fases

As heurísticas de duas fases subdividem o problema em duas etapas distintas: o *clustering* dos vértices e a construção das rotas (Toth & Vigo, 2002).

Assim sendo, certos algoritmos executam primeiro a fase de *clustering* dos vértices (algoritmos *Cluster-First Route-Second*) enquanto outros inicializam com a fase de construção da rota (algoritmos *Route-First Cluster-Second*).

2.3.2.3.1 *Cluster-First Route-Second*

Nos métodos *Cluster-First Route-Second* é realizado, inicialmente, um *clustering* dos nodos, sendo que a construção das rotas é realizada apenas na segunda fase.

Este tipo de heurísticas já é utilizado há muitos anos, contudo continua a ser aplicado devido à sua simplicidade (Garside & Laili, 2019).

Algoritmo *Sweep*

O algoritmo *Sweep* baseia-se no *clustering* dos clientes através das suas coordenadas polares (ou seja, no plano bidimensional), onde o depósito é considerado a coordenada central (0,0). Os clientes são agrupados pelo ângulo que formam com o depósito e tendo em conta a capacidade de transporte do veículo (Hanafi et al., 2020).

Na Figura 7 está representado um exemplo de aplicação do algoritmo *Sweep* com 11 clientes. Junto a cada nodo encontra-se a procura do cliente respetivo.

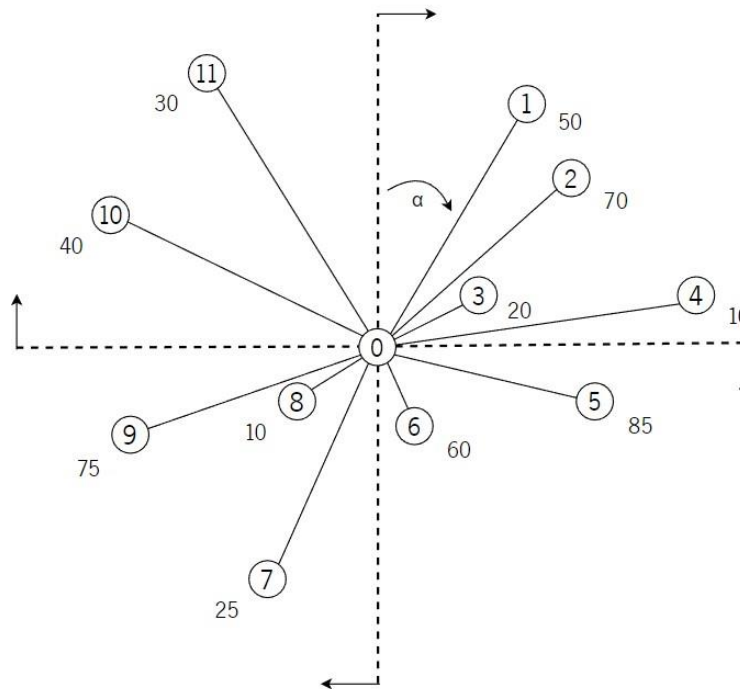


Figura 7 - Algoritmo Sweep: Situação inicial

Neste momento inicial, todos os nodos são conectados à coordenada central de forma a se verificar o ângulo formado entre os clientes e o depósito. No exemplo, a medição dos ângulos é iniciada no primeiro quadrante e é feita no sentido dos ponteiros do relógio. Contudo, tanto o quadrante inicial quanto a direção da medição são totalmente arbitrários, resultando, porém, em clusters diferentes. A título ilustrativo, representou-se o ângulo referente ao cliente 1 com α .

De seguida, deve proceder-se ao *clustering* dos nodos de acordo com a capacidade de transporte dos veículos. Se se assumir uma capacidade máxima de 150 unidades de mercadoria, os clusters obtidos são os exibidos na Figura 8.

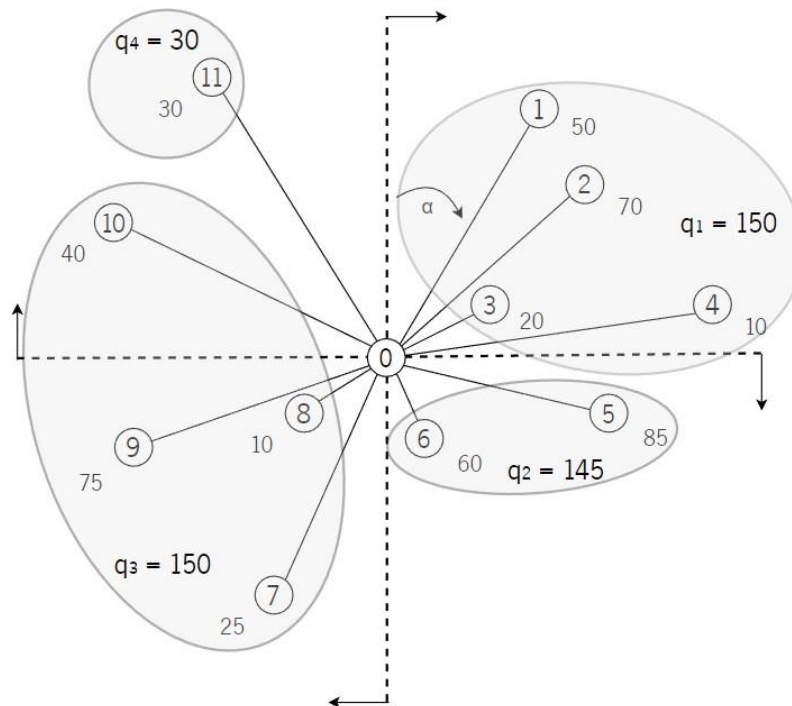


Figura 8 - Algoritmo Sweep: Clustering

Constata-se, então, que são criados quatro clusters de clientes, sendo que cada um deles originará uma rota. Nos clusters 1 e 3 a procura total é de 150 unidades (correspondendo à capacidade máxima de transporte), no cluster 2 de 145 unidades e no cluster 4 de apenas 30 unidades.

Por fim, para obtenção das rotas, dever-se-á resolver um TSP com os nodos de cada cluster e o depósito (Toth & Vigo, 2002).

Algoritmo de Fisher e Jaikumar

O algoritmo de Fisher e Jaikumar foi introduzido em 1981 pelos investigadores que lhe dão nome. O algoritmo trata o VRP como um Problema Generalizado de Atribuição. Este método, à semelhança do *Sweep*, também se baseia no plano bidimensional, todavia o número de veículos disponíveis e, conseqüentemente, o número final de rotas, são conhecidos à priori (Toth & Vigo, 2002).

Este algoritmo baseia-se na divisão do plano em diversos cones, sendo que o número de cones é igual ao número de veículos disponíveis.

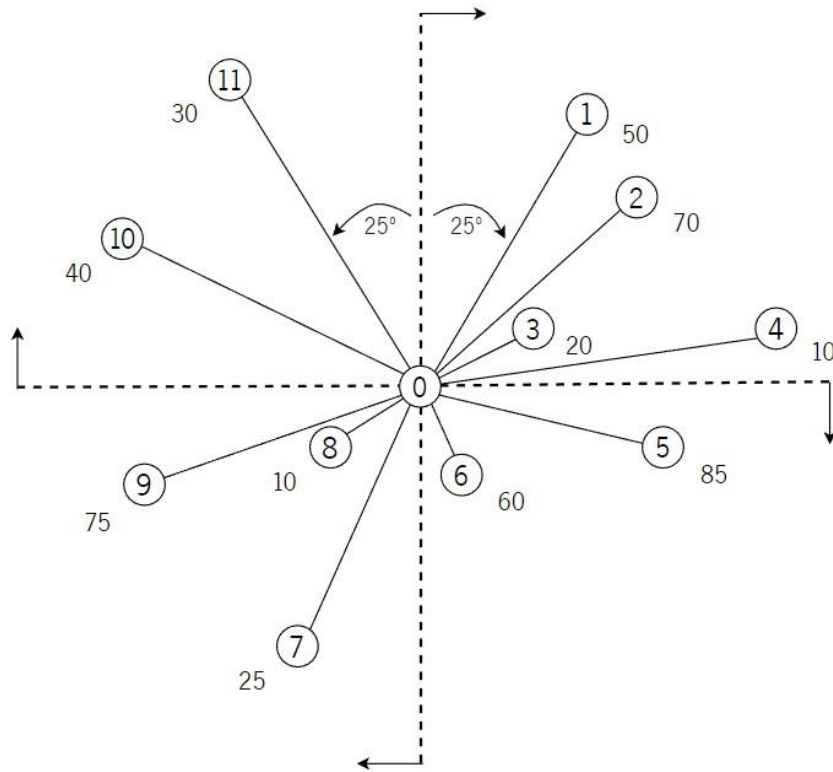


Figura 9 - Algoritmo de Fisher e Jaikumar: Situação inicial

Na Figura 9 é possível observar uma instância do problema com 11 clientes e onde existem 5 veículos disponíveis. O cliente com menor ângulo (cliente 1) perfaz 25° com o eixo das ordenadas, e o cliente com maior ângulo (cliente 11) perfaz 335° ($360^\circ - 25^\circ$). Assim sendo, o espaço a dividir será de apenas $335^\circ - 25^\circ = 310^\circ$. Como se sabe que o número de veículos disponíveis é de 5, então este espaço será dividido em 5 cones, cada um com $310^\circ / 5 = 62^\circ$. Os cones estão representados na Figura 10.

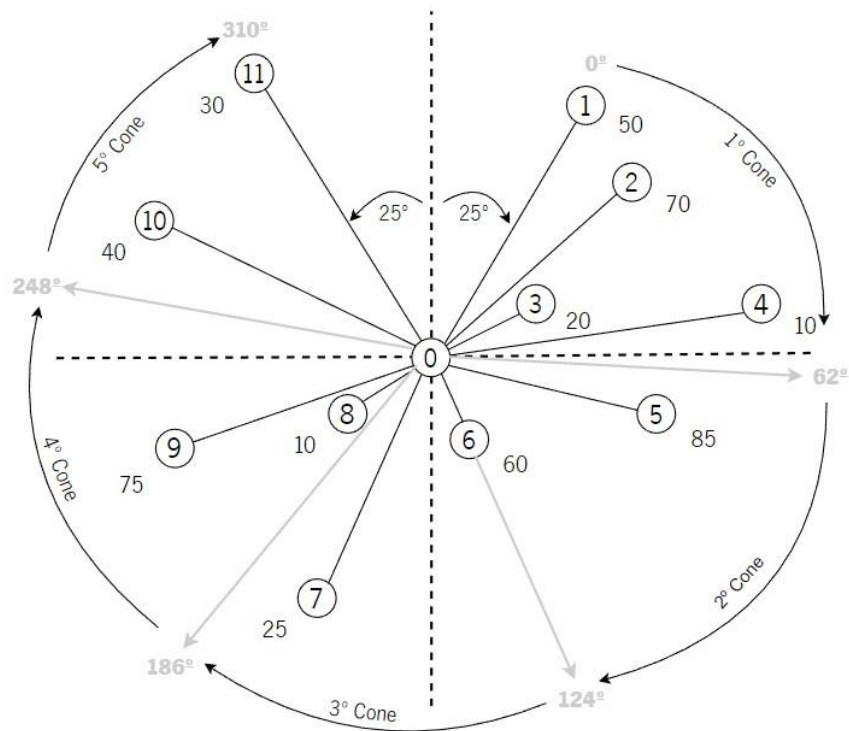


Figura 10 - Algoritmo de Fisher e Jaikumar: Clustering

Em cada um dos cones é selecionado um cliente como sendo o cliente *seed*, e é criada uma rota incluindo apenas o depósito e esse cliente. O cliente *seed* será o cliente com maior procura ou o cliente mais afastado do depósito. De seguida, é calculado o custo de inserção dos restantes clientes nas diferentes rotas. Os clientes são, finalmente, atribuídos às rotas de acordo com o seu custo de inserção, começando pelos custos mais reduzidos. Por último, para a construção final das rotas, resolve-se um TSP (Sultana et al., 2017).

2.3.2.3.2 Route-First Cluster-Second

Estes métodos constroem, numa primeira fase, uma rota gigante com todos os nodos e ao estilo das rotas do caixeiro-viajante. De seguida, na segunda fase, a rota é decomposta em rotas mais pequenas (Toth & Vigo, 2002).

De acordo com Beasley (1983), uma das grandes vantagens destes métodos é a simplicidade de dividir a rota gigante em rotas mais pequenas de forma ótima. Para além disso, o uso da rota gigante garante que clientes que se encontram próximos serão muito provavelmente servidos pela mesma rota.

Ainda assim, e apesar de todas estas vantagens, não existem resultados computacionais que comprovem que os métodos *Route-First Cluster-Second* sejam competitivos em comparação com outras metodologias (Toth & Vigo, 2002).

2.3.3 Meta-heurísticas

Tendo em consideração que instâncias de grandes dimensões do VRP tornam-se muito morosas de resolver de forma ótima, esse processo torna-se impraticável em contextos de vida real. Tal como as heurísticas, as meta-heurísticas tem como objetivo resolver instâncias do VRP em tempos computacionais razoáveis (Gendreau et al., 2008).

As meta-heurísticas são soluções genéricas - isto é, podem ser aplicadas à resolução de vários problemas - que exploram o espaço de soluções para tentar identificar soluções razoáveis (Toth & Vigo, 2002). O desafio prende-se, então, com a adaptação dessas soluções genéricas ao problema em causa (Gendreau et al., 2008).

Os dois tipos de meta-heurísticas mais relevantes são as meta-heurísticas baseadas em pesquisa local e as meta-heurísticas de pesquisa populacional.

2.3.3.1 Meta-heurísticas Baseadas em Pesquisa Local

Os algoritmos de pesquisa local exploram o espaço de soluções movendo a solução atual para outra solução na sua vizinhança (Toth & Vigo, 2002). Assim sendo, ao invés de se pesquisar sistematicamente todo o espaço de soluções, as mesmas são encontradas melhorando a solução atual através de pequenas modificações.

A vizinhança de uma solução engloba todas as soluções que são possíveis de alcançar com uma modificação à solução atual. Desta forma, a vizinhança de uma solução é diferente consoante o tipo de modificação que se pretenda efetuar. Os algoritmos de pesquisa local, após identificarem a vizinhança de uma solução, determinam qual a melhor solução vizinha, normalmente através do seu resultado na função objetivo. Quando uma solução não apresenta soluções vizinhas de qualidade superior, então é porque foi atingido um mínimo local (Arnold & Sörensen, 2019). Os algoritmos apresentam, normalmente, mecanismos que tentam evitar que a solução fique encurralada em mínimos locais (Toth & Vigo, 2002).

2.3.3.1.1 Pesquisa Tabu

A pesquisa tabu é uma meta-heurística que enfatiza a necessidade de os algoritmos não ficarem presos em mínimos locais (Barbarosoglu & Ozgur, 1999). Nos algoritmos de pesquisa tabu os movimentos são sempre realizados em direção à melhor solução vizinha, isto é, o algoritmo seleciona sempre a melhor solução da vizinhança da solução atual (Toth & Vigo, 2002).

Para evitar que a solução fique encurralada num mínimo local, as soluções já visitadas recentemente são incluídas numa lista – a lista tabu – que as proíbe de serem visitadas novamente por um certo número de iterações. É também habitual que a lista tabu inclua atributos das soluções e não as soluções propriamente ditas (Gendreau et al., 2008). Assim, caso uma solução contenha um desses atributos, não poderá ser selecionada.

2.3.3.1.2 Recozimento Simulado

A principal ideia que sustenta os algoritmos de recozimento simulado é a de aceitar, ocasionalmente, soluções de menor qualidade na tentativa de escapar de ótimos locais (Afifi et al., 2013). Desta forma, modificações à solução atual que produzam piores resultados poderão, ainda assim, ser aceites de acordo com uma certa probabilidade (Gendreau et al., 2008).

Os algoritmos de recozimento simulado surgem como analogia ao processo físico de recozimento, utilizado na metalurgia. O tratamento térmico de recozimento de um material consiste na sua exposição a uma temperatura elevada durante um período prolongado, sendo posteriormente arrefecido lentamente de uma forma específica. Este tratamento pode ser utilizado, por exemplo, com o objetivo de aliviar tensões ou de tornar o material mais mole (Jr. & Rethwisch, 2018).

Nas meta-heurísticas de recozimento simulado a solução atual é modificada a cada iteração. Caso a nova solução seja melhor que a solução atual, então torna-se a solução atual. Caso contrário, a nova solução poderá ser aceite de acordo com um critério probabilístico, definido pelo valor de um parâmetro, chamado temperatura (em analogia ao processo físico de recozimento). No decorrer das iterações a temperatura é reduzida progressivamente, o que significa que a probabilidade de se aceitar soluções de menor qualidade diminui. Ao contrário de outras meta-heurísticas, foi provado que o recozimento simulado converge para o ótimo global (Gendreau et al., 2008).

2.3.3.2 Meta-heurísticas de Pesquisa Populacional

Os algoritmos de pesquisa populacional têm como inspiração conceitos naturais, como é exemplo a evolução de espécies e o comportamento de insetos (Toth & Vigo, 2002). Ao contrário dos algoritmos de pesquisa local, este tipo de meta-heurística considera um conjunto de soluções que vai sendo progressivamente alterado (Ezzatneshan, 2010).

