

MESTRADO

MÉTODOS QUANTITATIVOS PARA A DECISÃO ECONÓMICA E EMPRESARIAL

TRABALHO FINAL DE MESTRADO

TRABALHO DE PROJETO ELABORADO PARA A OBTENÇÃO
DO GRAU DE MESTRE

OTIMIZAÇÃO DE ROTAS DA REDE DE
TRANSPORTES PÚBLICOS DO CONCELHO DE
BRAGANÇA COM IMPLEMENTAÇÃO EM VBA

MARIANA ISABEL AGROCHÃO COELHO

Orientação:

PROFESSOR DOUTOR JOSÉ PEDRO ROMANA GAIVÃO

OUTUBRO - 2021

Resumo

Este Trabalho Final de Mestrado debruça-se sobre a problemática da baixa procura da rede de transportes públicos do concelho de Bragança, com o intuito específico de otimizar as suas rotas.

Com o envelhecimento da população do interior e o isolamento das suas aldeias, seria de esperar que uma rede de transportes públicos com uma boa cobertura, preços acessíveis e vários passes sociais se revelasse atrativa para a população do concelho de Bragança, em especial para reformados e estudantes. Assim, foram analisados artigos e estudos anteriores, tanto sobre a problemática dos transportes públicos em zonas rurais, como para o caso específico do concelho de Bragança, de forma a perceber as características da rede atual e as necessidades da população.

Para proceder à tentativa de otimização da rede foram estudadas algumas opções de algoritmos heurísticos e meta-heurísticos. Dadas as características dos diferentes algoritmos considerados e o esforço computacional que cada um exige, foi selecionada a meta-heurística *Tabu Search*, com inicialização por um algoritmo de tipo *cluster-first route-second*. Estes foram implementados em VBA, tendo por base um conjunto de paragens pré-estabelecido e a matriz das distâncias respetivas.

A título de comparação, e para avaliar a qualidade das soluções obtidas pelos métodos selecionados, foram ainda efetuados testes com auxílio da ferramenta *Microsoft Excel VRP Spreadsheet Solver*.

Palavras-chave: Otimização de redes; Transportes públicos; *Tabu search*; *Cluster-first route-second*; *VRP Solver*.

Abstract

This master's final project focuses on the problem of low demand of the public transport network in Bragança's county, with the specific purpose of optimizing its routes.

With the ageing population in the interior and the isolation of their small villages, it would be expected that a public transport network with good coverage, accessible prices and several social passes would reveal itself attractive to the population of Bragança, in particular to retired people and students. Therefore, previous articles and studies were analysed, not only about the problematic of public transport in rural areas, but also for the specific case of Bragança, in order to understand the characteristics of the current network and the needs of the population.

To proceed with the attempt to optimize the network, some options of heuristic and meta-heuristic algorithms were studied. Given the characteristics of the different algorithms taken into account and the computational effort required by each, the Tabu search meta-heuristic was chosen, initiated by a cluster-first-route-second algorithm. These were implemented in VBA, based on a set of pre-established stops and the matrix of distances between each pair of them.

As a way of comparison, and to evaluate the quality of the solutions obtained by the selected methods, further tests were performed using the Microsoft Excel VRP Spreadsheet Solver tool.

Keywords: Route optimization; Public transports; *Tabu search*; *Cluster-first route-second*; *VRP Solver*.

ÍNDICE

RESUMO	I
ABSTRACT	II
AGRADECIMENTOS	IX
CAPÍTULO 1. INTRODUÇÃO	1
CAPÍTULO 2. REVISÃO DE LITERATURA	3
2.1. Transportes Públicos em Zonas Rurais.....	3
2.2. Serviços de Transportes Urbanos de Bragança	4
CAPÍTULO 3. METODOLOGIA	6
3.1. Problema de Otimização de Redes de Transporte Público	6
3.2. Meta-heurísticas.....	8
3.2.1. Meta-heurística <i>Simulated Annealing</i>	8
3.2.2. Meta-heurística <i>Tabu Search</i>	8
3.2.3. Meta-heurística <i>GRASP</i>	9
3.2.4. Seleção da Meta-heurística	9
3.3. Heurísticas Construtivas	11
3.3.1. Heurística de <i>Savings</i>	12
3.3.2. Heurísticas de <i>Cluster - First / Route – Second</i>	12
3.3.3. Heurísticas de <i>Route - First / Cluster - Second</i>	12
3.3.4. Seleção da Heurística Construtiva.....	13
CAPÍTULO 4. TRATAMENTO DE DADOS E IMPLEMENTAÇÃO	15
4.1. Tratamento de Dados.....	15
4.2. Implementação.....	17
4.2.1. Algoritmo <i>Cluster-First / Route-second</i>	18
4.2.2. Meta-heurística <i>Tabu Search</i>	21
4.2.3. <i>Microsoft Excel VRP Spreadsheet Solver</i>	25
CAPÍTULO 5. RESULTADOS	27
5.1. Resultados do Algoritmo <i>Cluster-First / Route-Second</i>	27

5.2.	Resultados da Meta-heurística <i>Tabu Search</i>	27
5.3.	<i>Microsoft Excel VRP Spreadsheet Solver</i>	30
	CONCLUSÕES	33
	REFERÊNCIAS BIBLIOGRÁFICAS	35
	ANEXOS	38
i.	Tratamento de Dados	38
ii.	Resultados.....	41

ÍNDICE DE FIGURAS

FIGURA 1: DISTRIBUIÇÃO MODAL INTRAMUNICIPAL DOS MODOS PREFERENCIAIS DE DESLOCAÇÃO NA CIM -TTM PARA TRABALHADORES.....	5
FIGURA 2: DISTRIBUIÇÃO MODAL INTRAMUNICIPAL DOS MODOS PREFERENCIAIS DE DESLOCAÇÃO NA CIM -TTM PARA ESTUDANTES.....	5
FIGURA 3: REDE URBANA ATUAL DO CONCELHO DE BRAGANÇA FONTE: (TIS, 2020, PÁG. 45).....	5
FIGURA 4: EXEMPLO DE DIVISÃO DO ESPAÇO PARA FORMAÇÃO DOS CLUSTERS COM IGUAL NÚMERO DE PARAGENS.....	13
FIGURA 5: LOCALIZAÇÕES DAS PARAGENS ATUAIS, FORNECIDAS PELA CÂMARA MUNICIPAL DE BRAGANÇA.....	15
FIGURA 6: CÓDIGO DO GOOGLE SCRIPT PARA OBTENÇÃO DA MATRIZ DAS DISTÂNCIAS.....	17
FIGURA 7: SECÇÃO DO CÓDIGO PARA A CONSTRUÇÃO DO CLUSTER 1 E INICIALIZAÇÃO DA ROTA 1.....	19
FIGURA 8: SECÇÃO DO CÓDIGO PARA A CONSTRUÇÃO DA ROTA INICIAL 1.....	20
FIGURA 9: FLUXOGRAMA DA CRIAÇÃO DAS ROTAS INICIAS.....	21
FIGURA 10: SECÇÃO DO CÓDIGO DA REMOÇÃO DAS PARAGENS DA ROTA 1 A TROCAR PARA A ROTA 2 E ATUALIZAÇÃO DA DISTÂNCIA PERCORRIDA.....	22
FIGURA 11: SECÇÃO DO CÓDIGO DAS TROCAS 2-OTIMAS.....	23
FIGURA 12: FLUXOGRAMA DAS TROCAS 2-OTIMAS.....	23
FIGURA 13: SECÇÃO DE CÓDIGO PARA A SELEÇÃO DO MELHOR VIZINHO NA VIZINHANÇA.....	24
FIGURA 14: SELEÇÃO DOS RESULTADOS DE ITERAÇÕES DO TESTE 9.....	30
FIGURA 15: PEDIDO DE DISPONIBILIZAÇÃO DE DADOS À CÂMARA MUNICIPAL DE BRAGANÇA.....	38
FIGURA 16: MATRIZ DAS DISTÂNCIAS (MATRIZ_DISTÂNCIAS) EM METROS ENTRE CADA PAR DE PARAGENS.....	40
FIGURA 17: RESULTADOS DAS ITERAÇÕES DO TESTE 1.....	41
FIGURA 18: RESULTADOS DAS ITERAÇÕES DO TESTE 2.....	42
FIGURA 19: RESULTADOS DAS ITERAÇÕES DO TESTE 3.....	43
FIGURA 20: RESULTADOS DAS ITERAÇÕES DO TESTE 4.....	44
FIGURA 21: : RESULTADOS DAS ITERAÇÕES DO TESTE 5.....	45
FIGURA 22: RESULTADOS DAS ITERAÇÕES DO TESTE 6.....	46
FIGURA 23: RESULTADOS DAS ITERAÇÕES DO TESTE 7.....	47
FIGURA 24: RESULTADOS DAS ITERAÇÕES DO TESTE 8 (PARTE I).....	48
FIGURA 25: RESULTADOS DAS ITERAÇÕES DO TESTE 8 (PARTE II).....	49
FIGURA 26: RESULTADOS DAS ITERAÇÕES DO TESTE 8 (PARTE III).....	50
FIGURA 27: RESULTADOS DAS ITERAÇÕES DO ESTE 9 (PARTE I).....	51
FIGURA 28: RESULTADOS DAS ITERAÇÕES DO ESTE 9 (PARTE II).....	52
FIGURA 29: RESULTADOS DAS ITERAÇÕES DO ESTE 9 (PARTE III).....	53
FIGURA 30: RESULTADOS DAS ITERAÇÕES DO ESTE 9 (PARTE IV).....	54
FIGURA 31: RESULTADOS DAS ITERAÇÕES DO TESTE 10 (PARTE I).....	55
FIGURA 32: RESULTADOS DAS ITERAÇÕES DO TESTE 10 (PARTE II).....	56

FIGURA 33: RESULTADOS DAS ITERAÇÕES DO TESTE 11 (PARTE I).....	57
FIGURA 34: RESULTADOS DAS ITERAÇÕES DO TESTE 11 (PARTE II).....	58
FIGURA 35: RESULTADOS DO TESTE COM 3 LINHAS NO MICROSOFT EXCEL VRP SPREADSHEET SOLVER.....	59
FIGURA 36: RESULTADOS DO TESTE COM 2 LINHAS NO MICROSOFT EXCEL VRP SPREADSHEET SOLVER.....	60
FIGURA 37: RESULTADOS DO TESTE SEM AS 15 PARAGENS ADICIONAIS NO MICROSOFT EXCEL VRP SPREADSHEET SOLVER	61

ÍNDICE DE TABELAS

TABELA I: LISTA DAS PARAGENS CONSIDERADAS E RESPETIVAS COORDENADAS	16
TABELA II: PARÂMETROS FIXOS AO LONGO DOS TESTES DA META-HEURÍSTICA TABU- SEARCH.....	25
TABELA III: PARÂMETROS VARIÁVEIS AO LONGO DOS TESTES EFETUADOS	25
TABELA IV: MELHOR RESULTADO OBTIDO EM CADA TESTE EFETUADO	28

LSTA DE SIGLAS

STUB – Serviços de Transportes Urbanos de Bragança

CIM-TTM – Comunidade Intermunicipal das Terras de Trás-os-Montes

CMB – Câmara Municipal de Bragança

VBA - *Visual Basic for Applications*

Agradecimentos

Este projeto foi idealizado quando, passados quase 5 anos, voltei a passar mais de um mês seguido em Bragança, devido à pandemia. Foi tempo de me lembrar como, ao longo dos anos, muitas foram as vezes que andei a pé pela cidade sem nunca ponderar usar a rede de transportes públicos. O mesmo acontece com a grande maioria dos meus conhecidos.

Debruçar-me sobre um tema da minha cidade natal no meu Trabalho Final de Mestrado é, de certa forma, uma homenagem àquela que será sempre a minha casa.

Posto isto, começo por agradecer ao Professor Doutor José Pedro Romana Gaivão, por aceitar orientar-me neste tema e por toda a disponibilidade e apoio ao longo deste projeto.

À Câmara Municipal de Bragança, na pessoa do Dr. Miguel Abrunhosa, agradeço pela disponibilização dos dados que permitiram efetuar este estudo.

Às amigas de sempre, certezas para a vida, e aos que se tornaram pilares imprescindíveis ao longo dos últimos anos.

À minha família, que sempre me mostrou que tudo é possível com vontade e determinação.

Aos meus pais.

À Beatriz.

Capítulo 1. Introdução

O concelho de Bragança é servido por uma rede urbana de transportes públicos desde dezembro de 1984. Com uma frota urbana inicial de 4 autocarros, os Serviços de Transportes Urbanos de Bragança (STUB) vieram facilitar as deslocações dentro da cidade. Um ano mais tarde, deu-se o primeiro alargamento da rede às zonas rurais do concelho, passando a cobrir algumas das localidades na periferia da cidade de Bragança. A última grande reestruturação das linhas data de 2008 e resulta num conjunto de 4 linhas urbanas e 12 linhas rurais. Atualmente, as linhas rurais são treze.

A temática deste Trabalho Final de Mestrado surge do contacto com a realidade e importância deste serviço. O concelho de Bragança conta com 39 freguesias, 37 das quais rurais. Para além da população envelhecida, a maioria dos serviços primários encontram-se na sua sede. Assim, torna-se vital a existência de uma rede de transportes públicos compacta e capaz de responder às necessidades da população. Também a própria cidade de Bragança conta com bairros residenciais deslocados do centro, onde se encontram a grande maioria de escolas e serviços. O alargamento das linhas na primeira década dos anos 2000 permitiu angariar mais utilizadores, mas não em número suficiente para garantir a autossustentabilidade do serviço. O presente projeto pretende analisar a possibilidade de tornar a rede de transportes mais eficiente e atrativa, propondo-se a uma otimização das rotas utilizadas nesta rede, com base em heurísticas. Neste contexto, entende-se por rota o percurso de um veículo, que visita um conjunto de paragens, partindo e regressando ao depósito (garagem), num tempo limite. A relevância deste tema confirma-se pelo estudo que está a ser levado a cabo pela TiS, consultora da área da mobilidade e transportes, encomendado pela Câmara Municipal de Bragança.

O trabalho encontra-se dividido em cinco capítulos, começando pela Introdução, Revisão de Literatura, que apresenta uma breve introdução sobre a realidade e os desafios dos serviços de transportes públicos em zonas rurais, especificando a situação no concelho de Bragança. Segue-se a Metodologia, com a definição do problema e uma análise dos modelos aplicáveis, a Implementação, que

apresenta o processo de tratamento de dados e implementação dos algoritmos seleccionados e, por fim, os Resultados.