2.3.3.2.1 Algoritmos Genéticos

Os algoritmos genéticos são meta-heurísticas que procuram imitar a evolução das espécies de acordo com os princípios Darwinianos da seleção natural. É considerada uma população de soluções – também designada de cromossomas – que evoluem de geração em geração através da aplicação de operadores semelhantes aos encontrados na natureza, como a mutação, cruzamento genético ou a sobrevivência do mais apto. Neste processo, as melhores soluções têm maior probabilidade de serem selecionadas para serem as “progenitoras” das gerações seguintes, garantindo que os cromossomas aumentam a sua qualidade ao longo do algoritmo (Gendreau et al., 2008).

Uma das grandes vantagens dos algoritmos genéticos é a exploração do espaço de soluções em diferentes direções, já que os vários cromossomas da população se comportam como agentes individuais (Gandomi et al., 2013).

2.3.3.2.2 Colónia de Formigas

Os algoritmos de colónia de formigas foram introduzidos, tal como o nome indica, em analogia às colónias de formigas em busca por alimento. Na procura de fontes de alimentação, as formigas marcam o seu caminho com uma essência aromática chamada feromona. A quantidade de feromona deixada num certo caminho depende do comprimento do percurso e da qualidade da comida. Este fenómeno fornece informação às outras formigas, sendo que, ao longo do tempo, fontes de alimento mais recompensadoras - fontes em grande quantidade e perto do ninho – tornam-se mais frequentadas e marcadas com mais quantidade de feromona. Este processo gera uma procura de alimentos eficiente para toda a colónia (Toth & Vigo, 2002).

Colorni et al. (1991) propuseram esta meta-heurística para o TSP, sendo que a qualidade da comida é associada à função objetivo e os trilhos de feromona são simulados por valores que vão sendo progressivamente alterados.

3. IMPLEMENTAÇÃO DE METODOLOGIAS DE SOLUÇÃO

No presente capítulo são explanadas as implementações do Algoritmo de Poupanças de Clarke and Wright (APCW) e do Algoritmo Baseado em Recozimento Quântico (ABRQ). Ambos os algoritmos foram programados utilizando a linguagem *Python 3*.

3.1 Algoritmo de Poupanças de Clarke and Wright

3.1.1 Introdução

O primeiro algoritmo implementado neste trabalho foi o APCW. Optou-se por programar este algoritmo já que se trata, provavelmente, da mais conhecida heurística para o CVRP. Para além disso, é fácil de implementar e bastante rápido. A implementação deste algoritmo tem como objetivo a comparação do mesmo com o algoritmo baseado em recozimento quântico descrito posteriormente.

Sendo uma heurística muito conhecida, é possível consultar e utilizar diversas implementações da mesma. Ainda assim, optou-se por desenvolver uma versão própria.

3.1.2 O algoritmo

Inicialmente, antes da execução do algoritmo, o utilizador deve preparar um ficheiro que inclua as coordenadas dos clientes e do depósito e um outro onde se armazene a procura de cada nodo. A capacidade do camião deve ser inserida manualmente pelo utilizador no código.

O algoritmo inicia, precisamente, pela leitura dos ficheiros das coordenadas e da procura. De seguida, é produzida a matriz de distâncias euclidianas, que inclui a distância de cada cliente para os restantes e para o depósito. A matriz obtida é uma matriz simétrica, já que se pretende modelar o CVRP simétrico, em que a distância do cliente i para o cliente j é a mesma que do cliente j para o cliente i .

Seguidamente, é tarefa do algoritmo a criação da matriz das poupanças, essencial para a heurística. Pelo mesmo motivo da matriz das distâncias, esta será uma matriz simétrica. A fórmula de cálculo das poupanças é dada pela Equação 1.

Obtida a matriz das poupanças, são criadas as rotas iniciais, em que cada cliente é conectado ao depósito, tal como exemplificado na Figura 2 no ponto 2.3.2.1.1 Algoritmo de Poupanças de Clarke and Wright.

Posto isto, são de seguida ordenadas as poupanças em ordem decrescente. O algoritmo verifica, logo após, e começando pelos pares de clientes com maiores poupanças, se as suas rotas podem ser unidas. Esta junção das rotas deve respeitar a posição de cada cliente na rota e a capacidade dos camiões. Todas as poupanças são verificadas.

Por fim, são calculadas as distâncias de cada rota e a distância total. Os dados obtidos são exportados para um ficheiro .csv para futura análise.

No Apêndice 1 – Fluxograma: Algoritmo de Poupanças de Clarke and Wright encontra-se um fluxograma do algoritmo implementado.

3.2 Algoritmo baseado em recozimento quântico para resolução do CVRP

3.2.1 Introdução

Neste capítulo apresenta-se um algoritmo para resolução do CVRP baseado em recozimento quântico. O algoritmo foi construído com base num algoritmo já existente para o TSP, e que foi publicado por McCaffrey (2022) na *Visual Studio Magazine*.

Primeiramente, é imperativo esclarecer que o ABRQ aqui retratado não utiliza nem necessita de qualquer tipo de hardware quântico. O algoritmo é uma modificação da meta-heurística de recozimento simulado utilizando alguns conceitos de mecânica quântica.

A pedra angular do algoritmo é o tunelamento quântico. Este conceito é um fenómeno da mecânica quântica no qual algumas partículas, ao encontrarem uma barreira que, classicamente, seria intransponível, acabam por atravessá-la. Por outras palavras, enquanto que na física clássica uma partícula é repelida pela barreira caso não tenha energia suficiente para a atravessar, na física quântica existe uma probabilidade finita da partícula transpor a barreira.

Este fenómeno é utilizado no ABRQ como forma de escapar a mínimos locais. A Figura 11 demonstra como o tunelamento pode permitir que o algoritmo evite mínimos locais e se movimente para soluções mais próximas do mínimo global.

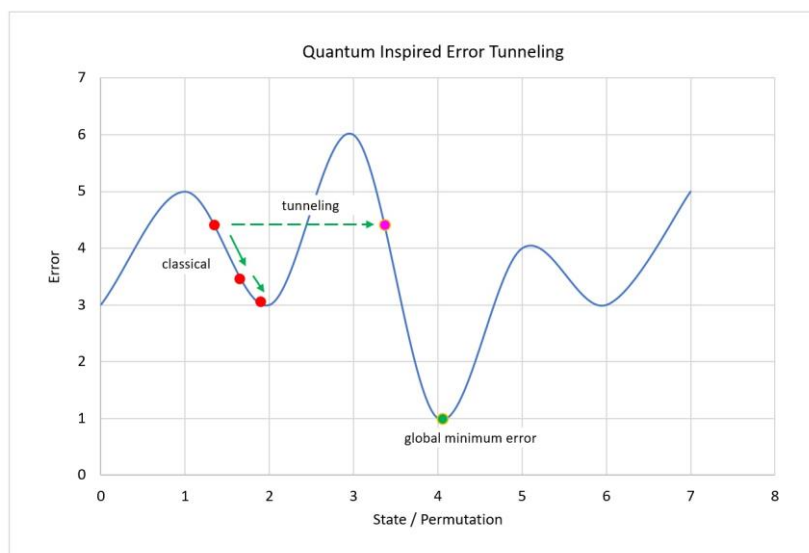


Figura 11 - Tunelamento Quântico (McCaffrey, 2022)

Para um melhor entendimento do algoritmo, encontra-se no Apêndice 2 – Fluxograma: Algoritmo Baseado em Recozimento Quântico um fluxograma do mesmo e no Apêndice 3 – Pseudocódigo: Algoritmo Baseado em Recozimento Quântico o seu pseudocódigo.

3.2.2 Recozimento Quântico

O recozimento quântico é uma meta-heurística usada na resolução de problemas de otimização combinatória baseada nos princípios da mecânica quântica (Kadowaki & Nishimori, 1998).

A computação quântica utiliza princípios da mecânica quântica como a superposição e o entrelaçamento para melhorar os algoritmos em comparação com as aplicações clássicas (Harikrishnakumar et al., 2020). Recentemente, e com os desenvolvimentos na área das tecnologias quânticas, têm surgido uma variedade de hardware especialmente idealizado para resolver problemas de otimização (X. Liu et al., 2019).

O recozimento quântico tem sido utilizado para resolver uma série de problemas de otimização combinatória em áreas como a bioinformática (Perdomo-Ortiz et al., 2012), química quântica (Babbush et al., 2015), biologia computacional (Li et al., 2018), otimização de fluxos de tráfego (Neukart et al., 2017), diagnóstico de falhas (Perdomo-Ortiz et al., 2015), aprimoramento de redes neurais (J. Liu et al., 2018), encaminhamento de veículos (Martonak et al., 2004) e escalonamento em assistência médica (Ikeda et al., 2019).

3.2.3 Configuração inicial

Inicialmente, devem ser introduzidos no código pelo utilizador valores para os parâmetros relativos ao número de camiões, capacidade dos mesmos e o número de iterações. Os parâmetros relacionados com o recozimento quântico e com o recozimento simulado (probabilidade de tunelamento, temperatura, *alpha* e taxa de aceitação), que serão descritas ulteriormente, também devem ser inseridos pelo utilizador.

O ABRQ inicia com a leitura de dois ficheiros de texto (.txt): um contendo as coordenadas dos clientes e outro com a sua respetiva procura. Assim sendo, os ficheiros devem ser preparados previamente.

3.2.4 Cálculo de distâncias

A primeira tarefa do algoritmo é o cálculo da matriz de distâncias, que contém a distância de cada cliente para os restantes e, naturalmente, para o depósito. Como se pretende modelar o CVRP simétrico, ou seja, em que a distância do cliente *i* para o cliente *j* é a mesma que do cliente *j* para o cliente *i*, a matriz a obter será uma matriz simétrica.

As distâncias são calculadas de acordo com a fórmula clássica de cálculo de distâncias no plano cartesiano. Sendo (x_i, y_i) as coordenadas do cliente *i* e (x_j, y_j) as coordenadas do cliente *j*, então a distância d_{ij} entre os dois clientes é dada por:

Equação 2 - Fórmula de cálculo de distâncias no plano cartesiano

$$d_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$$

3.2.5 Solução inicial

Para a geração da solução inicial, o algoritmo divide aleatoriamente os clientes pelas diferentes rotas, tendo em consideração o número de camiões disponíveis. Esta divisão é feita da forma mais equilibrada possível, ou seja, procura-se que as rotas alberguem o mesmo número de clientes ou o número mais similar possível.

De seguida, verifica-se se a procura total de cada rota não excede a capacidade do camião. Se essa condição se verificar, a solução obtida passa a ser a solução inicial. Caso contrário, o algoritmo

tenta outra combinação aleatória de clientes e rotas. O processo descrito encontra-se esquematizado no fluxograma da Figura 12.

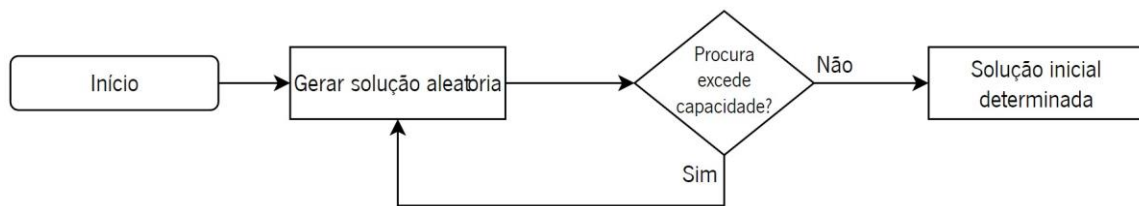


Figura 12 - Fluxograma: Solução inicial

3.2.6 Recozimento Simulado

3.2.6.1 Tunelamento

O algoritmo efetua, a cada iteração, uma troca de um par aleatório de clientes, quer se encontrem na mesma rota ou em rotas diferentes. A solução é aceite caso diminua o valor da função objetivo que é, neste caso, a distância total de todas as rotas.

Contudo, existe a probabilidade de em certas iterações se efetuar um tunelamento, isto é, de ser efetuada mais de uma troca. Este efeito túnel simula o tunelamento quântico e tem como objetivo escapar de mínimos locais. Ao serem efetuadas várias trocas numa só iteração, a solução move-se para longe da solução atual, abrindo a possibilidade de, em iterações seguintes, se obter soluções com melhor qualidade (ver Figura 11). A probabilidade do algoritmo realizar um tunelamento é um dos parâmetros definidos pelo utilizador.

Como se pressupõe que em iterações mais avançadas do ABRQ a solução seja de melhor qualidade, deve-se garantir que o número de trocas a efetuar em cada tunelamento diminua à medida que se processam mais iterações. Para tal, definiu-se a seguinte fórmula para determinar o número de trocas de um tunelamento numa determinada iteração:

Equação 3 – Fórmula de cálculo do número de trocas por tunelamento

$$\text{número de trocas} = \text{percentagem de iterações em falta} \times \text{número de clientes}$$

Com isto, no início do algoritmo a percentagem de iterações em falta será de 100%, resultando num número total de trocas igual ao número total de clientes. Com o decorrer do algoritmo, e com menos iterações em falta, este valor vai diminuindo gradualmente. A Tabela 1 mostra o número de trocas a

efetuar em várias iterações de uma instância executada com 100 iterações e 15 clientes. De referir ainda que o número de trocas é arredondado à unidade mais próxima.

Tabela 1 - Número de trocas por tunelamento (exemplo)

Nº iterações	100	
Nº clientes	15	
Iteração	Percentagem iterações em falta	Nº de trocas
1	100%	15
10	91%	14
27	74%	11
50	51%	8
75	26%	4
91	10%	2
100	1%	0

O fluxograma da Figura 13 esquematiza o processo de tunelamento.

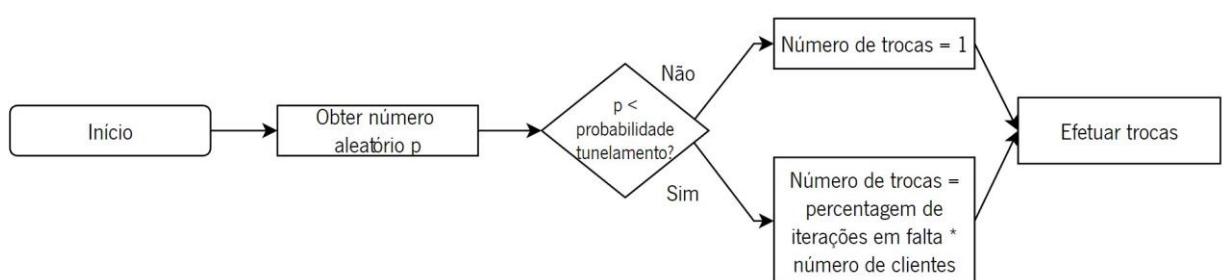


Figura 13 - Fluxograma: Tunelamento

3.2.6.2 Soluções de pior qualidade

A solução obtida através do tunelamento será aceite caso diminua o valor da função objetivo, ou então, e à semelhança do recozimento simulado clássico, poderá ser aceite de acordo com uma certa probabilidade. Assim sendo, garante-se que o algoritmo se desloque, ocasionalmente, para uma solução de pior qualidade na expectativa de alcançar melhores soluções em iterações seguintes.

De forma a obter a probabilidade de aceitação de um túnel que não melhora a qualidade da solução, definem-se inicialmente três parâmetros, sendo eles:

- **Temperatura**, um valor numérico abstrato, e cujo conceito é emprestado do recozimento simulado;
- **Alpha**, a taxa de arrefecimento à qual a temperatura vai diminuindo a cada iteração;

- **Taxa de aceitação**, uma proporção segundo a qual a probabilidade de aceitação do túnel aumenta ou diminui.

Seguindo a mesma lógica do número de trocas, e de acordo com as meta-heurísticas de recozimento simulado, deve-se garantir que em iterações mais avançadas do algoritmo a probabilidade de se aceitar uma solução de pior qualidade seja menor. Por conseguinte, a probabilidade de se aceitar um túnel que não melhore a qualidade da solução é obtida pela seguinte expressão:

Equação 4 - Fórmula de cálculo da probabilidade de aceitação

$$\text{probabilidade aceitação} = (\text{temperatura atual} \div \text{temperatura inicial}) \times \text{taxa de aceitação}$$

Assim sendo, à medida que o algoritmo avança, e como a temperatura vai sendo atualizada à taxa *alpha*, a parcela *temperatura atual / temperatura inicial* torna-se cada vez mais diminuta, conduzindo a uma probabilidade de aceitação menor. Garante-se, desta forma, que em iterações mais avançadas do algoritmo a probabilidade de aceitação de túneis que não melhoram a solução seja muito menor.

A título de exemplo, e considerando uma taxa de aceitação de 0.1, uma temperatura inicial de 100 e um *alpha* de 0.9, verifica-se, tal como é demonstrado na Tabela 2, que na primeira iteração a probabilidade de aceitação da solução é igual à taxa de aceitação (0.1) mas que com o decorrer das iterações a probabilidade vai diminuindo progressivamente, atingindo valores muito próximos de 0.