Capítulo 2. Revisão de Literatura

2.1. Transportes Públicos em Zonas Rurais

É ponto de acordo entre autores que os serviços de transporte público trazem grandes benefícios às comunidades urbanas e rurais (e.g. Burkhardt et al., 1998). Segundo Burkhardt et al (1998), a relação entre os serviços de transporte público em zonas rurais e a viabilidade económica dos mesmos deve ser medida de acordo com as seguintes questões:

- A que nível o transporte público rural tem impacto na atividade económica rural?
- De que forma o transporte público rural afeta segmentos específicos da economia?
- Em que medida o transporte público rural facilita o crescimento económico?

Os mesmos autores concluem ainda que dois fatores de sucesso a ter em conta no cálculo do impacto económico nas localidades rurais servidas por um determinado serviço de transportes públicos, são o facto de estes permitirem diferentes níveis de empregabilidade aos seus utentes e a oportunidade de muitos, vivendo sozinhos, terem acesso a cuidados médicos. Moseley (2003), considera que o foco social destes serviços em zonas rurais passa por:

- Diminuir o isolamento daqueles que sofrem privação de acessibilidade, de forma a poderem alcançar aquilo que desejam e/ou
- Levar até à população mais isolada mais daquilo que desejam.

O despovoamento destas áreas rurais de Portugal, em especial no interior, e a conseqüente falta de procura de bens e serviços, levaram a que o seu fornecimento se tenha tornado pouco atrativo (Sousa et al., 2009). Isto aplica-se aos serviços de transportes públicos uma vez que os fatores indicados como essenciais nesta problemática são a procura e o retorno, sendo que a primeira aumenta com a densidade populacional e a qualidade e abrangência do serviço prestado, e o segundo aumenta com a primeira (Carpinteiro, 2007).

Em resumo, segundo Pacione (referido em Sousa et al., 2009), os transportes públicos em áreas rurais devem conciliar baixo custo, bons níveis de

acessibilidade e alargada cobertura territorial. De acordo com um estudo levado a cabo em Inglaterra pela Social Exclusion Unit – Office of the Deputy Prime Minister (2003), formas de corresponder a tais exigências poderão passar pela criação de serviços de transportes mais flexíveis em que as rotas se adaptam às necessidades da procura.

2.2. Serviços de Transportes Urbanos de Bragança

O concelho de Bragança, com 34 589 habitantes¹, é atualmente servido pelo Serviço de Transportes Urbanos de Bragança (STUB), com quatro linhas urbanas e treze rurais, e que cobrem as áreas a Norte, Sul e Poente do concelho, e dois operadores privados, com serviço rural (TiS, 2020). As linhas urbanas efetuam serviço convencional com frequência de 30 minutos, à exceção de uma linha que tem uma frequência de 15 minutos e a particularidade de não apresentar paragens pré-estabelecidas. Todas atuam apenas nos dias úteis, entre as 7h45 e as 19h35 e apresentam circuitos circulares. No que diz respeito às linhas rurais, irradiam da cidade para as restantes localidades do concelho abrangidas por este serviço público. A frequência nestas linhas é escassa, respondendo à procura em cada um dos aglomerados populacionais cobertos, sendo que os principais utilizadores são estudantes e utentes com mais de 65 anos (Câmara Municipal de Bragança, 2010). A procura é muito reduzida, em especial nas freguesias mais distantes da cidade, mas a oferta é tida como muito razoável, uma vez que a maioria dos aglomerados estão cobertos, tanto em período escolar como de férias (TiS, 2020).

Os STUB têm uma grande importância social e de combate ao isolamento, funcionando também como serviço de transporte escolar. Desde a sua criação, foram já introduzidos vários tipos de descontos de acordo com o rendimento *per capita* das famílias e a idade dos utilizadores, em particular para estudantes. Em dezembro de 2009, a percentagem de passes ativos que não beneficiavam de reduções era inferior a 8%. Com isto, os resultados económicos da atuação do STUB são negativos desde a sua origem (Câmara Municipal de Bragança, 2010).

¹ Valores dos Censos de 2021

No entanto, a cobertura dada por este serviço de transportes públicos e os baixos valores praticados, aliados aos descontos existentes, não são suficientemente atrativos para que os habitantes do concelho o escolham como principal meio de deslocação. Nas Figuras 1 e 2 vemos que no concelho de Bragança o claro domínio dos modos de deslocação pertence ao *Automóvel*, com mais de 70% de utilização no caso de trabalhadores e com cerca de 60% para os estudantes. Para além da forte utilização automóvel, é evidente que o *Transporte Público* é ainda preterido quando comparado com *A pé* sendo que, no caso dos trabalhadores, o primeiro tem um peso residual (Bento et al., 2015).

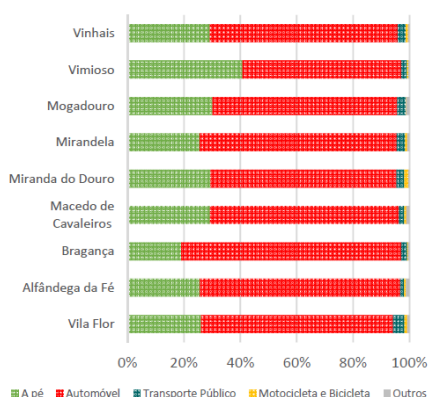


Figura 1: Distribuição modal intramunicipal dos modos preferenciais de deslocação na CIM - TTM para trabalhadores

(Fonte: Bento et al., 2015, pág 36)

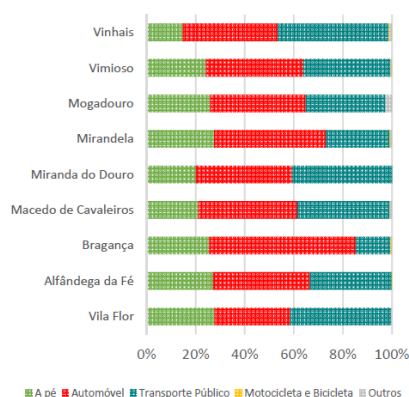


Figura 2: Distribuição modal intramunicipal dos modos preferenciais de deslocação na CIM - TTM para estudantes

(Fonte: Bento et al., 2015, pág 36)

Uma vez que as linhas rurais têm um carácter social de apoio às populações mais isoladas, é necessário que o seu perfil seja tão adaptado quanto possível às necessidades das mesmas. Existe já a intenção de fazer as linhas com menos procura migrarem para um serviço de transporte a pedido e, conseqüentemente, este projeto não irá estudar a otimização da rede rural.

A cidade de Bragança concentra grande parte da população do concelho e, portanto, é onde existe maior potencial de captação de utilizadores. No entanto, no que diz respeito à rede urbana, sobre a qual este projeto se irá debruçar, a procura tem vindo a decrescer desde 2011, com uma média de 6 pessoas por circulação, embora exista sobreocupação ao início do dia e à hora de almoço (TiS, 2020). O mesmo estudo apresenta como fatores que desincentivam a utilização deste serviço a existência de quatro linhas circulares unidireccionais e a necessidade de transbordos em quase todos os percursos, o que implica a

compra de mais do que um bilhete. Aponta ainda como propostas a utilização de apenas duas linhas circulares bidirecionais que cobrem todos os pontos e permitem duplicar o número de alternativas de ligação entre qualquer par origem-destino, reduzir o tempo médio de viagem de 17 para 12 minutos e garantir que todas as ligações são feitas em menos de 30 minutos, sendo que 68% são já atualmente feitas em menos de 15 minutos.