Tabela 2 - Evolução da probabilidade de aceitação (cenário 1)

Taxa de aceitação	0.1	
Temperatura inicial	1000	
Alpha	0.9	
Iteração	Temperatura atual	Probabilidade de aceitação
1	1000.00	0.100000
10	387.42	0.038742
27	64.61	0.006461
50	5.73	0.000573
75	0.41	0.000041
91	0.08	0.000008
100	0.03	0.000003

Já no cenário descrito na Tabela 3, onde o *alpha* é um valor muito próximo de 1, a probabilidade, na primeira iteração é, igualmente, de 0.1, mas com uma diminuição muito mais lenta que no caso anterior.

Tabela 3 - Evolução da probabilidade de aceitação (cenário 2)

Taxa de aceitação	0.1	
Temperatura inicial	1000	
Alpha	0.9999	
Iteração	Temperatura atual	Probabilidade de aceitação
1	1000.00	0.100000
10	999.10	0.099910
27	997.40	0.099740
50	995.11	0.099511
75	992.63	0.099263
91	991.04	0.099104
100	990.15	0.099015

Na Figura 14 encontra-se um fluxograma que sumariza o processo de aceitação de soluções com pior qualidade.

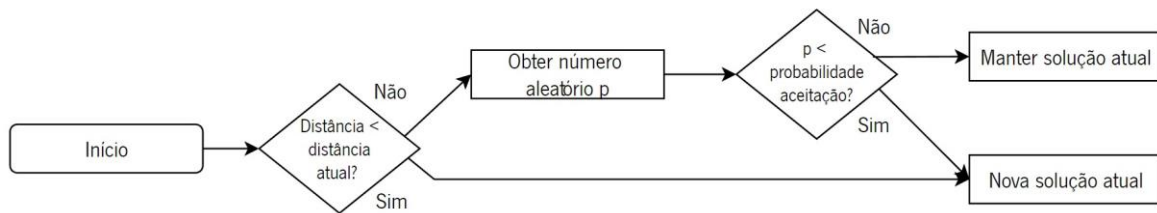


Figura 14 - Processo de aceitação de soluções com pior qualidade

3.2.7 Dados gerados pelo algoritmo

Executadas todas as iterações, o ABRQ devolve ao utilizador diversas informações. Os dados são exportados para um ficheiro .csv para serem posteriormente analisados.

Os dados armazenados são os seguintes:

- **Informações gerais:** instância utilizada, número de clientes;
- **Parâmetros introduzidos pelo utilizador:** número de camiões, capacidade dos camiões, probabilidade de tunelamento, temperatura máxima, *alpha*, taxa de aceitação, número de iterações;

- **Dados gerados pelo algoritmo:** rotas geradas, distância total (função objetivo), número total de túneis, número de túneis aceites com solução de pior qualidade, temperatura final, iteração em que a temperatura atinge zero¹, tempo total de processamento.

¹ Considera-se temperatura zero qualquer temperatura inferior a 0.000005, já que o valor nunca atinge, verdadeiramente, zero.

4. RESULTADOS EXPERIMENTAIS

No presente capítulo encontram-se analisados os resultados obtidos pelos algoritmos implementados. Inicialmente, faz-se uma breve análise às instâncias utilizadas no processamento. Em seguida são analisados e discutidos os resultados obtidos pelo APCW. Por fim são apreciados os resultados obtidos pelo ABRQ.

Todos os algoritmos foram executados num computador com um processador *AMD Ryzen 5 3500U* e um RAM instalada de 8,00 GB. Os resultados estão naturalmente condicionados às especificidades do computador e da linguagem de programação utilizada, o Python 3.

4.1 Instâncias utilizadas

De forma a testar o algoritmo foram selecionadas 40 instâncias, todas elas retiradas do *website CVRPLIB* (Lima et al., 2014), uma biblioteca online de instâncias do CVRP. As instâncias são bastante heterogêneas, divergindo no número de clientes e na capacidade e número de veículos. As instâncias foram selecionadas aleatoriamente, sendo que o único critério de aceitação utilizado foi estar disponível a sua solução ótima. A frota de veículos é homogênea, isto é, todos os veículos de uma dada instância apresentam a mesma capacidade de transporte.

A maior parte dos nomes das instâncias apresentam uma nomenclatura similar. A primeira letra do nome refere-se ao conjunto a que a instância pertence, o segmento seguinte refere-se ao número de nodos da instância (número de clientes + depósito) e o segmento final indica o número de veículos disponíveis. A primeira instância, por exemplo, A-n32-k5, pertence ao grupo de instâncias A, tem 32 nodos e 5 veículos. Na Tabela 4 encontram-se os dados mais relevantes das instâncias utilizadas.

Tabela 4 - Dados das instâncias utilizadas

Instância	Nº Clientes	Nº Veículos	Capacidade Camião	Solução Ótima
A-n32-k5	31	5	100	784.00
A-n37-k6	36	6	100	949.00
A-n44-k6	43	6	100	937.00
A-n53-k7	52	7	100	1010.00
A-n64-k9	63	9	100	1401.00
A-n80-k10	79	10	100	1763.00
B-n31-k5	30	5	100	672.00
B-n44-k7	43	7	100	909.00
B-n52-k7	51	7	100	747.00
B-n66-k9	65	9	100	1316.00
B-n78-k10	77	10	100	1221.00
CMT1	50	5	160	524.61
CMT2	75	10	140	835.26
CMT5	199	17	200	1291.29
CMT12	100	10	200	819.56
CMT13	120	11	200	1541.14
E-n22-k4	21	4	6000	375.00
E-n33-k4	32	4	8000	835.00
E-n51-k5	50	5	160	521.00
E-n76-k10	75	10	140	830.00
E-n101-k14	100	14	112	1067.00
F-n45-k4	44	4	2010	724.00
F-n135-k7	134	7	2210	1162.00
Golden_13	252	26	1000	857.19
Golden_14	320	29	1000	1080.55
Golden_17	240	22	200	707.76
Golden_20	420	38	200	1817.59
M-n101-k10	100	10	200	820.00
M-n151-k12	150	12	200	1015.00
M-n200-k17	199	17	200	1275.00
P-n16-k8	15	8	35	450.00
P-n20-k2	19	2	160	216.00
P-n40-k5	39	5	150	458.00
P-n51-k10	50	10	80	741.00
P-n60-k10	59	10	120	744.00
P-n70-k10	69	10	135	827.00
P-n101-k4	100	4	400	681.00
X-n101-k25	100	25	206	27591.00
X-n148-k46	147	46	18	43448.00
X-n655-k131	654	131	5	106780.00

Tal como é possível verificar, a instância com menor número de clientes é a P-n16-k8, com apenas 15 clientes, contrastando com os 654 da instância X-n655-k131, instância que registra também o maior número de veículos disponíveis: 131. De registrar ainda que a instância P-n20-k2 dispõe de apenas 2 veículos.

Algumas das instâncias selecionadas apresentam um número de clientes extremamente elevado para o problema em questão. No TSP, por exemplo, o número de soluções possíveis de enumerar é calculado como $n!$, o fatorial do número de nodos a visitar da instância. Assim sendo, uma instância com apenas 5 clientes terá 120 soluções possíveis de enumerar. No caso de uma instância com 101 nodos, como é caso da M-n101-k10, então existiriam $9.4259E+159$ soluções possíveis, um número superior ao número de átomos no universo observável. No caso da maior instância selecionada, a X-n655-k131, com 655 nodos, existiria um total irrisório de $9.6035474365E+1561$ soluções possíveis. Sendo o VRP um problema geralmente mais complexo que o TSP, é perceptível a complexidade do problema em questão e a dificuldade computacional de se encontrarem soluções de boa qualidade.

4.2 Soluções obtidas pelo Algoritmo de Poupanças de Clarke and Wright

O primeiro algoritmo testado foi o APCW. O algoritmo conseguiu encontrar uma solução para todas as 40 instâncias. Os dados obtidos encontram-se na Tabela 5.

Tabela 5 - Soluções obtidas pelo APCW

Instância	Nº Clientes	Veículos	Capacidade Veículo	Solução	Tempo Processamento (segundos)	Solução ótima	Desvio % Solução Ótima
A-n32-k5	31	5	100	2615.87	0.13	784.00	233.66%
A-n37-k6	36	6	100	3801.79	0.24	949.00	300.61%
A-n44-k6	43	6	100	2284.72	0.45	937.00	143.83%
A-n53-k7	52	7	100	3769.39	0.84	1010.00	273.21%
A-n64-k9	63	9	100	7646.22	1.81	1401.00	445.77%
A-n80-k10	79	10	100	7795.68	3.97	1763.00	342.18%
B-n31-k5	30	5	100	2275.33	0.12	672.00	238.59%
B-n44-k7	43	7	100	3721.99	0.41	909.00	309.46%
B-n52-k7	51	7	100	3010.86	0.82	747.00	303.06%
B-n66-k9	65	9	100	6551.67	1.88	1316.00	397.85%
B-n78-k10	77	10	100	5557.10	4.27	1221.00	355.13%
CMT1	50	6	160	1769.17	0.69	524.61	237.24%
CMT12	100	10	200	2863.87	8.84	819.56	249.44%
CMT13	120	7	200	4007.36	18.79	1541.14	160.03%
CMT2	75	11	140	5040.30	3.22	835.26	503.44%
CMT5	199	17	200	9014.75	124.99	1291.29	598.12%
E-n101-k14	100	14	112	7319.32	9.94	1067.00	585.97%
E-n22-k4	21	4	6000	1018.55	0.04	375.00	171.61%
E-n33-k4	32	4	8000	2322.84	0.16	835.00	178.18%
E-n51-k5	50	6	160	1769.17	0.71	521.00	239.57%
E-n76-k10	75	2	830	1035.21	3.24	830.00	24.72%
F-n135-k7	134	7	2210	2436.34	27.95	1162.00	109.67%
F-n45-k4	44	4	2010	2549.67	0.48	724.00	252.16%
Golden_13	252	27	1000	17073.96	289.96	857.19	1891.85%
Golden_14	320	31	1000	20856.00	1020.61	1080.55	1830.13%
Golden_17	240	22	200	5688.94	316.88	707.76	703.80%
Golden_20	420	38	200	26009.07	2945.35	1817.59	1330.96%
M-n101-k10	100	10	200	2863.87	9.47	820.00	249.25%
M-n151-k12	150	12	200	6189.94	49.37	1015.00	509.85%
M-n200-k17	199	18	200	8940.85	137.28	1275.00	601.24%
P-n101-k4	100	4	400	1636.61	9.05	681.00	140.32%
P-n16-k8	15	9	35	2321.73	0.03	450.00	415.94%
P-n20-k2	19	2	160	377.13	0.04	216.00	74.60%
P-n40-k5	39	5	150	970.81	0.30	458.00	111.97%
P-n51-k10	50	12	80	4498.27	0.77	741.00	507.05%
P-n60-k10	59	10	120	4978.99	1.27	744.00	569.22%
P-n70-k10	69	11	135	6245.33	2.28	827.00	655.18%
X-n101-k25	100	27	206	403490.36	10.26	27591.00	1362.40%
X-n148-k46	147	49	18	849464.56	50.03	43448.00	1855.13%
X-n655-k131	654	131	5	8042373.51	20417.06	106780.00	7431.72%
Média					636.85		672.35%

O tempo médio de execução do algoritmo foi de cerca de 636 segundos por instância (aproximadamente 10 minutos e meio) e o desvio percentual face à solução ótima foi de cerca de 672%. Contudo, se não incluirmos a instância X-n655-k131 nos cálculos, os valores descem drasticamente, já que esta apresenta valores de desvio e de tempo de processamento imensamente superiores a todas as outras instâncias. Estas diferenças podem ser consultadas na Tabela 6.

Tabela 6 - Diferenças no Desvio Médio e no Tempo de Processamento com e sem a instância X-n655-k131

	Média Desvio	Média Tempo Processamento (segundos)
Com X-n655-k131	672.35%	636.85
Sem X-n655-k131	499.04%	129.66
Diferença	173.31%	507.18

Quando não se inclui a instância X-n655-k131 nos cálculos, a média do desvio percentual face à solução ótima desce uns consideráveis 173%. Contudo, a principal diferença acaba por ser na média do tempo de processamento, que diminui 507 segundos (cerca de 8 minutos e meio). De referir ainda que 16 das instâncias apresentaram um tempo de execução menor que um segundo. A instância com menor desvio foi a E-n76-k10, com um desvio de menos de 25%.

De seguida procurou-se encontrar alguma relação entre o número de clientes e o desvio do tempo de processamento. Na Figura 15 encontra-se um gráfico de dispersão com o número de clientes das instâncias e o seu respetivo desvio. Já o gráfico de dispersão da Figura 16 compara o número de clientes e o tempo de processamento. A cinzento encontra-se, em ambas as figuras, uma linha de tendência linear.

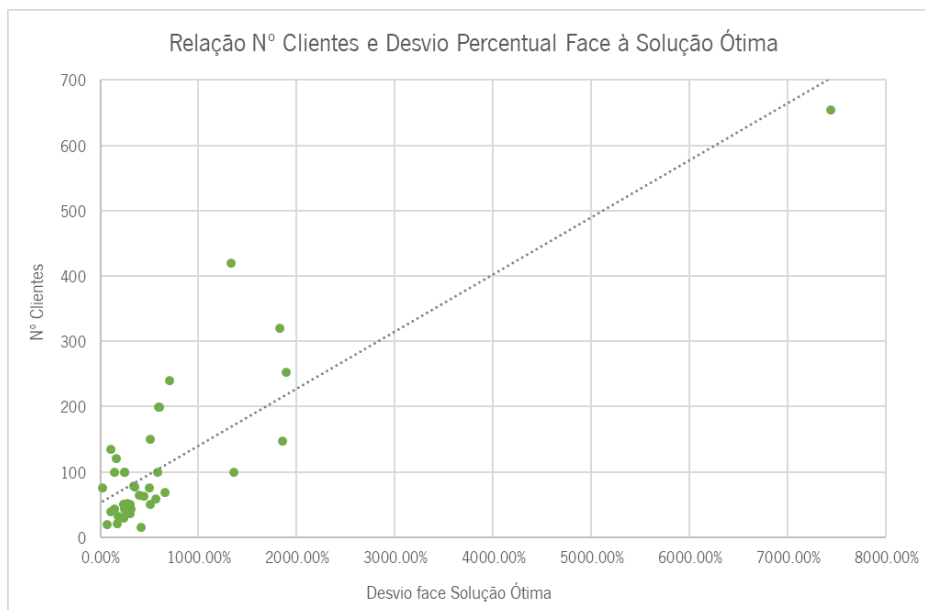


Figura 15 - Gráfico de Dispersão: Relação entre o n° de clientes e o desvio percentual face à solução ótima

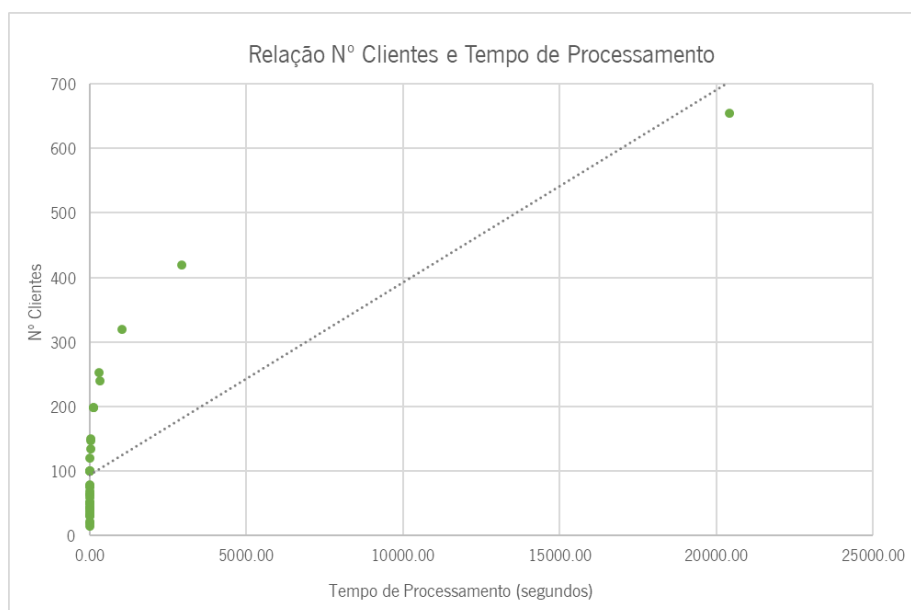


Figura 16 - Gráfico de Dispersão: Relação entre o n° de clientes e o tempo de processamento

Tal como é possível verificar, existe uma tendência para instâncias com maior número de clientes apresentarem piores resultados, isto é, desvios e tempos de processamento com valores mais elevados. Este é um resultado esperado, já que ao trabalhar com um maior número de clientes o algoritmo terá de efetuar mais cálculos. Na elaboração da matriz de distâncias (e, de igual forma, na matriz de poupanças), para a instância A-n32-k5 o APCW terá de efetuar apenas $31^2 = 961$ cálculos. Já na instância X-n655-k131, apenas para a criação de cada uma das matrizes, o algoritmo tem de realizar $654^2 = 427\,716$ cálculos.