É necessário passar agora ao estudo dos métodos para a otimização da rede urbana do concelho de Bragança. A Figura 3 apresenta a rede urbana atual.

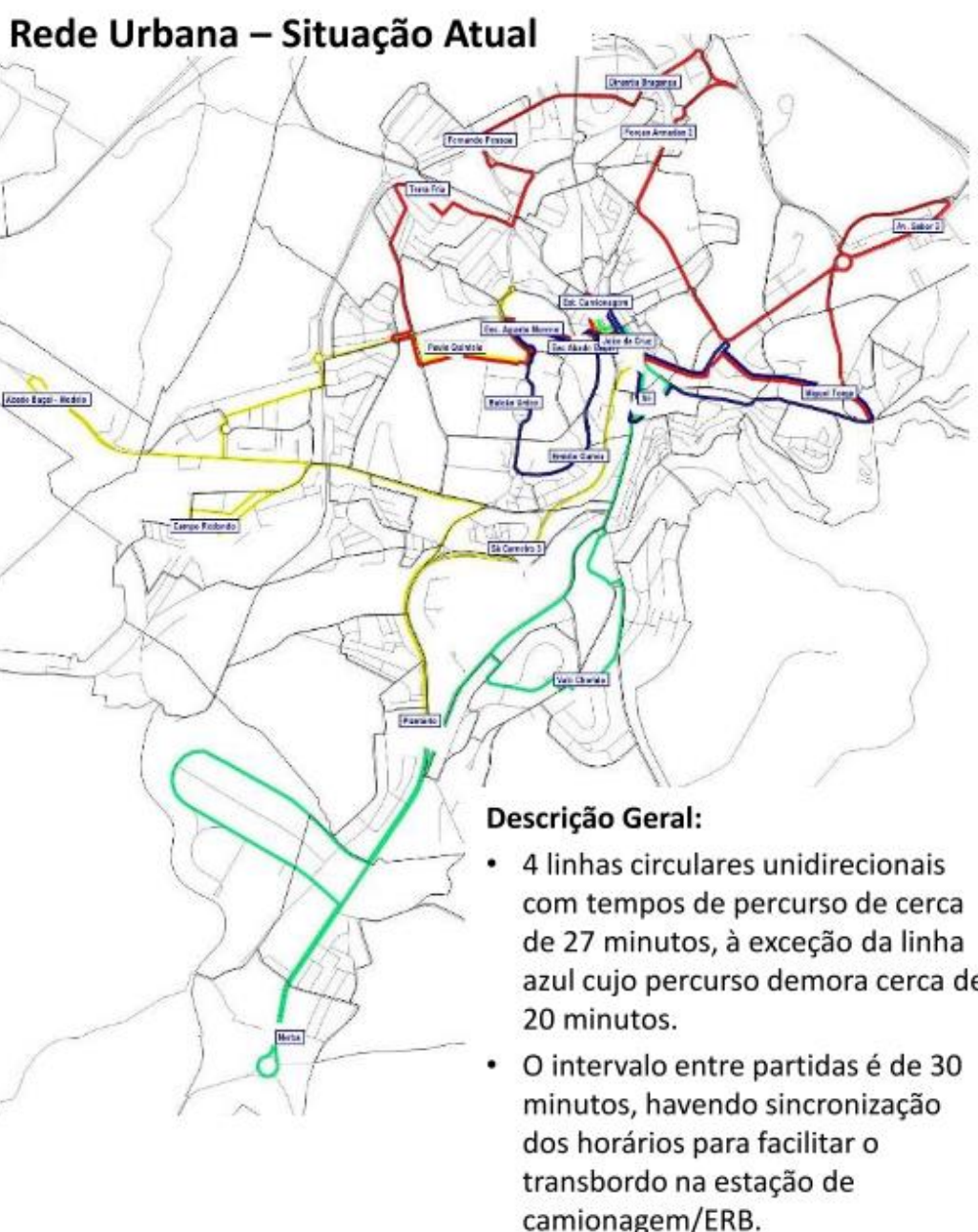


Figura 3: Rede Urbana atual do concelho de Bragança
 Fonte: (TiS, 2020, pág. 45)

Capítulo 3. Metodologia

3.1. Problema de Otimização de Rotas de uma Rede de Transporte Público

Nesta secção será feita a revisão de literatura referente à formulação e aos métodos a utilizar para resolver o problema a que este projeto se propõe. De forma mais abrangente, os problemas de transportes públicos têm como objetivo minimizar os custos de operação e de utilização (que consideram, além de outros custos, os tempos de espera e do percurso), podendo considerar-se multiobjectivo. De facto, para minimizar os tempos de espera dos utilizadores, é necessário aumentar a frequência com que os veículos percorrem as rotas, o que aumenta o custo de operação (Jha et al., 2019). Esta abordagem é utilizada por diversos autores para otimizar redes de transportes em autocarro (Fan & Machemehl, 2006; Afandizadeh et al., 2013; Jha et al., 2019). Existem ainda modelos que têm o objetivo adicional de minimizar o custo da procura não satisfeita (Fan & Machemehl, 2006). Esta corresponde às situações em que um utilizador prefere outro meio de transporte, uma vez que no serviço em questão teria de realizar um determinado número de trocas de linha ou não consegue embarcar, acabando por desistir (Guihaire & Hao, 2008; Arbex & Cunha, 2015). Quanto às restrições aplicadas e objetivos a cumprir, Zhao & Zeng (referido em Kepaptsoglou et al., 2009) destacam

1. Frequência viável para cada linha,
2. Capacidade máxima e mínima dos veículos da frota,
3. Extensão máxima e mínima e itinerário da rota das linhas,
4. Cobertura da rede,
5. Número de linhas na rede,
6. Número de veículos na frota,
7. Restrições orçamentais.

Guihaire & Hao (2008) consideram ainda o histórico da rota, a satisfação da procura e a coordenação entre as rotas, de forma a minimizar os tempos de espera nas trocas de linha.

Barbarosoglu & Ozgur (1999) apresentam um *Vehicle Routing Problem* (VRP) como o problema de determinar rotas ótimas através de um conjunto de localizações definidas num grafo orientado $G = (V, A)$, com $V = \{0\} \cup N = \{0, 1, \dots, n\}$ o conjunto de n paragens (vértices) e do depósito, e $A = \{(i, j) : i, j \in V, i \neq j\}$ o conjunto de arcos. O vértice 0 representa o depósito que serve como ponto de partida e chegada ao conjunto dos K veículos.