4.3 Soluções obtidas pelo Algoritmo Baseado em Recozimento Quântico

Nesta secção analisam-se os resultados obtidos no ABRQ. De forma a aferir a qualidade dos resultados, os mesmos serão comparados com os resultados obtidos no APCW e com a solução ótima.

4.3.1 Limitações Computacionais e de Tempo de Processamento

As características do hardware e do software utilizado no processamento do algoritmo afetam diretamente o seu tempo de processamento. Devido ao limite temporal restrito da presente dissertação, a realização de experiências computacionais com o ABRQ sofreu algumas limitações, que são explicadas de seguida. Algumas das instâncias selecionadas apresentam características dimensionais que tornam o seu processamento ainda mais moroso e impraticável no contexto do ABRQ. Algumas instâncias foram

excluídas do processamento do ABRQ, enquanto outras estarão limitadas a um número mais reduzido de iterações.

Uma das características que afeta diretamente o tempo de processamento, e à semelhança do que acontece no APCW, é o número de clientes da instância. Em casos com muitos clientes, o Python tem de lidar com listas muito extensas, o que dificulta todas as fases do algoritmo, desde a criação da matriz de distâncias às trocas efetuadas na fase do recozimento quântico.

Como o algoritmo apenas considera uma iteração como concluída quando encontra uma solução que respeite todas as restrições a que está sujeita, cada iteração pode, na verdade, realizar uma grande quantidade de cálculos que não gerem nenhum resultado. Instâncias em que seja mais difícil encontrar trocas viáveis terão um tempo de processamento mais demorado. Esta dificuldade em encontrar trocas viáveis torna-se bastante clara em instâncias em que a capacidade total de transporte seja muito próxima da procura total. Nestes casos, a probabilidade de uma troca respeitar a restrição de capacidade é muito reduzida, sendo necessário efetuar uma nova troca, originando, naturalmente, tempos computacionais mais elevados.

Em termos práticos, excluíram-se do estudo do ABRQ as instâncias que não tenham avançado mais do que uma iteração ao final de 30 minutos. As instâncias com um tempo por iteração superior a 0.1 segundos serão processadas com um limite mais reduzido de iterações.

Na Tabela 7 apresentam-se o número de clientes, quantidade total a transportar, procura total e ocupação dos veículos de cada instância. Para além disso, encontram-se indicadas a verde as instâncias sem qualquer limitação de processamento, a vermelho as instâncias que foram descartadas e a amarelo as instâncias que foram processadas com um número reduzido de iterações.

Tabela 7 - Características das instâncias que afetam o processamento

Instância	Nº Clientes	Quantidade total a transportar	Procura total	Ocupação veículos
A-n32-k5	31	500	410	82.00%
A-n37-k6	36	600	570	95.00%
A-n44-k6	43	600	570	95.00%
A-n53-k7	52	700	664	94.86%
A-n64-k9	63	900	848	94.22%
A-n80-k10	79	1000	942	94.20%
B-n31-k5	30	500	412	82.40%
B-n44-k7	43	700	641	91.57%
B-n52-k7	51	700	606	86.57%
B-n66-k9	65	900	861	95.67%
B-n78-k10	77	1000	937	93.70%
CMT1	50	800	777	97.13%
CMT2	75	1400	1364	97.43%
CMT5	199	3400	3186	93.71%
CMT12	100	2000	1810	90.50%
CMT13	120	2200	1375	62.50%
E-n22-k4	21	24000	22500	93.75%
E-n33-k4	32	32000	29370	91.78%
E-n51-k5	50	800	777	97.13%
E-n76-k10	75	1400	1364	97.43%
E-n101-k14	100	1568	1458	92.98%
F-n45-k4	44	8040	7220	89.80%
F-n135-k7	134	15470	14620	94.51%
Golden_13	252	26000	25136	96.68%
Golden_14	320	29000	28672	98.87%
Golden_17	240	4400	4320	98.18%
Golden_20	420	7600	7560	99.47%
M-n101-k10	100	2000	1810	90.50%
M-n151-k12	150	2400	2235	93.13%
M-n200-k17	199	3400	3186	93.71%
P-n16-k8	15	280	246	87.86%
P-n20-k2	19	320	310	96.88%
P-n40-k5	39	750	618	82.40%
P-n51-k10	50	800	777	97.13%
P-n60-k10	59	1200	1134	94.50%
P-n70-k10	69	1350	1313	97.26%
P-n101-k4	100	1600	1458	91.13%
X-n101-k25	100	5150	5147	99.94%
X-n148-k46	147	828	817	98.67%
X-n655-k131	654	655	654	99.85%

Das 40 instâncias selecionadas inicialmente, 14 foram descartadas do processamento, 10 foram processadas com um menor número de iterações, e as restantes 16 não sofreram qualquer tipo de

limitação. É também perceptível que as instâncias excluídas apresentam um número de clientes elevado aliado a uma alta ocupação dos veículos.

Uma análise mais atenta da Tabela 7 revela dois casos contraditórios ao referido até agora: a instância P-n16-k8, com apenas 15 clientes e uma ocupação dos veículos de apenas 87,86% é excluída, e a instância X-n655-k131, que com 654 clientes e uma ocupação de 99,85% não sofrerá qualquer tipo de limitação. Em relação ao primeiro caso, por ser uma instância pequena e com uma ocupação dos veículos baixa, seria de esperar que o algoritmo não tivesse qualquer tipo de dificuldade em processá-la. A exclusão desta instância deriva das características da sua procura e da forma como é obtida a solução inicial do ABRQ. A instância contém dois clientes cuja procura é tão elevada que os mesmos têm de ser servidos sozinhos, ou seja, numa rota dedicada exclusivamente a cada um desses clientes. O cliente 3 apresenta uma procura de 30 e o cliente 7 de 31 unidades. Como a capacidade dos veículos é de 35 unidades e o cliente com menor procura é de 6 unidades, os clientes 3 e 7 não podem de maneira nenhuma ser combinados com outros clientes numa mesma rota. Porém, na criação da solução inicial, o algoritmo divide os clientes em rotas com o mesmo número de clientes ou com o número mais similar possível. Tendo a instância 15 clientes e 8 veículos disponíveis, serão criadas 7 rotas com 2 clientes e apenas uma rota com 1 cliente. Assim sendo, e como os clientes 3 e 7 devem estar em rotas exclusivas para que a restrição de capacidade seja cumprida, o algoritmo nunca conseguirá encontrar uma solução inicial, comprometendo todo o algoritmo.

Em relação à instância X-n655-k131, seria de esperar que com 654 clientes e uma ocupação de veículos de 99.85% o algoritmo tivesse dificuldades em processar a instância. Contudo, a procura de todos os clientes desta instância é de apenas uma unidade. Desta forma, após se encontrar uma solução inicial, todas as trocas a efetuar serão válidas, já que a restrição de capacidade será sempre satisfeita. Estes dois casos demonstram que todas as instâncias são diferentes entre si e que as suas características próprias podem afetar a performance do algoritmo. Apesar do número de clientes e da taxa de ocupação dos veículos da instância serem os fatores que mais afetam a performance do algoritmo, todas as particularidades das instâncias devem ser tidas em conta para um melhor entendimento do funcionamento do processamento.

4.3.2 Primeiro Processamento e Observações Iniciais

No primeiro processamento realizado considerou-se, para todas as instâncias, uma temperatura inicial de 1000 e um *alpha* de 99.99%, o que resulta numa diminuição da temperatura bastante lenta. Por outro lado, utilizou-se uma taxa de aceitação de apenas 1%, o que reduz a probabilidade de soluções de pior qualidade serem aceites. Tendo em conta os valores definidos por estes parâmetros, a probabilidade de aceitação (ver Equação 4) foi, para todas as iterações de uma dada instância, semelhante e reduzida. Para além disso, considerou-se uma probabilidade de tunelamento de 10%. Os resultados obtidos encontram-se na Tabela 8.

Tabela 8 - Resultados do primeiro processamento

Instância	Distância total	Desvio % Solução ótima	Desvio % APCW
A-n32-k5	1483.35	89%	-43%
A-n37-k6	1952.76	192%	-49%
A-n44-k6	1908.25	104%	-16%
A-n53-k7	2589.84	176%	-31%
A-n64-k9	3297.61	135%	-57%
A-n80-k10	4501.99	155%	-42%
B-n31-k5	901.63	34%	-60%
B-n44-k7	2086.68	130%	-44%
B-n52-k7	2089.4	180%	-31%
B-n66-k9	3171.23	141%	-52%
B-n78-k10	3557.29	191%	-36%
CMT1	1309.64	150%	-26%
CMT12	3381.95	313%	18%
CMT13	5843.35	279%	46%
E-n22-k4	600.16	60%	-41%
E-n33-k4	1293.06	55%	-44%
E-n51-k5	1303.74	150%	-26%
F-n45-k4	1849.56	155%	-27%
F-n135-k7	6223.25	483%	155%
M-n101-k10	3499.88	327%	22%
M-n151-k12	4834.72	376%	-22%
P-n20-k2	339.09	57%	-10%
P-n40-k5	987.77	116%	2%
P-n60-k10	1813.57	144%	-64%
P-n101-k4	2957.85	132%	81%
X-n655-k131	237149.76	122%	-97%
Média		171%	-19%

Na tabela encontram-se os valores para o desvio percentual da solução obtida pelo ABRQ face à solução ótima e para o desvio percentual da solução obtida pelo ABRQ face à solução do APCW. Estes valores foram calculados segundo as seguintes expressões:

Equação 5 - Fórmula de cálculo do desvio percentual face à solução ótima

$$\text{Desvio Percentual} = \frac{\text{Solução ABRQ} - \text{Solução Ótima}}{\text{Solução Ótima}}$$

Equação 6 - Fórmula de cálculo do desvio percentual face à solução obtida no APCW

$$\text{Desvio Percentual} = \frac{\text{Solução ABRQ} - \text{Solução APCW}}{\text{Solução APCW}}$$

Tal como é possível verificar, e como seria de esperar, o algoritmo apresenta desvios bastante elevados quando comparados com a solução ótima. A média do desvio é de 171%, ou seja, as soluções obtidas são, em média, 171% superiores aos valores da solução ótima. A instância com menor desvio foi a P-n20-k2, com apenas 57%, contrastando com os 483% atingidos pela instância F-n135-k7, a instância que registou maior desvio face à solução ótima.

Por outro lado, quando comparado com o APCW, o algoritmo apresentou soluções inferiores em 21 das 26 instâncias. A instância X-n655-k131 apresentou um desvio de -97%, isto é, o valor obtido pelo ABRQ é 97% inferior à solução obtida pelo APCW. Por outro lado, a instância F-n135-k7 foi a que apresentou o maior desvio, com 155% face ao APCW. A média dos desvios foi de -19%.

Foram necessárias cerca de 22 horas para o processamento de todas as instâncias, variando desde 16 segundos para a instância A-n32-k5 até aproximadamente 3 horas e meia para a instância A-n80-k10. Estes valores são bastante superiores aos registados pelo APCW, contudo os mesmos não podem ser comparados já que o tempo de processamento do ABRQ depende diretamente do número de iterações. Como não existe nenhuma informação relativa ao tempo despendido para encontrar a solução ótima, também não será possível fazer qualquer tipo de análise comparativa.

De forma a tentar encontrar alguma relação entre o desvio e o os parâmetros a definir, criaram-se três gráficos de dispersão:

- Para aferir possíveis relações do parâmetro **taxa de aceitação**, calculou-se a percentagem de túneis aceites com uma solução de pior qualidade e a mesma foi comparada com o desvio percentual face à solução ótima;
- Para aferir possíveis relações do parâmetro ***alpha*** comparou-se a temperatura final com o desvio percentual face à solução ótima;
- Para aferir possíveis relações do parâmetro **probabilidade de tunelamento**, calculou-se a percentagem de iterações em que foi efetuado um tunelamento e a mesma foi comparada com o desvio percentual face à solução ótima.

É ainda de referir que o valor do parâmetro temperatura não terá nenhuma influência no algoritmo, já que o mesmo é condicionado inteiramente pelo *alpha*. Nesse sentido, utilizou-se o valor de 1000 para o parâmetro temperatura em todos os processamentos deste trabalho.

A Figura 17 apresenta a relação entre a percentagem de túneis aceites com uma solução de pior qualidade e o desvio percentual face à solução ótima.

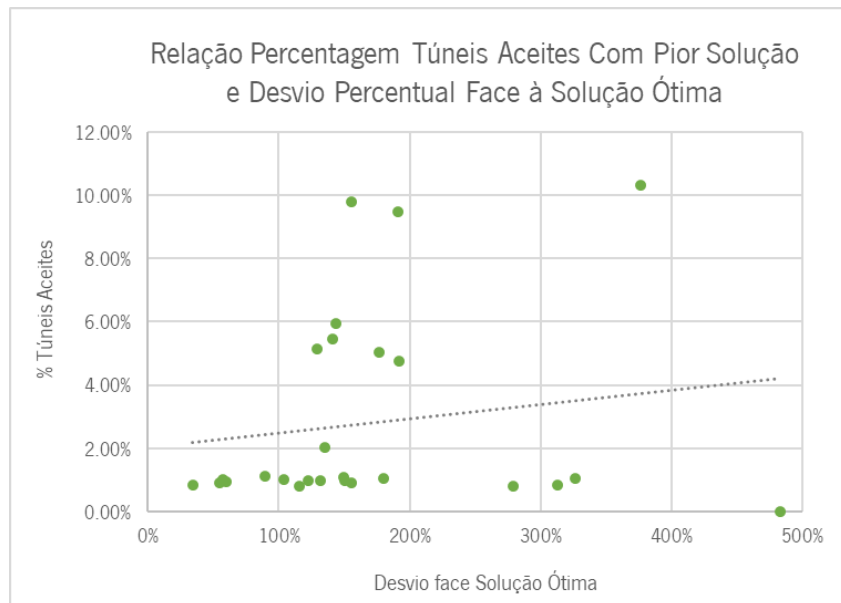


Figura 17 - Gráfico de dispersão: Relação entre a percentagem de túneis aceites com soluções de pior qualidade e o desvio percentual face à solução ótima

A linha de tendência da Figura 17 revela que existe uma relação positiva entre o desvio face à solução ótima e a percentagem de túneis aceites com solução de pior qualidade. Assim sendo, poderá ser benéfico para o algoritmo a diminuição da taxa de aceitação, de forma a reduzir o número de túneis aceites com soluções de pior qualidade. Contudo, é de notar que a taxa de aceitação não é o único parâmetro que afeta a probabilidade de aceitação e por isso alterações que se façam ao parâmetro estão condicionadas ao valor atribuído ao parâmetro *alpha*.

Na Figura 18 encontra-se um gráfico de dispersão que compara a temperatura final com o desvio percentual face à solução ótima.

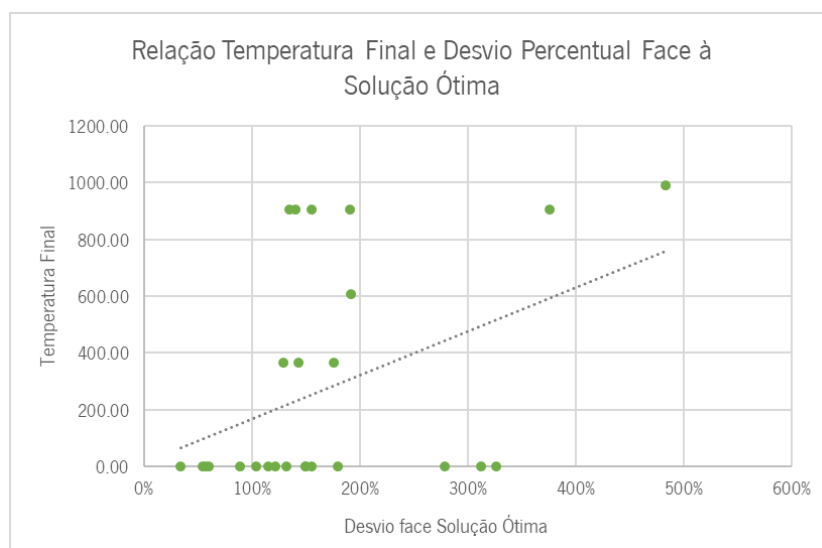


Figura 18 - Gráfico de dispersão: Relação entre a temperatura final e o desvio percentual face à solução ótima

É necessário assinalar que o *alpha* utilizado foi o mesmo para todas as instâncias independentemente do número de iterações a processar, o que leva a diferenças substanciais na temperatura final. Parece existir uma correlação positiva entre as duas variáveis em estudo, o que pode indicar que taxas de alpha menos elevadas gerem soluções com menor desvio, já que irão gerar temperaturas finais mais reduzidas. Contudo, as instâncias com menor número de clientes e com menor ocupação de veículos são as que são processadas com maior número de iterações (já que são as que são processadas mais rapidamente pelo algoritmo) e, em consequência, atingem temperaturas mais baixas. Assim, esta relação entre as duas variáveis pode ser enganosa. Em processamentos seguintes a taxa de *alpha* é ajustada ao número de iterações para que a temperatura final seja semelhante para todas as instâncias.

A Figura 19 ilustra a relação entre a percentagem de iterações em que existiu um tunelamento e o desvio percentual face à solução ótima.