A formulação apresentada neste artigo e adaptada ao problema em questão é a seguinte:

$$\text{Min} \sum_{(i,j) \in A} \sum_{v=1}^K t_{ij} x_{ij}^v \quad (1)$$

$$\left\{ \begin{array}{l} \sum_{i:(i,j) \in A} \sum_{v=1}^K x_{ij}^v = 1 \quad \forall j \in V \end{array} \right. \quad (2)$$

$$\left\{ \begin{array}{l} \sum_{i:(i,p) \in A} x_{ip}^v - \sum_{j:(p,j) \in A} x_{pj}^v = 0 \quad \forall p \in V, v \in \{1, \dots, K\} \end{array} \right. \quad (3)$$

$$\left\{ \begin{array}{l} \sum_{(i,j) \in A} t_{ij}^v x_{ij}^v \leq T_v \quad \forall v \in \{1, \dots, K\} \end{array} \right. \quad (4)$$

$$\text{s. a:} \left\{ \begin{array}{l} \sum_{(0,j) \in A} x_{0j}^v \leq 1 \quad \forall v \in \{1, \dots, K\} \end{array} \right. \quad (5)$$

$$\left\{ \begin{array}{l} \sum_{(i,0) \in A} x_{i0}^v \leq 1 \quad \forall v \in \{1, \dots, K\} \end{array} \right. \quad (6)$$

$$\left\{ \begin{array}{l} \sum_{i \in S} \sum_{j \in V \setminus S} \sum_{v=1}^K x_{ij}^v \geq 1 \quad \forall S \subset V \end{array} \right. \quad (7)$$

$$\left\{ \begin{array}{l} x_{ij}^v \in \{0, 1\} \quad \forall i, j \in V, v \in \{1, \dots, K\} \end{array} \right. \quad (8)$$

Consideram-se t_{ij}^v o tempo de viagem do veículo v entre i e j , t_i^v o tempo que o veículo v demora na paragem i , T_v a duração máxima permitida da rota percorrida pelo veículo v e x_{ij}^v a variável binária que toma o valor 1 quando o veículo v faz a ligação entre i e j .

3.2. Meta-heurísticas

3.2.1. Meta-heurística *Simulated Annealing*

Kuo (2014) propõe o *Simulated Annealing* (SA) como método de otimização de rotas de transportes com linhas de tipo linear, circular ou híbrido. Este algoritmo é adaptado da física e se, partindo de uma solução X , encontra uma melhor X' , então o movimento de X para X' é aceite, continuando a procura por uma solução melhor até se atingir um critério de paragem. No caso de não ser possível encontrar uma solução melhor que X , o movimento de X para uma solução pior, X'' , é aceite com probabilidade $e^{\left(\frac{-\Delta Z}{U}\right)}$ (sendo U o valor da “temperatura”, definida antes da primeira iteração e sujeita a uma taxa de arrefecimento α , e $\Delta Z = Z(X'') - Z(X)$ a variação no valor da função objetivo gerada pelo movimento de X para X''). Este movimento para uma solução pior pretende evitar a escolha de um ótimo local como solução. Se ΔZ tomar um valor grande, o que indica que X'' é muito pior que X , então a probabilidade de o movimento ser aceite é baixa. No entanto, com o aumento do número de iterações e a diminuição da “temperatura”, a probabilidade de se aceitar um movimento para um valor pior da função objetivo tende a aumentar.

3.2.2. Meta-heurística *Tabu Search*

O *Tabu Search* explora o campo de soluções movimentando-se para a solução com melhor valor da função objetivo na vizinhança (conjunto de rotas alternativas à solução admissível atual, próximas da mesma e obtidas com base em variações desta), mesmo que este seja um movimento não melhorativo. No entanto, para alargar as buscas e evitar regressar a soluções já encontradas, aquelas que são selecionadas no final de cada iteração (X^i , com i o número da iteração), entram para a *Lista Tabu*, onde são guardadas por um período pré-definido. Partindo de uma solução admissível, se a melhor solução admissível encontrada em determinada iteração fizer parte da Lista Tabu nesse momento, então passa-se à segunda melhor. Este algoritmo permite atingir uma boa solução num período aceitável de tempo (Fan & Machemehl, 2008).

3.2.3. Meta-heurística GRASP

O Algoritmo GRASP divide-se em duas fases:

1. Fase de construção de uma solução admissível:

Partindo de um vértice aleatório inicial i encontram-se os $m \geq 2$ vértices que, ao serem acrescentados a i na rota, respeitam todas as restrições. Destes m vértices seleciona-se aleatoriamente o que entra na rota. Os restantes vértices vão sendo inseridos na rota, de acordo com o mesmo racional, mas tendo em conta todos os vértices que já se juntaram ao vértice i desde a iteração inicial. O processo repete-se até se formar um conjunto de rotas que constitua uma solução admissível para o problema, X^k .

2. Fase melhorativa

Através de um algoritmo de pesquisa local procura-se, na vizinhança da melhor solução encontrada até ao momento, o conjunto de rotas que representem uma solução admissível, X' , e com melhor valor de função objetivo. Caso a solução X' seja melhor do que X^k , então atualiza-se $X^k = X'$.

Esta fase repete-se até que seja atingido um critério de paragem (García-Archilla et al., 2013).

3.2.4. Seleção da Meta-heurística

De entre os algoritmos brevemente apresentados acima, será feita a seleção daquele que virá a ser utilizado na resolução do problema a que este projeto se propõe. Tanto o algoritmo *Tabu Search* como o SA têm particularidades interessantes e que serão úteis na obtenção de uma boa solução. O último algoritmo exposto, aparenta ser o mais limitado, não tendo características que permitam uma abrangência tão grande de soluções intermédias. Por isto, será excluído das hipóteses. Battiti & Tecchiolli (1994) apresentam um estudo comparativo entre as duas meta-heurísticas que restam. Este destaca que no início da procura, o *Simulated Annealing* converge para um ponto de atração, enquanto o *Tabu Search* avalia diversos pontos da vizinhança. Se a solução encontrada pelo SA não for admissível, o algoritmo volta ao início e tem de correr

novamente para produzir uma solução que o seja, enquanto o segundo usa toda a informação disponível para efetuar a pesquisa, diversificando-a. Os autores defendem ainda que o esforço computacional acrescido pela memória a longo ou médio prazo necessária no *Tabu Search* para armazenar a Lista Tabu é compensado pela “fuga” a polos de atração (sequência de soluções intermédias que levam o algoritmo até um ótimo local) com elevada probabilidade e que prejudicam os valores da função objetivo. Posto isto, este trabalho irá utilizar a meta-heurística *Tabu Search*, cujo algoritmo, baseado na publicação de *Barbarosoglu & Ozgur (1999)* e adaptado para o caso em que a rede inicial foi criada com base em *clusters*, se encontra abaixo:

Meta-heurística Tabu Search adaptada ao problema proposto

0. Definir N_{max} , α_{max} , β_{max} , M , P , θ_{min} , θ_{max} com
 - N_{max} número máximo de iterações sem qualquer melhoria,
 - α_{max} número máximo de vizinhos em cada vizinhança,
 - β_{max} número máximo de trocas 2-otimais (procedimento descrito em seguida),
 - M , P número máximo de vértices a trocar entre as rotas,
 - θ_{min} , θ_{max} limites inferior e superior, respetivamente, do tempo tabu (contabilizado em iterações)
1. Criação de um novo vizinho:
 - 1.1. Selecionar duas rotas de forma aleatória: $R1$ e $R2$;
 - 1.2. Obter $R1'$ e $R2'$, rotas resultantes da remoção de $\mu = \min(M, \# \text{vértices de } R1)$ e $\pi = \min(P, \# \text{vértices de } R2)$ vértices selecionados de forma aleatória, respetivamente;
 - 1.3. Inserir os vértices do conjunto correspondente a π em $R1'$ (rota $R1$ após a remoção dos vértices), gerando $R1''$ e os vértices do conjunto correspondente a μ em $R2'$, obtendo $R2''$, de acordo com o *procedimento de inserção*, descrito adiante;
 - 1.4. Aplicar o *procedimento 2-otimal* a $R1''$ e $R2''$, de forma independente, β_{max} vezes, de forma a tentar melhorar as rotas selecionadas;
 - 1.5. Determinar a distância percorrida em $R1''$ e $R2''$;
 - 1.6. Calcular o valor da função objetivo do vizinho criado.
2. Repetir o passo 1 α_{max} vezes, de forma a obter o total de vizinhos pretendidos.
3. Implementar o movimento não-tabu com menor valor da função objetivo e que gere uma solução admissível e adicioná-lo à lista tabu de onde

só será removido após θ iterações, com $\theta_{min} < \theta \leq \theta_{max}$. Obter a solução da iteração.

procedimento de inserção

Inserir os vértices de um conjunto $S \subset V$ na rota R :

Input. S, R ;

1. Encontrar o vértice $k \in S$ com menor c_{ki} , i.e. menor distância de k a qualquer vértice i já em R ;
2. Encontrar o arco $(i, j) \in R$ que minimiza $c_{ik} + c_{kj} - c_{ij}$ e inserir k em R , entre i e j ; Retirar k de S ;
3. Repetir 1 e 2 até $S = \Phi$;

Output. Nova rota R , com vértices de S .