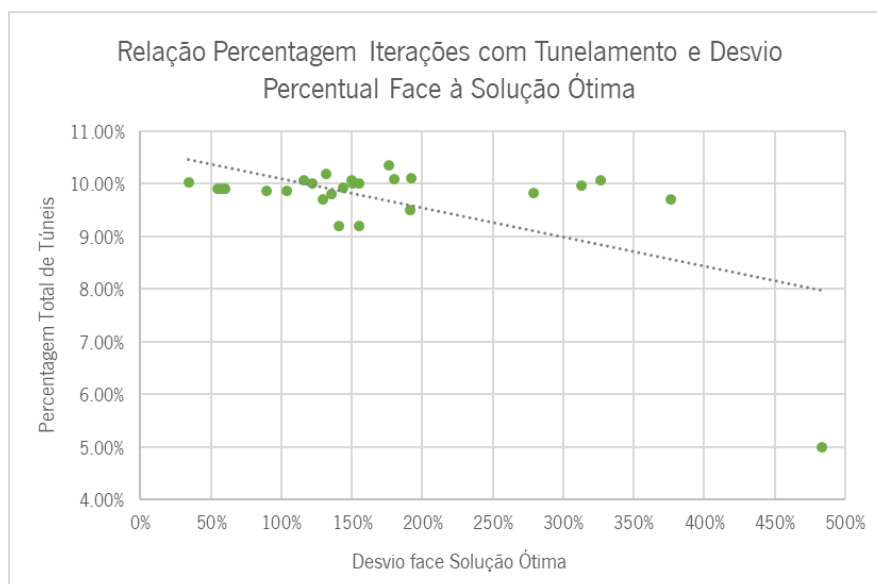


Figura 19 - Gráfico de dispersão: Relação entre a percentagem de iterações em que existiu um tunelamento e o desvio percentual face à solução ótima

Tal como exposto, parece existir uma relação negativa entre a percentagem de iterações com tunelamento e o desvio percentual face à solução ótima, ou seja, quanto maior a percentagem de iterações em que ocorre o tunelamento menor será o desvio. Posto isto, um aumento no parâmetro probabilidade de tunelamento poderá ser vantajoso para o ABRQ. Como a percentagem total de túneis apenas é afetada pelo parâmetro probabilidade de tunelamento, o valor da variável irá sempre convergir para o valor do parâmetro, neste caso 10%, tornando os resultados desta análise pouco pertinentes.

Finda a análise ao primeiro conjunto de experiências computacionais, e tendo em conta todos os aspetos analisados, decidiu-se que num segundo processamento se iria alterar a taxa de aceitação de soluções com pior qualidade de 1% para 10%. Apesar da relação da Figura 17 indicar que uma diminuição da taxa de aceitação pode ser vantajosa para os resultados do algoritmo, decidiu-se comprovar que um aumento na taxa de aceitação não melhora os resultados computacionais do ABRQ. Decidiu-se, também, não ajustar as taxas de *alpha* ao número de iterações no segundo processamento para não ser necessário analisar simultaneamente dois parâmetros que afetam a probabilidade de aceitação. Desta forma, uma melhoria ou uma deterioração dos resultados obtidos serão da inteira responsabilidade da taxa de aceitação.

4.3.3 Segundo Processamento

No segundo processamento a única alteração efetuada aos parâmetros de recozimento quântico foi o aumento da taxa de aceitação de 1% para 10%. Os resultados obtidos encontram-se na Tabela 9, assim como a comparação relativamente ao processamento 1.

Tabela 9 - Resultados do segundo processamento

Instância	Distância total	Desvio % Solução ótima	Diferença Desvio Solução Ótima	Desvio % APCW	Diferença Desvio APCW
A-n32-k5	1564.30	100%	10%	-40%	3%
A-n37-k6	2034.00	204%	12%	-46%	2%
A-n44-k6	1843.93	97%	-7%	-19%	-3%
A-n53-k7	3082.71	229%	53%	-18%	13%
A-n64-k9	3659.82	161%	26%	-52%	5%
A-n80-k10	4633.88	163%	7%	-41%	2%
B-n31-k5	951.59	42%	7%	-58%	2%
B-n44-k7	2091.88	130%	1%	-44%	0%
B-n52-k7	2272.74	204%	25%	-25%	6%
B-n66-k9	3671.57	179%	38%	-44%	8%
B-n78-k10	4237.53	247%	56%	-24%	12%
CMT1	1297.67	147%	-2%	-27%	-1%
CMT12	3527.90	330%	18%	23%	5%
CMT13	5839.23	279%	0%	46%	0%
E-n22-k4	593.24	58%	-2%	-42%	-1%
E-n33-k4	1278.22	53%	-2%	-45%	-1%
E-n51-k5	1333.73	156%	6%	-25%	2%
F-n45-k4	1844.63	155%	-1%	-28%	0%
F-n135-k7	6448.41	504%	21%	165%	9%
M-n101-k10	3562.13	334%	8%	24%	2%
M-n151-k12	4993.91	392%	16%	-19%	3%
P-n20-k2	345.84	60%	3%	-8%	2%
P-n40-k5	1043.31	128%	12%	7%	6%
P-n60-k10	2023.30	172%	28%	-59%	4%
P-n101-k4	2946.58	131%	-1%	80%	-1%
X-n655-k131	238140.49	123%	1%	-97%	0%
Média		184%	13%	-16%	3%

Expectavelmente, a alteração realizada conduziu a um aumento generalizado da função objetivo. A média dos desvios face à solução ótima aumentou para cerca de 184%, contrastando com os 171% prévios. Também o desvio médio face ao APCW aumentou, estando ainda em terrenos negativos, sendo agora de -16%. Seis das instâncias melhoraram o seu resultado, ainda que as melhorias sejam pouco significativas. Para o processamento foram necessárias mais de 22 horas.

Tendo em conta que a alteração efetuada à taxa de aceitação não resultou numa melhoria dos resultados obtidos, em processamentos seguintes utilizou-se a taxa de 1%, correspondente ao primeiro processamento. Decidiu-se ainda, para um terceiro processamento, um aumento da probabilidade de tunelamento para 25% e que se ajustaria o α de cada instância consoante o seu número de iterações.

4.3.4 Terceiro Processamento

No terceiro processamento efetuado aumentou-se a probabilidade de ocorrência de um tunelamento de 10% para 25%. Para além disso, a taxa de *alpha* foi ajustada ao número de iterações de cada instância, de forma que a temperatura final seja semelhante para todas as instâncias. Desta forma, definiu-se o valor de 1 para a temperatura final e calculou-se qual seria a taxa de *alpha* que geraria este valor. Os resultados encontram-se na Tabela 10.

Tabela 10 - Valores de *alpha* ajustados ao número de iterações

Temperatura Inicial	Iterações	Alpha	Temperatura Final
1000	100000	0.999931	1.00755
1000	10000	0.999309	0.99538
1000	5000	0.998619	0.99799
1000	1000	0.993116	0.99995
1000	100	0.933254	0.99997

Tal como é possível observar, o ajustamento das taxas de *alpha* ao número de iterações permite que a temperatura final em todas as instâncias seja muito similar e muito próxima do valor escolhido. Instâncias com maior número de iterações foram então processadas com um *alpha* mais elevado para que a descida da temperatura seja mais lenta. Por outro lado, instâncias com menor número de iterações são processadas com um *alpha* mais elevado para se acelerar o processo de descida da temperatura.

Findo o processamento, obtiveram-se os resultados da Tabela 11. Como base para os cálculos das diferenças dos desvios utilizou-se o processamento 1, já que foi o que obteve melhores resultados até ao momento.

Tabela 11 - Resultados do terceiro processamento

Instância	Distância total	Desvio % Solução ótima	Diferença Desvio Solução Ótima	Desvio % APCW	Diferença Desvio APCW
A-n32-k5	1554.23	98%	9%	-41%	3%
A-n37-k6	1718.84	157%	-35%	-55%	-6%
A-n44-k6	1914.40	104%	1%	-16%	0%
A-n53-k7	2427.39	159%	-17%	-36%	-4%
A-n64-k9	3076.19	120%	-16%	-60%	-3%
A-n80-k10	4252.14	141%	-14%	-45%	-3%
B-n31-k5	966.12	44%	10%	-58%	3%
B-n44-k7	1902.75	109%	-20%	-49%	-5%
B-n52-k7	2147.07	187%	8%	-29%	2%
B-n66-k9	3062.77	133%	-8%	-53%	-2%
B-n78-k10	3566.80	192%	1%	-36%	0%
CMT1	1424.46	172%	22%	-19%	6%
CMT12	3489.38	326%	13%	22%	4%
CMT13	5751.53	273%	-6%	44%	-2%
E-n22-k4	564.59	51%	-9%	-45%	-3%
E-n33-k4	1258.42	51%	-4%	-46%	-1%
E-n51-k5	1335.43	156%	6%	-25%	2%
F-n45-k4	1803.77	149%	-6%	-29%	-2%
F-n135-k7	6060.61	468%	-15%	149%	-7%
M-n101-k10	3424.17	318%	-9%	20%	-3%
M-n151-k12	4758.15	369%	-8%	-23%	-1%
P-n20-k2	339.03	57%	0%	-10%	0%
P-n40-k5	977.73	113%	-2%	1%	-1%
P-n60-k10	1758.02	136%	-7%	-65%	-1%
P-n101-k4	2989.43	134%	2%	83%	2%
X-n655-k131	236871.93	122%	0%	-97%	0%
Média		167%	-4%	-20%	-1%

Este terceiro processamento atingiu uma média do desvio percentual face à solução ótima de apenas 167%, melhor valor registrado até agora e cerca de 4% melhor do que o processamento 1. Ainda assim, e ainda comparativamente ao primeiro processamento, 9 instâncias apresentaram um pior resultado, sendo estes valores compensados pelas grandes descidas no valor da função objetivo de instâncias como a A-n37-k6 e a B-n44-k7, com diminuições a rondar os 35% e os 20% respetivamente. A instância com menor desvio continua a ser a B-n31-k5 com apenas 44% de desvio face à solução ótima.

Face ao APCW atingiu-se agora uma diferença de -20%, uma melhoria de cerca de 1% face ao processamento 1. O ABRQ parece ter bastantes dificuldades em encontrar soluções de qualidade para a instância F-n135-k7 em comparação com o APCW, apresentando um desvio de 149%.

Foram necessárias quase 35 horas para processar todas as instâncias, valor que contrasta com as 22 horas necessárias para os processamentos anteriores. Este aumento substancial do tempo de processamento deriva do aumento da probabilidade de túnel, já que ao serem efetuados mais túneis o algoritmo tem de efetuar todas as tarefas relativas ao tunelamento, nomeadamente trocas de vários pares de clientes e os cálculos necessários para a possível aceitação de soluções com pior qualidade.

Tendo este processamento obtido os melhores resultados até ao momento, será a base de comparação para o processamento seguinte, no qual a probabilidade de tunelamento será aumentada novamente, desta vez para 40%.

4.3.5 Quarto Processamento

No quarto e último processamento decidiu-se aumentar a probabilidade de tunelamento para 40% em detrimento dos 25% usados no processamento anterior. Mantiveram-se, também, as taxas de *alpha* ajustadas ao número de iterações. Os resultados obtidos encontram-se simplificados na Tabela 12, onde foram utilizados os valores do terceiro processamento como base para os cálculos das diferenças percentuais.

Tabela 12 - Resultados do quarto processamento

Instância	Distância total	Desvio % Solução ótima	Diferença Desvio Solução Ótima	Desvio % APCW	Diferença Desvio APCW
A-n32-k5	1549.65	98%	-1%	-41%	0%
A-n37-k6	1709.22	155%	-1%	-55%	0%
A-n44-k6	1947.25	108%	4%	-15%	1%
A-n53-k7	2376.08	154%	-5%	-37%	-1%
A-n64-k9	3135.64	124%	4%	-59%	1%
A-n80-k10	4148.78	135%	-6%	-47%	-1%
B-n31-k5	954.76	42%	-2%	-58%	0%
B-n44-k7	1912.14	110%	1%	-49%	0%
B-n52-k7	2154.88	188%	1%	-28%	0%
B-n66-k9	3158.26	140%	7%	-52%	1%
B-n78-k10	3780.93	210%	18%	-32%	4%
CMT1	1407.17	168%	-3%	-20%	-1%
CMT12	3443.31	320%	-6%	20%	-2%
CMT13	5951.89	286%	13%	49%	5%
E-n22-k4	550.39	47%	-4%	-46%	-1%
E-n33-k4	1373.24	64%	14%	-41%	5%
E-n51-k5	1339.19	157%	1%	-24%	0%
F-n45-k4	1937.58	168%	18%	-24%	5%
F-n135-k7	5969.85	459%	-9%	145%	-4%
M-n101-k10	3499.58	327%	9%	22%	3%
M-n151-k12	4661.02	359%	-10%	-25%	-2%
P-n20-k2	321.91	49%	-8%	-15%	-5%
P-n40-k5	1052.86	130%	16%	8%	8%
P-n60-k10	1712.57	130%	-6%	-66%	-1%
P-n101-k4	2967.48	133%	-2%	81%	-1%
X-n655-k131	237402.94	122%	0%	-97%	0%
Média		169%	2%	-19%	1%

A média dos desvios percentuais face à solução ótima foi neste processamento de 169%, um aumento de cerca de 2% face ao processamento anterior. Ainda assim, metade das instâncias apresentaram uma solução mais reduzida, ainda que essas diminuições sejam pouco significativas. As instâncias com menor e maior desvio continuaram a ser a B-n31-k5 e a M-n151-k12, respetivamente.

O algoritmo continuou a comportar-se melhor que o APCW, principalmente nas instâncias com menor número de clientes. O desvio percentual face ao APCW foi nesta instância de -19%.

Para o processamento foram necessárias mais de 51 horas e meia. Este aumento no tempo de processamento está novamente relacionado com o aumento da probabilidade de tunelamento e todas as tarefas a ele inerentes.

4.3.6 Avaliação da Prestação do Algoritmo

Concluídas as experiências computacionais com o ABRQ, procede-se à sua avaliação em termos de prestação. A melhor combinação de parâmetros obtida foi no terceiro processamento, onde se definiu uma probabilidade de tunelamento de 25%, uma taxa de aceitação de soluções com pior qualidade de 1% e taxas de *alpha* ajustadas ao número de iterações de cada instância e que geram temperaturas finais a rondar 1 unidade. Obteve-se um desvio percentual face à solução ótima de 166.90% e um desvio percentual face ao APCW de -19.93%. Foram necessárias 35 horas para processar todas as instâncias.

É de referir, em primeiro lugar, que o facto do algoritmo apresentar resultados de melhor qualidade para a função objetivo do que o APCW é notável. Contudo, o ABRQ não conseguiu encontrar resultados para uma grande parte das instâncias selecionadas inicialmente. Para além disso, o APCW revela-se muito mais rápido que o ABRQ, o que num contexto da vida real é muito valorizado. O APCW encontrou melhores soluções em várias instâncias, não se tendo encontrado nenhuma relação das características das instâncias e o facto do APCW se comportar melhor que o ABRQ.

Para tentar entender de que forma as características do computador utilizado nas experiências computacionais afetam o tempo de processamento, executaram-se duas instâncias do ABRQ num computador com especificidades de computação bastante superiores. Nesse computador, que possui um processador *Intel Core i9 10900 10-Core* e 32 GB de RAM, os tempos computacionais da instância A-n32-k5 (a instância que foi processada mais rapidamente no quarto processamento) e da instância A-n80-10 (a instância que foi processada mais lentamente no quarto processamento) reduziram para cerca de metade, registando um desvio de -51% e -55% respetivamente. Estas reduções são consideráveis e atenuam um dos principais problemas do ABRQ. Se todos os processamentos fossem efetuados neste computador, provavelmente seria possível processar mais instâncias das 40 inicialmente selecionadas em tempos computacionais razoáveis.

Assim sendo, não é possível definir qual dos dois algoritmos se comportou melhor, já que cada agente de decisão privilegia certas características em detrimento de outras. Caso a instância em causa apresente um elevado número de clientes e uma ocupação de veículos também elevada, recomenda-se a utilização do APCW, pois o ABRQ teria muito mais dificuldades em encontrar uma solução em tempos computacionais razoáveis. Por outro lado, caso se pretenda uma solução mais próxima da otimalidade e o tempo de processamento não seja um problema, então recomenda-se o ABRQ.

Para uma análise mais extensiva dos resultados, encontram-se no Apêndice 4 – Resultados Obtidos pelo Primeiro Processamento, Apêndice 5 – Resultados Obtidos pelo Segundo Processamento,

Apêndice 6 – Resultados Obtidos pelo Terceiro Processamento e Apêndice 7 – Resultados Obtidos pelo Quarto Processamento os resultados completos dos quatro processamentos do ABRQ. No capítulo seguinte enumeram-se as várias limitações do algoritmo, bem como propostas de trabalho futuro que o tornariam mais robusto.

5. CONCLUSÃO

No presente capítulo apresentam-se as conclusões resultantes da realização desta dissertação, destacando-se também as suas principais limitações e sugestões de trabalho futuro.

5.1 Considerações finais

A dificuldade inerente à realização da presente dissertação e à concretização dos seus objetivos foi elevada. Apesar de ser amplamente abordado na literatura e das suas aplicações práticas serem de extrema importância no contexto atual, o VRP é um problema que, para ser corretamente abordado, necessita de um conhecimento profundo por parte do investigador. Para além disso, a meta-heurística que se decidiu construir tem como base princípios de mecânica quântica, área de estudo de extrema complexidade. Foi ainda necessária a aquisição de conhecimento e compreensão da linguagem de programação *Python*.

Considera-se que foram alcançados os principais objetivos propostos para este projeto: obteve-se um conhecimento profundo da problemática em estudo, implementou-se uma heurística construtiva, desenvolveu-se e implementou-se uma meta-heurística baseada em recozimento quântico e analisaram-se e avaliaram-se os resultados obtidos.

A revisão de literatura realizada em primeira instância revelou-se fulcral nos restantes tópicos desta dissertação, já que foi possível entender a complexidade do problema, desde a importância da correta escolha da variante ao entendimento dos diferentes métodos de resolução. Pretendeu-se não só conhecer as aplicações das diferentes variantes e métodos de resolução e as suas abordagens na literatura, mas também perceber o funcionamento dos mesmos de forma a facilitar o processo de implementação e torná-lo o mais adequado possível. Foi também importante a análise atenta ao algoritmo de recozimento quântico apresentado no artigo de McCaffrey (2022) publicado na *Visual Studio Magazine*, dedicado ao TSP, e que serviu de motivação para o desenvolvimento do ABRQ dedicado ao CVRP.

Partiu-se então para a programação dos algoritmos com um conhecimento vasto no tema em estudo e com intenções claras das características dos algoritmos que se pretendia modelar. Sendo inúmeras as implementações do APCW existentes na literatura, estando disponíveis, inclusive, o código das mesmas, poder-se-ia ter utilizado uma delas ao invés da implementação própria. Contudo, a implementação do APCW permitiu que se tivesse um primeiro contacto com a implementação deste tipo

de algoritmos, principalmente no que diz respeito à linguagem de programação utilizada. A implementação deste algoritmo em primeiro lugar foi fundamental para uma implementação mais simples e cuidadosa do ABRQ. A implementação do ABRQ foi ainda mais desafiadora, mas os conhecimentos adquiridos previamente revelaram-se essenciais na construção do algoritmo. Apesar de todas as suas limitações, o algoritmo pôe em prática os conceitos do recozimento quântico pretendidos e foi até capaz de encontrar soluções para o problema que se consideram razoáveis.

Na análise dos resultados computacionais, conseguiu-se analisar de forma simples e clara os resultados obtidos, onde se destacam-se os tempos computacionais de processamento elevados. Obteve-se, no melhor dos processamentos, um desvio percentual face à solução ótima de 166.90% e um desvio percentual face ao Algoritmo de Poupanças de Clarke and Wright de -19.93%, tendo sido necessárias 35 horas de processamento.

Conseguiu-se ultrapassar todos os obstáculos e cumprir com os principais objetivos deste trabalho, fazendo-se um balanço bastante positivo do mesmo. Ainda assim, tem-se consciência das limitações e das possíveis retificações que poderão aumentar a solidez da presente dissertação, e que poderão ser consultadas no tópico seguinte.

5.2 Limitações e Trabalho Futuro

Encontram-se nesta secção explanadas as principais limitações desta dissertação, principalmente no que ao ABRQ diz respeito. Sempre que possível, apresentam-se soluções que podem mitigar os problemas e que poderiam ser implementadas posteriormente para melhoria da qualidade deste projeto.

Em primeiro lugar, indica-se como principal fragilidade desta dissertação o facto do ABRQ não conseguir processar uma parte considerável das instâncias seleccionadas. Apenas 26 das 40 instâncias seleccionadas inicialmente foram analisadas neste estudo devido ao tempo de processamento extremamente elevado. É conhecimento que estes tempos seriam menores se os algoritmos fossem executados num computador com capacidades de processamento superiores, contudo existem algumas modificações que tornariam os algoritmos mais céleres. Logo na fase inicial do algoritmo, aquando da leitura do ficheiro das coordenadas, sugere-se que no futuro se faça ao invés uma leitura de um ficheiro contendo a matriz de distâncias, que deveria ser calculada previamente. Desta forma todo o tempo de processamento inerente à criação da matriz das distâncias seria poupado. Poder-se-ia argumentar que

a criação prévia da matriz de distâncias poderia ser muito trabalhosa e pouco recompensadora, porém, e tendo em conta que se trabalha com um número limitado de instâncias, a criação da matriz teria de ser feita apenas uma vez e a poupança de tempo seria perceptível em todos os processamentos efetuados. Adicionalmente, o mesmo processo poderia ser efetuado no APCW com a matriz de poupanças, apesar da poupança de tempo ser menor já que o algoritmo apenas é processado uma vez.

Sugere-se ainda que se inclua uma lista tabu de soluções não viáveis, isto é, que todas as soluções testadas e que não cumpram com os requisitos do problema sejam armazenadas numa lista. Desta forma, ao efetuar as trocas ou ao gerar a solução inicial (processos aleatórios), essas soluções seriam automaticamente descartadas caso fossem selecionadas novamente, e poupar-se-ia tempo de processamento já que não seria necessário calcular as quantidades a transportar em cada rota nem verificar se as mesmas cumprem a restrição de capacidade.

De forma a reduzir os tempos de processamento poder-se-ia ainda implementar os algoritmos noutra linguagem de programação, nomeadamente C e C++, já que o *Python* prioriza a legibilidade do código sobre a velocidade de processamento.

Outra das grandes limitações do ABRQ é que as trocas efetuadas são apenas trocas simples, isto é, é selecionado um par de clientes e os mesmos trocam de posição quer estejam na mesma rota ou em rotas diferentes. Decorrente disto, o número de clientes em cada rota é sempre o mesmo e é definido aquando da geração da solução inicial. É fácil entender que esta limitação condiciona em grande escala as soluções obtidas pelo algoritmo, pois caso se permitisse outro tipo de trocas (acrescentar ou remover clientes de uma rota, por exemplo) estariam disponíveis soluções que na versão atual do algoritmo nunca serão encontradas. Da mesma forma, o modo como é gerada a solução inicial poderia ser alterada: ao invés de se tentar dividir os clientes de forma equilibrada pelo total das rotas, deveria existir a possibilidade das rotas conterem um número de veículo mais dispar. Esta alteração permitiria não só que fossem encontradas, logo numa fase inicial, soluções de melhor qualidade, mas também diminuiria o tempo de processamento e evitaria casos como a instância P-n16-k8, em que é impossível encontrar solução inicial (ver ponto 4.3.1).

Acredita-se, também, que traria ainda mais valor para esta dissertação uma modificação à forma como são calculados o número de trocas a realizar em cada tunelamento. A fórmula atual, explanada na Equação 3, tem apenas em consideração a percentagem de iterações em falta e o número de clientes. Seria vantajoso que o número de trocas diminuísse a uma taxa definida pelo utilizador, à semelhança do que acontece com a temperatura. Considera-se ainda que o elevado número de trocas que se permite

no início do algoritmo (valor igual ao número total de clientes) seja excessivo e que deve, de alguma forma, ser reduzido.

Recomenda-se que se faça uma análise entre as características das instâncias (número de clientes, número de veículos, capacidade dos veículos, ocupação total dos veículos) e as soluções obtidas de forma a perceber se seria vantajoso a definição de valores diferentes para os parâmetros consoante as suas características.

Por fim, e como é natural, recomenda-se que sejam realizadas mais experiências computacionais com mais combinações dos valores dos parâmetros. Estes processamentos poderiam revelar afinações de melhor qualidade, permitindo melhorar a eficácia e eficiência do algoritmo.

REFERÊNCIAS BIBLIOGRÁFICAS

- Affi, S., Dang, D.-C., & Moukrim, A. (2013). *A Simulated Annealing Algorithm for the Vehicle Routing Problem with Time Windows and Synchronization Constraints* (pp. 259–265). https://doi.org/10.1007/978-3-642-44973-4_27
- Archetti, C., Fernández, E., & Huerta-Muñoz, D. L. (2017). The Flexible Periodic Vehicle Routing Problem. *Computers & Operations Research*, *85*, 58–70. <https://doi.org/10.1016/j.cor.2017.03.008>
- Ardon, L. (2022). *Reinforcement Learning to Solve NP-hard Problems: an Application to the CVRP*.
- Arnold, F., & Sörensen, K. (2019). Knowledge-guided local search for the vehicle routing problem. *Computers & Operations Research*, *105*, 32–46. <https://doi.org/10.1016/j.cor.2019.01.002>
- Arvianto, A., Saptadi, S., Budiawan, W., & Nartadhi, R. L. (2019). *Vehicle routing problem model and simulation with probabilistic demand and sequential insertion*. 020017. <https://doi.org/10.1063/1.5112401>
- Avdoshin, S. M., & Beresneva, E. N. (2019). Local search metaheuristics for Capacitated Vehicle Routing Problem: a comparative study. *Proceedings of the Institute for System Programming of RAS*, *31*(4), 121–138. [https://doi.org/10.15514/ISPRAS-2019-31\(4\)-8](https://doi.org/10.15514/ISPRAS-2019-31(4)-8)
- Babbush, R., Love, P. J., & Aspuru-Guzik, A. (2015). Adiabatic Quantum Simulation of Quantum Chemistry. *Scientific Reports*, *4*(1), 6603. <https://doi.org/10.1038/srep06603>
- Baldacci, R., Battarra, M., & Vigo, D. (2008). Routing a Heterogeneous Fleet of Vehicles. In *The Vehicle Routing Problem: Latest Advances and New Challenges* (pp. 3–27). Springer US. https://doi.org/10.1007/978-0-387-77778-8_1
- Ballou, R. H. (2008). *Gerenciamento da Cadeia de Suprimentos/Logística Empresarial* (5th ed.). Bookman.
- Barbarosoglu, G., & Ozgur, D. (1999). A tabu search algorithm for the vehicle routing problem. *Computers & Operations Research*, *26*(3), 255–270. [https://doi.org/10.1016/S0305-0548\(98\)00047-1](https://doi.org/10.1016/S0305-0548(98)00047-1)
- Beasley, J. (1983). Route first–Cluster second methods for vehicle routing. *Omega*, *11*(4), 403–408. [https://doi.org/10.1016/0305-0483\(83\)90033-6](https://doi.org/10.1016/0305-0483(83)90033-6)
- Campbell, A. M., & Savelsbergh, M. (2004). Efficient Insertion Heuristics for Vehicle Routing and Scheduling Problems. *Transportation Science*, *38*(3), 369–378. <https://doi.org/10.1287/trsc.1030.0046>
- Chen, D.-S., Batson, R. G., & Dang, Y. (2011). Branch-and-Cut Approach. In *Applied Integer Programming* (pp. 305–333). John Wiley & Sons, Inc. <https://doi.org/10.1002/9781118166000.ch12>

- Coene, S., Arnout, A., & Spieksma, F. C. R. (2010). On a periodic vehicle routing problem. *Journal of the Operational Research Society*, *61*(12), 1719–1728. <https://doi.org/10.1057/jors.2009.154>
- Colomi, A., Dorigo, M., & Maniezzo, V. (1991, September). Distributed Optimization by Ant Colonies. *Proceedings of the First European Conference on Artificial Life*.
- Dantzig, G. B., & Ramser, J. H. (1959). The Truck Dispatching Problem. *Management Science*, *6*(1), 80–91. <https://doi.org/10.1287/mnsc.6.1.80>
- Elshaer, R., & Awad, H. (2020). A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants. *Computers & Industrial Engineering*, *140*, 106242. <https://doi.org/10.1016/j.cie.2019.106242>
- Ezzatneshan, A. (2010). A Algorithm for the Vehicle Problem. *International Journal of Advanced Robotic Systems*, *7*(2), 14. <https://doi.org/10.5772/9698>
- Gandomi, A. H., Yang, X.-S., Talatahari, S., & Alavi, A. H. (2013). Metaheuristic Algorithms in Modeling and Optimization. In *Metaheuristic Applications in Structures and Infrastructures* (pp. 1–24). Elsevier. <https://doi.org/10.1016/B978-0-12-398364-0.00001-2>
- Garside, A. K., & Laili, N. R. (2019). A Cluster-First Route-Second Heuristic Approach to Solve The Multi-Trip Periodic Vehicle Routing Problem. *Jurnal Teknik Industri*, *20*(2), 172–181. <https://doi.org/10.22219/JTIUMM.Vol20.No2.172-181>
- Gendreau, M., Iori, M., Laporte, G., & Martello, S. (2008). A Tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints. *Networks*, *51*(1), 4–18. <https://doi.org/10.1002/net.20192>
- Goel, R., & Maini, R. (2017). *Vehicle routing problem and its solution methodologies: a survey*.
- Golden, B., Raghavan, S., & Wasil, E. (Eds.). (2008). *The Vehicle Routing Problem: Latest Advances and New Challenges* (Vol. 43). Springer US. <https://doi.org/10.1007/978-0-387-77778-8>
- Golden, B., & Stewart, W. (1975). Vehicle Routing Problem with Probabilistic Demands. *Computer Science and Statistics: Tenth Annual Symposium on the Interface*.
- Hanafi, R., Rusman, M., Mardin, F., Parenreng, S. M., & Azzazli, A. (2020). Distribution Route Optimization of a Capacitated Vehicle Routing Problem by Sweep Algorithm. *IOP Conference Series: Materials Science and Engineering*, *875*(1), 012066. <https://doi.org/10.1088/1757-899X/875/1/012066>
- Harikrishnakumar, R., Nannapaneni, S., Nguyen, N. H., Steck, J. E., & Behrman, E. C. (2020). *A Quantum Annealing Approach for Dynamic Multi-Depot Capacitated Vehicle Routing Problem*.
- He, H., Daumé, H., & Eisner, J. (2014). Learning to Search in Branch and Bound Algorithms. *NIPS*.

- Hoffman, K. L., & Ralphs, T. K. (2013). Integer and Combinatorial Optimization. In *Encyclopedia of Operations Research and Management Science* (pp. 771–783). Springer US. https://doi.org/10.1007/978-1-4419-1153-7_129
- Ikeda, K., Nakamura, Y., & Humble, T. S. (2019). Application of Quantum Annealing to Nurse Scheduling Problem. *Scientific Reports*, *9*(1), 12837. <https://doi.org/10.1038/s41598-019-49172-3>
- Jawarneh, S., & Abdullah, S. (2015). Sequential Insertion Heuristic with Adaptive Bee Colony Optimisation Algorithm for Vehicle Routing Problem with Time Windows. *PLOS ONE*, *10*(7), e0130224. <https://doi.org/10.1371/journal.pone.0130224>
- Jeřábek, K., Majercak, P., Klietnik, T., & Valaskova, K. (2016). Application of Clark and Wright 's Savings Algorithm Model to Solve Routing Problem in Supply Logistics. *Naše More*, *63*(3), 115–119. <https://doi.org/10.17818/NM/2016/SI7>
- Joubert, J., & Claasen, S. (2006). A sequential insertion heuristic for the initial solution to a constrained vehicle routing problem. *ORION*, *22*(1). <https://doi.org/10.5784/22-1-36>
- Jr., W. D. C., & Rethwisch, D. G. (2018). *Materials Science and Engineering: An Introduction* (10th ed.). Wiley.
- Kadowaki, T., & Nishimori, H. (1998). *Quantum Annealing in the Transverse Ising Model*. <https://doi.org/10.1103/PhysRevE.58.5355>
- Kallehauge, B., Larsen, J., Madsen, O. B. G., & Solomon, M. M. (2006). Vehicle Routing Problem with Time Windows. In *Column Generation* (pp. 67–98). Springer-Verlag. https://doi.org/10.1007/0-387-25486-2_3
- Kianfar, K. (2011). Branch-and-Bound Algorithms. In *Wiley Encyclopedia of Operations Research and Management Science*. John Wiley & Sons, Inc. <https://doi.org/10.1002/9780470400531.eorms0116>
- Kramer, R. H. F. R., Subramanian, A., & Penna, P. H. V. (2015). Problema de roteamento de veículos assimétrico com frota heterogênea limitada: um estudo de caso em uma indústria de bebidas. *Gestão & Produção*, *23*(1), 165–176. <https://doi.org/10.1590/0104-530X1442-14>
- Lapinskaitė, I., & Kuckailytė, J. (2014). The Impact Of Supply Chain Cost On The Price Of The Final Product. *Business, Management and Education*, *12*(1), 109–126. <https://doi.org/10.3846/bme.2014.08>
- Li, R. Y., di Felice, R., Rohs, R., & Lidar, D. A. (2018). Quantum annealing versus classical machine learning applied to a simplified computational biology problem. *Npj Quantum Information*, *4*(1), 14. <https://doi.org/10.1038/s41534-018-0060-8>