Procedimento 2-otimal

Input. R ;

1. Selecionar dois arcos distintos não consecutivos de R , seja $X = \{(i, i + 1), (j, j + 1)\}$ e calcular $\delta_x = (c_{i,i+1} + c_{j,j+1}) - (c_{ij} + c_{i+1,j+1})$, distância poupada na substituição dos arcos;
2. Se $\delta_x > 0$, substituir em R os arcos de X por (i, j) e $(i + 1, j + 1)$ e inverter em R o caminho de $i + 1$ a j ;
3. Repetir para todos outros conjuntos X selecionados aleatoriamente;

Output: Novo vizinho na vizinhança.

Como qualquer meta-heurística, o *Tabu Search* inicia-se com a construção de uma solução admissível a partir da qual se fará a pesquisa para tentar melhorar a solução admissível corrente. Passar-se-á ao estudo de algumas heurísticas construtivas e à seleção da mais adequada para o problema em questão.

3.3. Heurísticas Construtivas

As heurísticas construtivas analisadas como hipótese para a criação das rotas que servirão como base ao *Tabu Search* foram o algoritmo de *Savings*, ou de Clarke e Wright, um método de tipo *Cluster-First / Route-Second* e outro de tipo *Route-First / Cluster-Second*.

3.3.1. Heurística de *Savings*

O algoritmo de *Savings* começa por servir cada nó de forma independente (n rotas, para n pontos ligados ao “depósito”). Em cada iteração procede à ligação entre rotas por ordem de maior *saving*. O *saving* de juntar a rota $R' = ((0, i), (i, i + 1), \dots)$ no fim da rota $R = (\dots, (j - 1, j), (j, 0))$ é calculado por $s_{ij} = c_{0i} + c_{j0} - c_{ji}$ (Clarke & Wright, 1964). Esta heurística implica o cálculo de *savings* entre cada par de rotas iniciais, formadas com um só vértice e o depósito.

3.3.2. Heurísticas de *Cluster - First / Route - Second*

Passando agora a heurísticas de tipo *Cluster - First / Route - Second*, estas consistem em começar por determinar o número de *clusters* que se pretende criar, passando à partição dos nodos do grafo e geração de uma rota admissível a partir de cada *cluster* (Deitch & Ladany, 2000). Neste artigo, para um problema de rotas turísticas, os *clusters* são construídos com base na atratividade dos locais. Fisher (1995) salienta a importância da criação de *clusters* entre clientes que se encontram fisicamente próximos e afastados dos restantes, uma vez que estes devem ser incluídos na mesma rota, sob pena de haver um impacto significativo nos custos se tal não acontecer.

3.3.3. Heurísticas de *Route - First / Cluster - Second*

A terceira hipótese estudada foi uma heurística de tipo *Route - First / Cluster - Second*. O algoritmo apresentado por Beasley (1983) inicia-se formando uma rota “gigante” que visita todos os clientes da rede, com início e fim no “depósito”, e que é posteriormente partida em *clusters*. A divisão da rota inicial em diferentes rotas baseia-se numa matriz de elementos $d_{ij} = c_{0,(i+1)} + \sum_{k=i+1}^{j-1} c_{k,(k+1)} + c_{j0}$ (com c_{ij} a distância entre clientes ou entre estes e o depósito), se a rota $(0, i, i + 1, \dots, j, 0)$ é admissível, ou $d_{ij} = \infty$, caso contrário. Estes caminhos de menor custo que constituem a matriz $D = [d_{ij}]$ são obtidos com base no algoritmo de caminho mais curto de Floyd, sendo que se tratam das diversas partições possíveis do caminho inicial. O conjunto ótimo de partições é então selecionado. A desvantagem deste método, comparativamente ao anterior, será o facto de

exigir a criação da rota com todos os nodos do circuito. Computacionalmente, esta será uma tarefa demorada, acrescida ainda à utilização de um algoritmo de caminho mais curto para encontrar as possíveis partições do caminho.

3.3.4. Seleção da Heurística Construtiva

Assim, a meta-heurística *Tabu Search* será iniciada com a construção da solução inicial a partir de uma heurística do tipo *Cluster - First / Route - Second*. Quanto a esta, Sultana et al. (2017) apresentam uma variação do método original de Fisher (1995), que será o algoritmo usado neste trabalho:

Algoritmo Cluster – First / Route – Second

1. Partição da rede em K (número de veículos) *clusters*, de modo a que cada *cluster* contenha um número semelhante de clientes (Figura 4);

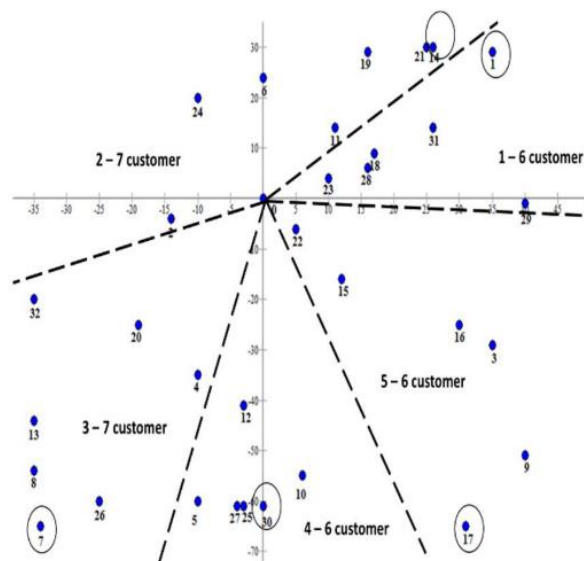


Figura 4: Exemplo de divisão do espaço para formação dos clusters com igual número de paragens

Fonte: (Sultana et al., 2017, pág. 714)

2. Em cada *cluster*, selecionar para “semente” o cliente mais distante do “depósito” e criar uma rota com o depósito e a semente;
3. Selecionar um *cluster* que ainda não tenha rota desenhada;
4. Para cada vértice i no *cluster* calcular $c_i = c_{Si} + c_{i0} - c_{S0}$, o custo de inserir i entre a semente s e o depósito;
5. Adicionar os clientes do *cluster* à rota por ordem crescente de custo de inserção, respeitando as restrições de duração da rota;
6. Repetir 3-5 até todos os clientes de todos os *clusters* estarem inseridos numa rota

O algoritmo descrito permite obter uma solução admissível, ou seja, um conjunto de rotas que cobre todos os nós da rede e respeita as restrições do problema.

Capítulo 4. Tratamento de Dados e Implementação

4.1. Tratamento de Dados

Como resposta ao pedido efetuado, disponibilizado no Anexo i., Figura 15, a Câmara Municipal de Bragança (CMB) forneceu a localização exata de cada uma das paragens da rede atual em formato *.kml*. O conjunto destas rotas percorre um total de 34 896 metros, com 12 030 metros na linha 1 (U1), 11 426 metros na linha 2 (U2) e 11 440 metros na linha 3 (U3) (a CMB disponibilizou a localização das paragens divididas em apenas três linhas, excluindo a linha sem paragens pré-estabelecidas). De forma a permitir a diversificação da pesquisa face às rotas atuais, nos casos em que a paragem se localiza em apenas um dos lados da via e existe possibilidade de circular nos dois sentidos, foi adicionada uma paragem adicional no sentido contrário (ex: paragens U2 2 e U2 2'). De seguida, apresenta-se a distribuição das paragens existentes, bem como a lista da totalidade das paragens consideradas para este estudo, com as respetivas coordenadas (Figura 5 e Tabela I, respetivamente).



Figura 5: Localizações das paragens atuais, fornecidas pela Câmara Municipal de Bragança

Linha	Paragem	#	Coordenadas
U1	1	1	41°48'34"N 6°45'39"W
U1	2	2	41°48'27"N 6°45'32"W
U1	3	3	41°48'24"N 6°45'30"W
U1	4	4	41°48'19"N 6°45'31"W
U1	5	5	41°48'12"N 6°45'33"W
U1	7	6	41°47'48"N 6°45'53"W
U1	8	7	41°47'42"N 6°46'02"W
U1	9	8	41°47'33"N 6°46'11"W
U2	20	53	41°48'02"N 6°46'52"W
U2	21	54	41°48'07"N 6°46'43"W
U2	22	55	41°48'12"N 6°46'33"W
U2	23	56	41°48'09"N 6°46'20"W
U2	24	57	41°48'06"N 6°46'05"W
U2	25	58	41°48'00"N 6°46'09"W
U2	26	59	41°47'48"N 6°46'17"W
U2	26'	60	41°47'48"N 6°46'16"W
U2	27	61	41°47'40"N 6°46'14"W

Linha	Paragem	#	Coordenadas
U1	10	9	41°47'24"N 6°46'15"W
U1	11	10	41°47'18"N 6°46'33"W
U1	12	11	41°47'26"N 6°46'45"W
U1	13	12	41°47'28"N 6°47'03"W
U1	14	13	41°47'18"N 6°46'50"W
U1	15	14	41°47'14"N 6°46'43"W
U1	16	15	41°47'06"N 6°46'33"W
U1	17	16	41°46'59"N 6°46'37"W
U1	18	17	41°47'00"N 6°46'39"W
U1	19	18	41°46'50"N 6°46'43"W
U1	20	19	41°46'59"N 6°46'39"W
U1	21	20	41°47'06"N 6°46'32"W
U1	22	21	41°47'14"N 6°46'23"W
U1	23	22	41°47'22"N 6°46'16"W
U1	26	23	41°47'38"N 6°45'56"W
U1	27	24	41°47'38"N 6°45'49"W
U1	28	25	41°47'38"N 6°45'43"W
U1	29	26	41°47'42"N 6°45'37"W
U1	30	27	41°47'53"N 6°45'34"W
U1	31	28	41°48'00"N 6°45'38"W
U1	33	29	41°48'20"N 6°45'26"W
U1	34	30	41°48'26"N 6°45'30"W
U2	2	31	41°48'29"N 6°45'41"W
U2	2'	32	41°48'28"N 6°45'41"W
U2	3	33	41°48'32"N 6°45'52"W
U2	3'	34	41°48'31"N 6°45'52"W
U2	4	35	41°48'26"N 6°45'54"W
U2	5	36	41°48'27"N 6°46'05"W
U2	6	37	41°48'29"N 6°46'13"W
U2	7	38	41°48'30"N 6°46'18"W
U2	7'	39	41°48'29"N 6°46'19"W
U2	8	40	41°48'28"N 6°46'27"W
U2	8'	41	41°48'27"N 6°46'27"W
U2	9	42	41°48'23"N 6°46'33"W
U2	10	43	41°48'22"N 6°46'37"W
U2	11	44	41°48'17"N 6°46'51"W
U2	12	45	41°48'14"N 6°46'53"W
U2	13	46	41°48'14"N 6°47'07"W
U2	14	47	41°48'22"N 6°47'24"W
U2	15	48	41°48'15"N 6°47'13"W
U2	16	49	41°48'13"N 6°46'53"W
U2	17	50	41°48'12"N 6°46'38"W
U2	18	51	41°48'05"N 6°46'50"W
U2	19	52	41°48'02"N 6°46'57"W
U2	20	53	41°48'02"N 6°46'52"W
U2	28	62	41°47'33"N 6°46'13"W
U2	28'	63	41°47'33"N 6°46'14"W
U2	29	64	41°47'40"N 6°46'13"W
U2	30	65	41°47'58"N 6°46'07"W
U2	30'	66	41°47'59"N 6°46'07"W
U2	31	67	41°47'58"N 6°45'53"W
U2	32	68	41°48'07"N 6°45'42"W
U2	33	69	41°48'17"N 6°45'36"W
U3	3	70	41°48'24"N 6°45'22"W
U3	4	71	41°48'26"N 6°45'14"W
U3	4'	72	41°48'26"N 6°45'16"W
U3	5	73	41°48'23"N 6°45'10"W
U3	6	74	41°48'21"N 6°44'53"W
U3	7	75	41°48'16"N 6°44'45"W
U3	8	76	41°48'28"N 6°44'52"W
U3	9	77	41°48'46"N 6°44'37"W
U3	10	78	41°48'43"N 6°44'35"W
U3	11	79	41°48'39"N 6°44'48"W
U3	12	80	41°48'33"N 6°45'03"W
U3	12'	81	41°48'33"N 6°45'04"W
U3	13	82	41°48'29"N 6°45'14"W
U3	13'	83	41°48'29"N 6°45'15"W
U3	14	84	41°48'38"N 6°45'21"W
U3	15	85	41°48'49"N 6°45'29"W
U3	15'	86	41°48'49"N 6°45'30"W
U3	16	87	41°48'58"N 6°45'23"W
U3	16'	88	41°48'58"N 6°45'24"W
U3	17	89	41°49'05"N 6°45'11"W
U3	17'	90	41°49'06"N 6°45'11"W
U3	18	91	41°49'08"N 6°45'15"W
U3	19	92	41°49'05"N 6°45'28"W
U3	20	93	41°49'04"N 6°45'39"W
U3	21	94	41°49'03"N 6°45'49"W
U3	22	95	41°48'58"N 6°46'00"W
U3	22'	96	41°48'58"N 6°46'01"W
U3	23	97	41°48'54"N 6°45'55"W
U3	24	98	41°48'52"N 6°45'51"W
U3	24'	99	41°48'52"N 6°45'50"W
U3	25	100	41°48'46"N 6°45'55"W
U3	26	101	41°48'49"N 6°46'04"W
U3	27	102	41°48'50"N 6°46'11"W
U3	28	103	41°48'48"N 6°46'15"W
U3	29	104	41°48'43"N 6°46'18"W
U3	30	105	41°48'31"N 6°46'14"W