- Lima, I., Uchoa, E., Pecin, D., Pessoa, A., Poggi, M., Vidal, T., & Subramanian, A. (2014). *Capacitated Vehicle Routing Problem Library*.
- Lin, S. (1965). Computer Solutions of the Traveling Salesman Problem. *Bell System Technical Journal*, 44(10), 2245–2269. <https://doi.org/10.1002/j.1538-7305.1965.tb04146.x>
- Liu, J., Spedalieri, F., Yao, K.-T., Potok, T., Schuman, C., Young, S., Patton, R., Rose, G., & Chamka, G. (2018). Adiabatic Quantum Computation Applied to Deep Learning Networks. *Entropy*, 20(5), 380. <https://doi.org/10.3390/e20050380>
- Liu, X., Ushijima-Mwesigwa, H., Mandal, A., Upadhyay, S., Safro, I., & Roy, A. (2019). *On Modeling Local Search with Special-Purpose Combinatorial Optimization Hardware*. <https://doi.org/10.13140/RG.2.2.11353.06241>
- Martonak, R., Santoro, G. E., & Tosatti, E. (2004). *Quantum annealing of the Traveling Salesman Problem*. <https://doi.org/10.1103/PhysRevE.70.057701>
- McCaffrey, J. (2022, January 20). *Quantum-Inspired Annealing Using C# or Python*. Visual Studio Magazine.
- Mirshekarian, S., Celikbilek, C., & Süer, G. A. (2015). *Proceedings of 26 th Annual Production and Operations Management Society Conference A Heuristic for the Vehicle Routing Problem with Tight Time Windows and Limited Working Times*.
- Neukart, F., Compostella, G., Seidel, C., von Dollen, D., Yarkoni, S., & Parney, B. (2017). Traffic Flow Optimization Using a Quantum Annealer. *Frontiers in ICT*, 4. <https://doi.org/10.3389/fict.2017.00029>
- Perdomo-Ortiz, A., Dickson, N., Drew-Brook, M., Rose, G., & Aspuru-Guzik, A. (2012). Finding low-energy conformations of lattice protein models by quantum annealing. *Scientific Reports*, 2(1), 571. <https://doi.org/10.1038/srep00571>
- Perdomo-Ortiz, A., Fluegemann, J., Narasimhan, S., Biswas, R., & Smelyanskiy, V. N. (2015). A quantum annealing approach for fault detection and diagnosis of graph-based systems. *The European Physical Journal Special Topics*, 224(1), 131–148. <https://doi.org/10.1140/epjst/e2015-02347-y>
- Pichpibul, T., & Kawtummachai, R. (2013). A Heuristic Approach Based on Clarke-Wright Algorithm for Open Vehicle Routing Problem. *The Scientific World Journal*, 2013, 1–11. <https://doi.org/10.1155/2013/874349>

- Prins, C. (2009). Two memetic algorithms for heterogeneous fleet vehicle routing problems. *Engineering Applications of Artificial Intelligence*, 22(6), 916–928. <https://doi.org/10.1016/j.engappai.2008.10.006>
- Ramos, T. R. P., Gomes, M. I., & Póvoa, A. P. B. (2020). Multi-depot vehicle routing problem: a comparative study of alternative formulations. *International Journal of Logistics Research and Applications*, 23(2), 103–120. <https://doi.org/10.1080/13675567.2019.1630374>
- Rao Tummala, V. M., Phillips, C. L. M., & Johnson, M. (2006). Assessing supply chain management success factors: a case study. *Supply Chain Management: An International Journal*, 11(2), 179–192. <https://doi.org/10.1108/13598540610652573>
- Reinelt, G. (1994). *The Traveling Salesman: Computational Solutions for TSP Applications*. Springer Verlag.
- Renaud, J., Laporte, G., & Boctor, F. F. (1996). A tabu search heuristic for the multi-depot vehicle routing problem. *Computers & Operations Research*, 23(3), 229–235. [https://doi.org/10.1016/0305-0548\(95\)00026-P](https://doi.org/10.1016/0305-0548(95)00026-P)
- Rizzoli, A. E., Montemanni, R., Lucibello, E., & Gambardella, L. M. (2007). Ant colony optimization for real-world vehicle routing problems. *Swarm Intelligence*, 1(2), 135–151. <https://doi.org/10.1007/s11721-007-0005-x>
- Roos, C. (2003). Linear Optimization. In *Encyclopedia of Physical Science and Technology* (pp. 597–616). Elsevier. <https://doi.org/10.1016/B0-12-227410-5/00379-3>
- Sultana, T., Akhand, M. A. H., & Hafizur Rahman, M. M. (2017). A variant fisher and Jaikumar algorithm to solve capacitated vehicle routing problem. *2017 8th International Conference on Information Technology (ICIT)*, 710–716. <https://doi.org/10.1109/ICITECH.2017.8079932>
- Tan, S.-Y., & Yeh, W.-C. (2021). The Vehicle Routing Problem: State-of-the-Art Classification and Review. *Applied Sciences*, 11(21), 10295. <https://doi.org/10.3390/app112110295>
- Tavares, G., Zsigraiova, Z., Semiao, V., & Carvalho, M. G. (2009). Optimisation of MSW collection routes for minimum fuel consumption using 3D GIS modelling. *Waste Management*, 29(3), 1176–1185. <https://doi.org/10.1016/j.wasman.2008.07.013>
- Taylor, B. W. (2013). Module C. Integer Programming: The Branch and Bound Method. In *Introduction to Management Science* (10TH ed.). Pearson.
- Toth, P., & Vigo, D. (Eds.). (2002). *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics.

- Toth, P., & Vigo, D. (Eds.). (2014). *Vehicle Routing: Problems, Methods, and Applications* (2nd ed.). Society for Industrial and Applied Mathematics, Mathematical Optimization Society.
- van Breedam, A. (1995). Improvement heuristics for the Vehicle Routing Problem based on simulated annealing. *European Journal of Operational Research*, *86*(3), 480–490.
[https://doi.org/10.1016/0377-2217\(94\)00064-J](https://doi.org/10.1016/0377-2217(94)00064-J)
- Zulfiqar, L. O. M., Isnanto, R. R., & Nurhayati, O. D. (2018). Using CVRP model in designing decision support system for optimizing distribution route and amounts of utilized vehicles. *2018 International Conference on Information and Communications Technology (ICOIACT)*, 789–792.
<https://doi.org/10.1109/ICOIACT.2018.8350799>

APÊNDICE 1 – FLUXOGRAMA: ALGORITMO DE POUPANÇAS DE CLARKE AND WRIGHT

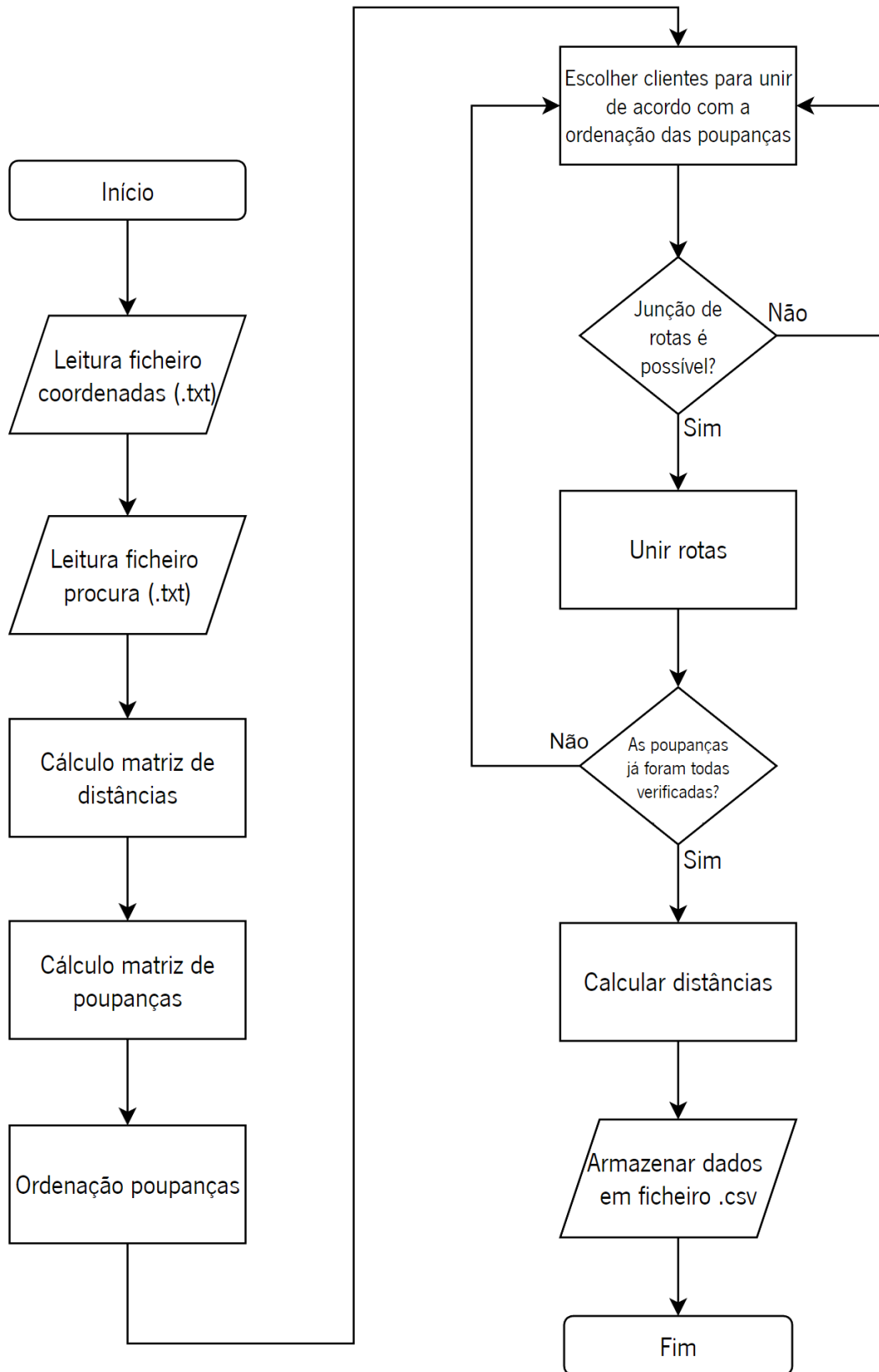


Figura 20 - Fluxograma: APCW

APÊNDICE 2 – FLUXOGRAMA: ALGORITMO BASEADO EM RECOZIMENTO QUÂNTICO

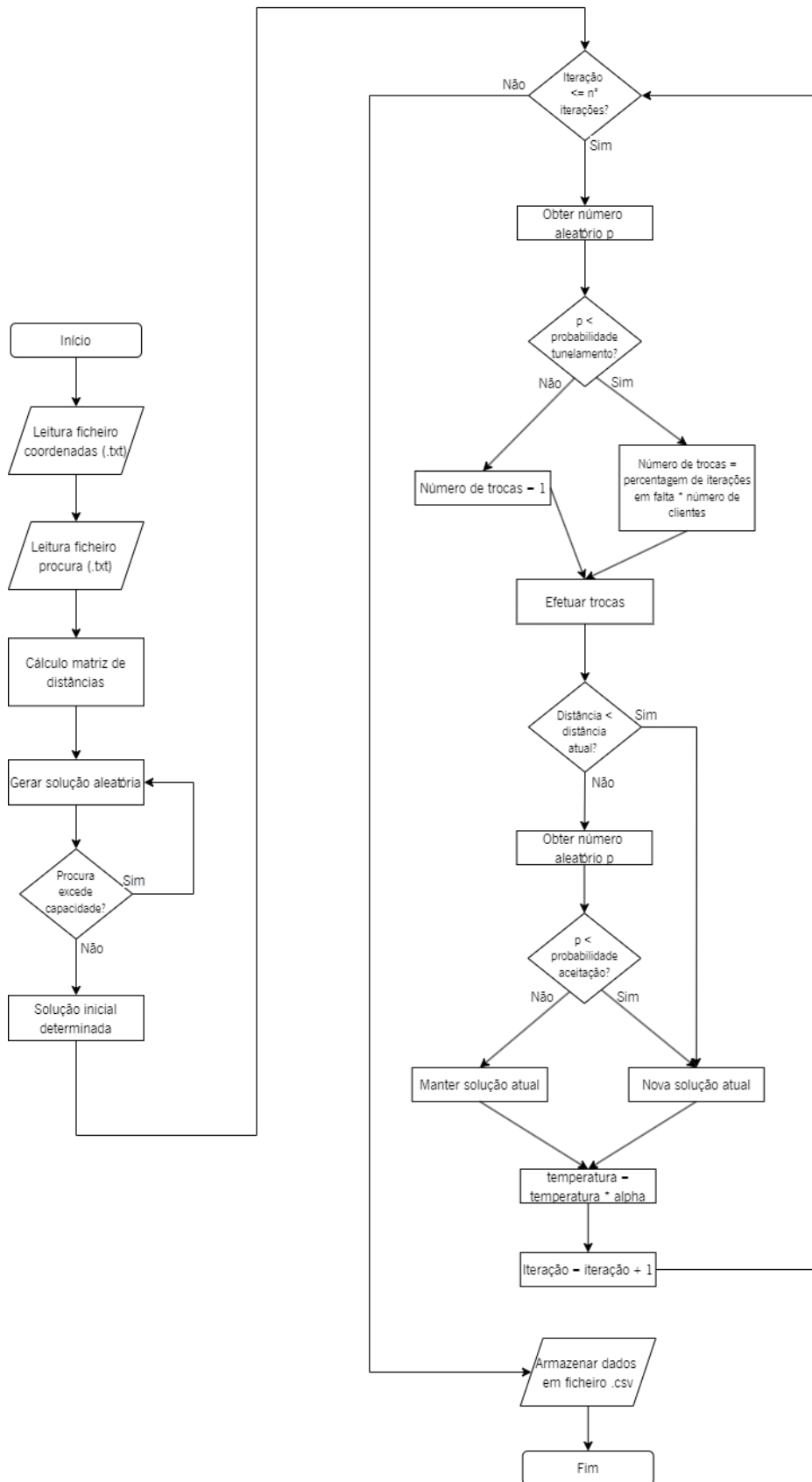


Figura 21 - Fluxograma: ABRQ

APÊNDICE 3 – PSEUDOCÓDIGO: ALGORITMO BASEADO EM RECOZIMENTO QUÂNTICO

//Anteriormente à execução do algoritmo, preparar ficheiro das coordenadas, ficheiro da procura e indicar valores para os parâmetros

Ler ficheiro coordenadas

Ler ficheiro procura

while procura < capacida camião do

 solucao_inicial ← solução aleatória

end while

while iteracao <= numero_iteracoes do

 p ← random in [0.0, 1.0]

 if p < probabilidade_tunelamento then

 numero_trocas ← round(percentagem_iteracoes_em_falta * nclientes)

 tunel ← 1

 else

 numero_trocas ← 1

 end if

 novas_rotas ← rotas com as trocas efetuadas

 distancia ← distancia total das rotas

 if distancia > distancia atual d

 p ← random in [0.0, 1.0]

 if p > probabilidade de aceitação do

 solucao ← solução atual // manter solução atual

 end if

 else

 solucao ← nova solução

 end if

 temperatura ← temperatura * alpha

 iteracao ← iteracao + 1

end while

Armazenar dados em ficheiro .csv

APÊNDICE 4 – RESULTADOS OBTIDOS PELO PRIMEIRO PROCESSAMENTO

Tabela 13 - Resultados completos do primeiro processamento

Instância	Nº Clientes	Nº Veículos	Capacidade Veículos	Probabilidade Tünel	Alpha	Taxa de Aceitação	Nº Iterações	Nº Tüneys	% Iterações com Tünel	Nº Tüneys Aceites (com pior solução)	% Tüneys Aceites	Distância Total	Temperatura Final	Tempo (segundos)	Desvio % Solução Ótima	Desvio % APCW
A-n32-k5	31	5	100	0.1	0.9999	0.01	100000	9867	9.87%	110	1.11%	1483	0.05	16.25	89%	-43%
A-n37-k6	36	6	100	0.1	0.9999	0.01	5000	505	10.10%	24	4.75%	1953	606.52	3080.93	192%	-49%
A-n44-k6	43	6	100	0.1	0.9999	0.01	100000	9862	9.86%	100	1.01%	1908	0.05	2064.49	104%	-16%
A-n53-k7	52	7	100	0.1	0.9999	0.01	10000	1034	10.34%	52	5.03%	2590	367.86	6615.66	176%	-31%
A-n64-k9	63	9	100	0.1	0.9999	0.01	1000	98	9.80%	2	2.04%	3298	904.83	9195.73	135%	-57%
A-n80-k10	79	10	100	0.1	0.9999	0.01	1000	92	9.20%	9	9.78%	4502	904.83	12123.22	155%	-42%
B-n31-k5	30	5	100	0.1	0.9999	0.01	100000	10021	10.02%	83	0.83%	902	0.05	194.50	34%	-60%
B-n44-k7	43	7	100	0.1	0.9999	0.01	10000	970	9.70%	50	5.15%	2087	367.86	3609.18	130%	-44%
B-n52-k7	51	7	100	0.1	0.9999	0.01	100000	10082	10.08%	105	1.04%	2089	0.05	353.08	180%	-31%
B-n66-k9	65	9	100	0.1	0.9999	0.01	1000	92	9.20%	5	5.43%	3171	904.83	5019.71	141%	-52%
B-n78-k10	77	10	100	0.1	0.9999	0.01	1000	95	9.50%	9	9.47%	3557	904.83	4943.69	191%	-36%
CMT1	50	5	160	0.1	0.9999	0.01	100000	10068	10.07%	108	1.07%	1310	0.05	5285.90	150%	-26%
CMT12	100	10	200	0.1	0.9999	0.01	100000	9960	9.96%	83	0.83%	3382	0.05	1895.17	313%	18%
CMT13	120	11	200	0.1	0.9999	0.01	100000	9815	9.82%	78	0.79%	5843	0.05	245.03	279%	46%
E-n22-k4	21	4	6000	0.1	0.9999	0.01	100000	9904	9.90%	93	0.94%	600	0.05	358.03	60%	-41%
E-n33-k4	32	4	8000	0.1	0.9999	0.01	100000	9892	9.89%	91	0.92%	1293	0.05	486.76	55%	-44%
E-n51-k5	50	5	160	0.1	0.9999	0.01	100000	10011	10.01%	97	0.97%	1304	0.05	5055.69	150%	-26%
F-n45-k4	44	4	2010	0.1	0.9999	0.01	100000	9992	9.99%	90	0.90%	1850	0.05	952.67	155%	-27%
F-n135-k7	134	7	2210	0.1	0.9999	0.01	100	5	5.00%	0	0.00%	6223	990.05	1843.19	483%	155%
M-n101-k10	100	10	200	0.1	0.9999	0.01	100000	10068	10.07%	106	1.05%	3500	0.05	1883.89	327%	22%
M-n151-k12	150	12	200	0.1	0.9999	0.01	1000	97	9.70%	10	10.31%	4835	904.83	3607.59	376%	-22%
P-n20-k2	19	2	160	0.1	0.9999	0.01	100000	9902	9.90%	101	1.02%	339	0.05	205.50	57%	-10%
P-n40-k5	39	5	140	0.1	0.9999	0.01	100000	10063	10.06%	81	0.80%	988	0.05	241.94	116%	2%
P-n60-k10	59	10	120	0.1	0.9999	0.01	10000	993	9.93%	59	5.94%	1814	367.86	7232.87	144%	-64%
P-n101-k4	100	4	400	0.1	0.9999	0.01	100000	10184	10.18%	99	0.97%	2958	0.05	295.51	132%	81%
X-n655-k131	654	131	5	0.1	0.9999	0.01	100000	9999	10.00%	96	0.96%	237150	0.05	1649.67	122%	-97%
Media									9.70%	66.96	2.81%			3017.53	171.02%	-19.04%

APÊNDICE 5 – RESULTADOS OBTIDOS PELO SEGUNDO PROCESSAMENTO

Tabela 14 - Resultados completos do segundo processamento

Instância	Nº Clientes	Nº Veículos	Capacidade Veículos	Probabilidade Tunnel	Alpha	Taxa de Aceitação	Nº Iterações	Nº Tunes	% Iterações com Tunnel	Nº Tunes Aceites (com pior solução)	% Tunes Aceites	Distância Total	Temperatura Final	Tempo (segundos)	Desvio % Solução Ótima	Desvio % APCW
A-n32-k5	31	5	100	0.1	0.9999	0.1	100000	9992	9.99%	815	8.16%	1564.3	0.0453772	193.9485	100%	-40%
A-n37-k6	36	6	100	0.1	0.9999	0.1	5000	544	10.88%	333	61.21%	2034	606.5155	3138.646	204%	-46%
A-n44-k6	43	6	100	0.1	0.9999	0.1	100000	9962	9.96%	836	8.39%	1843.9	0.0453772	2464.488	97%	-19%
A-n53-k7	52	7	100	0.1	0.9999	0.1	10000	999	9.99%	518	51.85%	3082.7	367.86105	5210.103	229%	-18%
A-n64-k9	63	9	100	0.1	0.9999	0.1	1000	124	12.40%	90	72.58%	3659.8	904.83289	11475.5	161%	-52%
A-n80-k10	79	10	100	0.1	0.9999	0.1	1000	98	9.80%	69	70.41%	4633.9	904.83289	9775.726	163%	-41%
B-n31-k5	30	5	100	0.1	0.9999	0.1	100000	10121	10.12%	799	7.89%	951.59	0.0453772	192.9804	42%	-58%
B-n44-k7	43	7	100	0.1	0.9999	0.1	10000	993	9.93%	512	51.56%	2091.9	367.86105	3658.517	130%	-44%
B-n52-k7	51	7	100	0.1	0.9999	0.1	100000	10040	10.04%	840	8.37%	2272.7	0.0453772	345.8443	204%	-25%
B-n66-k9	65	9	100	0.1	0.9999	0.1	1000	107	10.70%	81	75.70%	3671.6	904.83289	5274.924	179%	-44%
B-n78-k10	77	10	100	0.1	0.9999	0.1	1000	100	10.00%	75	75.00%	4237.5	904.83289	5563.802	247%	-24%
CMT1	50	5	160	0.1	0.9999	0.1	100000	9936	9.94%	853	8.58%	1297.7	0.0453772	5374.923	147%	-21%
CMT12	100	10	200	0.1	0.9999	0.1	100000	10120	10.12%	816	8.06%	3527.9	0.0453772	2058.975	330%	23%
CMT13	120	11	200	0.1	0.9999	0.1	100000	9906	9.91%	696	7.03%	5839.2	0.0453772	259.4874	279%	46%
E-n22-k4	21	4	6000	0.1	0.9999	0.1	100000	9889	9.89%	880	8.90%	593.24	0.0453772	368.0475	58%	-42%
E-n33-k4	32	4	8000	0.1	0.9999	0.1	100000	9938	9.94%	897	9.03%	1278.2	0.0453772	474.9403	53%	-45%
E-n51-k5	50	5	160	0.1	0.9999	0.1	100000	9981	9.98%	879	8.81%	1333.7	0.0453772	5380.753	156%	-25%
F-n45-k4	44	4	2010	0.1	0.9999	0.1	100000	9975	9.98%	871	8.73%	1844.6	0.0453772	705.6066	155%	-28%
F-n135-k7	134	7	2210	0.1	0.9999	0.1	100	10	10.00%	7	70.00%	6448.4	990.04934	3596.677	504%	165%
M-n101-k10	100	10	200	0.1	0.9999	0.1	100000	9934	9.93%	840	8.46%	3562.1	0.0453772	2027.453	334%	24%
M-n151-k12	150	12	200	0.1	0.9999	0.1	1000	105	10.50%	68	64.76%	4993.9	904.83289	3866.519	392%	-19%
P-n20-k2	19	2	160	0.1	0.9999	0.1	100000	9939	9.94%	839	8.44%	345.84	0.0453772	212.7009	60%	-8%
P-n40-k5	39	5	140	0.1	0.9999	0.1	100000	10095	10.10%	843	8.35%	1043.3	0.0453772	247.1997	128%	7%
P-n60-k10	59	10	120	0.1	0.9999	0.1	10000	1047	10.47%	528	50.43%	2023.3	367.86105	6319.605	172%	-59%
P-n101-k4	100	4	400	0.1	0.9999	0.1	100000	10061	10.06%	750	7.45%	2946.6	0.0453772	366.4127	131%	80%
X-n655-k131	654	131	5	0.1	0.9999	0.01	100000	10045	10.05%	94	0.94%	2381.40	0.0453772	1945.65	123%	-97%
Media									10.18%	570.35	29.58%			3096.13	183.81%	-15.97%

APÊNDICE 6 – RESULTADOS OBTIDOS PELO TERCEIRO PROCESSAMENTO

Tabela 15 - Resultados completos do terceiro processamento

Instância	Nº Clientes	Nº Veículos	Capacidade Veículos	Probabilidade Túnel	Alpha	Taxa de Aceitação	Nº Iterações	Nº Túneis	% Iterações com Túnel	Nº Túneis Aceites (com pior solução)	% Túneis Aceites	Distância Total	Temperatura Final	Tempo (segundos)	Desvio % Solução Última	Desvio % APCW
A-n32-k5	31	5	100	0.25	0.9999	0.01	100000	25068	25.07%	171	0.68%	1554.2	1.0075455	161.972	98%	-41%
A-n37-k6	36	6	100	0.25	0.9986	0.01	5000	1227	24.54%	9	0.73%	1718.8	0.997985	5344.329	157%	-55%
A-n44-k6	43	6	100	0.25	0.9999	0.01	100000	25255	25.26%	143	0.57%	1914.4	1.0075455	3632.732	104%	-16%
A-n53-k7	52	7	100	0.25	0.9993	0.01	10000	2462	24.62%	12	0.49%	2427.4	0.9953775	13452.34	159%	-36%
A-n64-k9	63	9	100	0.25	0.9931	0.01	1000	253	25.30%	2	0.79%	3076.2	0.9999512	12736.06	120%	-6%
A-n80-k10	79	10	100	0.25	0.9931	0.01	1000	254	25.40%	3	1.18%	4252.1	0.9999512	15442.33	141%	-45%
B-n31-k5	30	5	100	0.25	0.9999	0.01	100000	24916	24.92%	137	0.55%	966.12	1.0075455	136.6622	44%	-58%
B-n44-k7	43	7	100	0.25	0.9993	0.01	10000	2498	24.98%	12	0.48%	1902.8	0.9953775	6226.935	109%	-49%
B-n52-k7	51	7	100	0.25	0.9999	0.01	100000	24864	24.86%	158	0.64%	2147.1	1.0075455	361.3775	187%	-29%
B-n66-k9	65	9	100	0.25	0.9931	0.01	1000	272	27.20%	1	0.37%	3062.8	0.9999512	8097.602	133%	-53%
B-n78-k10	77	10	100	0.25	0.9931	0.01	1000	244	24.40%	1	0.41%	3566.8	0.9999512	11635.71	192%	-36%
CMT1	50	5	160	0.25	0.9999	0.01	100000	25096	25.10%	168	0.67%	1424.5	1.0075455	6803.03	172%	-19%
CMT12	100	10	200	0.25	0.9999	0.01	100000	25150	25.15%	137	0.54%	3489.4	1.0075455	2855.096	326%	22%
CMT13	120	11	200	0.25	0.9999	0.01	100000	24814	24.81%	121	0.49%	5751.5	1.0075455	161.0694	273%	44%
E-n22-k4	21	4	6000	0.25	0.9999	0.01	100000	25042	25.04%	132	0.53%	564.59	1.0075455	253.6679	51%	-45%
E-n33-k4	32	4	8000	0.25	0.9999	0.01	100000	25131	25.13%	162	0.64%	1258.4	1.0075455	439.6339	51%	-46%
E-n51-k5	50	5	160	0.25	0.9999	0.01	100000	25094	25.09%	129	0.51%	1335.4	1.0075455	6622.547	156%	-25%
F-n45-k4	44	4	2010	0.25	0.9999	0.01	100000	24862	24.86%	120	0.48%	1803.8	1.0075455	802.5194	149%	-25%
F-n135-k7	134	7	2210	0.25	0.9333	0.01	100	19	19.00%	0	0.00%	6060.6	0.9999678	6462.209	468%	149%
M-n101-k10	100	10	200	0.25	0.9999	0.01	100000	25040	25.04%	130	0.52%	3424.2	1.0075455	2873.461	318%	20%
M-n151-k12	150	12	200	0.25	0.9931	0.01	1000	267	26.70%	3	1.12%	4758.2	0.9999512	6005.219	369%	-23%
P-n20-k2	19	2	160	0.25	0.9999	0.01	100000	24907	24.91%	144	0.58%	339.03	1.0075455	146.6059	57%	-10%
P-n40-k5	39	5	140	0.25	0.9999	0.01	100000	25136	25.14%	125	0.50%	977.73	1.0075455	215.5871	113%	1%
P-n60-k10	59	10	120	0.25	0.9993	0.01	10000	2500	25.00%	15	0.60%	1758	0.9953775	11750.86	136%	-65%
P-n101-k4	100	4	400	0.25	0.9999	0.01	100000	25225	25.23%	118	0.47%	2989.4	1.0075455	284.8151	134%	83%
X-n655-k131	654	131	5	0.25	0.9999	0.01	100000	25026	25.03%	115	0.46%	236872	1.0075455	2604.407	122%	-97%
Medida									24.91%	87.23	0.58%			4827.26	166.90%	-19.93%

APÊNDICE 7 – RESULTADOS OBTIDOS PELO QUARTO PROCESSAMENTO

Tabela 16 - Resultados completos do quarto processamento

Instância	Nº Clientes	Nº Veículos	Capacidade Veículos	Probabilidade Túnel	Alpha	Taxa de Aceitação	Nº Iterações	Nº Túneis	% Iterações com Túnel	Nº Túneis Aceites (com pior solução)	% Túneis Aceites	Distância Total	Temperatura Final	Tempo (segundos)	Desvio % Solução Ótima	Desvio % APCW
A-n32-k5	31	5	100	0.4	0.9999	0.01	100000	39998	0.39998	158	0.40%	1549.7	1.0075455	130.8313	97.66%	-40.76%
A-n37-k6	36	6	100	0.4	0.9986	0.01	5000	2029	0.4058	5	0.25%	1709.2	0.997985	8278.239	155.49%	-55.04%
A-n44-k6	43	6	100	0.4	0.9999	0.01	100000	39716	0.39716	142	0.36%	1947.3	1.0075455	5397.837	107.82%	-14.77%
A-n53-k7	52	7	100	0.4	0.9993	0.01	10000	4034	0.4034	15	0.37%	2376.1	0.9953775	11947.42	153.58%	-36.96%
A-n64-k9	63	9	100	0.4	0.9931	0.01	1000	403	0.403	3	0.74%	3135.6	0.9999512	21236.41	123.81%	-58.99%
A-n80-k10	79	10	100	0.4	0.9931	0.01	1000	389	0.389	2	0.51%	4148.8	0.9999512	24328.5	135.33%	-46.78%
B-n31-k5	30	5	100	0.4	0.9999	0.01	100000	40052	0.40052	128	0.32%	954.76	1.0075455	146.2449	42.08%	-58.04%
B-n44-k7	43	7	100	0.4	0.9993	0.01	10000	4028	0.4028	15	0.37%	1912.1	0.9953775	9958.542	110.36%	-48.63%
B-n52-k7	51	7	100	0.4	0.9999	0.01	100000	39922	0.39922	154	0.39%	2154.9	1.0075455	480.3254	188.47%	-28.43%
B-n66-k9	65	9	100	0.4	0.9931	0.01	1000	392	0.392	1	0.26%	3158.3	0.9999512	12076.76	139.99%	-51.79%
B-n78-k10	77	10	100	0.4	0.9931	0.01	1000	416	0.416	1	0.24%	3780.9	0.9999512	19195.23	209.66%	-31.96%
CMT1	50	5	160	0.4	0.9999	0.01	100000	39670	0.3967	141	0.36%	1407.2	1.0075455	11803.2	168.23%	-20.46%
CMT12	100	10	200	0.4	0.9999	0.01	100000	39924	0.39924	155	0.39%	3443.3	1.0075455	4786.195	320.14%	20.23%
CMT13	120	11	200	0.4	0.9999	0.01	100000	39677	0.39677	130	0.33%	5951.9	1.0075455	286.1631	286.20%	48.52%
E-n22-k4	21	4	6000	0.4	0.9999	0.01	100000	40109	0.40109	134	0.33%	550.39	1.0075455	430.6391	46.77%	-45.96%
E-n33-k4	32	4	8000	0.4	0.9999	0.01	100000	40029	0.40029	156	0.39%	1373.2	1.0075455	693.8668	64.46%	-40.88%
E-n51-k5	50	5	160	0.4	0.9999	0.01	100000	39816	0.39816	124	0.31%	1339.2	1.0075455	10036.28	157.04%	-24.30%
F-n45-k4	44	4	2010	0.4	0.9999	0.01	100000	39907	0.39907	138	0.35%	1937.6	1.0075455	1207.549	167.62%	-24.01%
F-n135-k7	134	7	2210	0.4	0.9333	0.01	100	46	0.46	0	0.00%	5969.9	0.9999678	8445.424	459.50%	145.03%
M-n101-k10	100	10	200	0.4	0.9999	0.01	100000	40050	0.4005	150	0.37%	3499.6	1.0075455	4162.6	326.78%	22.20%
M-n151-k12	150	12	200	0.4	0.9931	0.01	1000	399	0.399	0	0.00%	4661	0.9999512	8759.727	359.21%	-24.70%
P-n20-k2	19	2	160	0.4	0.9999	0.01	100000	40102	0.40102	134	0.33%	321.91	1.0075455	146.869	49.03%	-14.64%
P-n40-k5	39	5	140	0.4	0.9999	0.01	100000	39960	0.3996	146	0.37%	1052.9	1.0075455	267.4379	129.88%	8.45%
P-n60-k10	59	10	120	0.4	0.9993	0.01	10000	3989	0.3989	14	0.35%	1712.6	0.9953775	17200.36	130.18%	-65.60%
P-n101-k4	100	4	400	0.4	0.9999	0.01	100000	40238	0.40238	126	0.31%	2967.5	1.0075455	340.0454	132.74%	81.32%
X-n655-k131	654	131	5	0.4	0.9999	0.01	100000	40060	0.4006	136	0.34%	237403	1.0075455	3917.689	122.33%	-97.05%
Média									40.24%	88.77	0.34%			7140.78	168.63%	-19.39%