Tabela I: Lista das paragens consideradas e respetivas coordenadas

De forma a obter a matriz de distâncias diretas, em metros, entre paragens, foi executado um *Google script* que, considerando todos os pares possíveis de coordenadas, devolve do *Google Maps* a menor distância encontrada percorível de carro. Na Figura 6 encontra-se o código do *Google script*. A matriz resultante

(matriz_distâncias) pode ser visualizada na Figura 16 no Anexo i..

```

var app=SpreadsheetApp;
var ui=app.getUi();

function onOpen()
{
ui.createMenu("Mapas").addItem("Exibir Dados", "myMap").addToUi();
}

function myMap()
{
for(i=533; i<=7226;i++){
var sheet=app.getActiveSpreadsheet().getSheetByName('Coordenadas');
var maps=Maps;
var directions=maps.newDirectionFinder()
.setOrigin(sheet.getRange(i,1).getValue())
.setDestination(sheet.getRange(i,2).getValue())
.setMode(Maps.DirectionFinder.Mode.DRIVING)
.getDirections();

var legs=directions.routes[0].legs[0];

sheet.getRange(i,3).setValue(legs.distance.value);

var steps=legs.steps;
var html="";
steps.map(function(elem,ind,obj){
html+=elem.html_instructions+'\n';
sheet.getRange(i,4).setValue(html.replace(/<[^>]*>?/gm, ''));}})}

```

Figura 6: Código do Google script para obtenção da matriz das distâncias

4.2. Implementação

Apesar da proposta no estudo efetuado por TiS (2020), vão ser mantidas as três rotas neste projeto, deixando apenas cair a linha sem paragens pré-estabelecidas, com o objetivo de perceber se é possível melhorar as rotas sem alterar completamente a sua estrutura. Posteriormente, será utilizado o *Microsoft Excel VRP Spreadsheet Solver* para

- i) a criação de um conjunto de três rotas, de forma a validar a seleção de algoritmos feita neste trabalho;
- ii) um conjunto de apenas duas rotas, que permite avaliar o impacto da proposta da TiS, de construir uma rede com duas linhas apenas; e
- iii) um teste que permite avaliar o impacto das 15 paragens adicionadas.

O *Microsoft Excel VRP Spreadsheet Solver* é uma ferramenta *open source* que permite identificar soluções admissíveis para *Vehicle Routing Problems*.

4.2.1. Algoritmo *Cluster-First / Route-second*

Para a implementação deste algoritmo, começou por ser efetuada a divisão em *clusters* das paragens a considerar. Tendo em conta que todos os *clusters* incluem o ponto #1, que será a origem e destino das três rotas, a divisão das restantes paragens foi feita de acordo com a sua proximidade e de forma a garantir o equilíbrio entre as rotas. Assim, este estudo parte dos seguintes *clusters*:

- *Cluster 1*: #1, #2, #3, ... #35;
- *Cluster 2*: #1, #36, #37, ... #70;
- *Cluster 3*: #1, #71, #72, ... #105.

As sementes de cada *cluster* são, respetivamente, #15, #48, #90.

O código na Figura 7 começa por armazenar as paragens que constituem o *cluster x* e determinar a respetiva semente (*semente_x*). De seguida, estabelece-se que a construção da rota *x* se inicia com (#1, *semente_x*, #1) e que a correspondente distância inicial é a soma da distância de #1 à *semente_x* com a distância de volta a #1.

```
i = 1;
dist = 0
Do While i <= 35
    ReDim Preserve cluster1(1 To i)
    cluster1(i) = matriz_distâncias(1, i)
    If matriz_distâncias(2, i + 1) + matriz_distâncias(i + 1, 2) > dist
Then
        dist = matriz_distâncias(2, i + 1) + matriz_distâncias(i + 1,
2) 'dist vai permitir perceber qual o nodo mais distante da origem em
cada cluster
        semente1 = i
    End If
    i = i + 1
Loop
```

```
dist_rota1 = matriz_distâncias(2, semente1 + 1) + matriz_distâncias
(semente1 + 1, 2)
```

Figura 7: Secção do código para a construção do cluster 1 e inicialização da rota 1

Por fim, as restantes paragens no *cluster* têm de ser acrescentadas à rota, por ordem de menor distância. Para isto, foram criadas as seguintes matrizes, que permitem seleccionar qual a melhor paragem a inserir na rota tendo em conta as paragens que esta já inclui:

- *CálculoDistâncias_Matriz*, que tem dimensão n° de paragens no cluster \times n° de arcos entre paragens na rota, e guarda as distâncias adicionais a percorrer caso se adicione qualquer uma das paragens no cluster entre qualquer par de paragens já na rota;
- *Nós_Matriz*, com a mesma dimensão, permite armazenar a paragem que corresponde a cada distância calculada na matriz anterior;
- *Posições_Matriz*, também com igual dimensão, guarda a posição em que a paragem armazenada na matriz anterior irá entrar na rota, caso seja a seleccionada.

Após obter todas as possíveis distâncias adicionais de inserção das paragens, selecciona-se a melhor opção e identifica-se a respetiva paragem, “*p*”, e posição em que irá ser inserida, “*q*”. Assim, a distância adicional de inserção da paragem “*p*” deve ser somada à distância já percorrida ao longo da rota e a paragem “*p*” deve ser inserida na posição “*q*” da rota.

A Figura 8 apresenta a secção do código para o *cluster 1* e a Figura 9 o fluxograma que resume os passos do algoritmo após a inicialização da rota, a título de exemplo.

```
Do While rota1.Count < UBound(cluster1) + 1 'Até que todas as paragens
entrem na rota
    i = 1
    j = 1
    a = 0
    e = 1
Do While i < UBound(cluster1) + 1 'Obtenção das distâncias adicionais de
inserir cada paragem entre qualquer par já presente na rota
```

```

        ReDim Preserve CálculoDistâncias_Matriz(1 To UBound(cluster1), 1 To
rota1.Count - 1)
        ReDim Preserve Nós_Matriz(1 To UBound(cluster1), 1 To rota1.Count
- 1)
        ReDim Preserve Posições_Matriz(1 To UBound(cluster1), 1 To
rota1.Count - 1)
        b = pertence(rota1, i) 'Função Boolean que devolve True caso a
paragem i se encontre na rota1
        a = a + 1
        If b <> True Then
            e = 1
            For j = 1 To rota1.Count - 1
                CálculoDistâncias_Matriz(a, e) = matriz_distâncias
(rota1(j) + 1, i + 1) + matriz_distâncias(i + 1, rota1(j + 1) + 1) -
matriz_distâncias(rota1(j) + 1, rota1(j + 1) + 1)
                Nós_Matriz(a, e) = i
                Posições_Matriz(a, e) = j
                e = e + 1
            Next j
        Else
            e = 1
            For j = 1 To rota1.Count - 1
                CálculoDistâncias_Matriz(a, e) = 10000
                e = e + 1
            Next j
        End If
        i = i + 1
    Loop
    C = 10000
    k = 1
    Do While k <= UBound(CálculoDistâncias_Matriz, 1) 'Seleção da melhor
opção
        For l = 1 To UBound(CálculoDistâncias_Matriz, 2)
            If CálculoDistâncias_Matriz(k, l) < C Then
                C = CálculoDistâncias_Matriz(k, l)
                p = Nós_Matriz(k, l)
                q = Posições_Matriz(k, l)
            End If
        Next l
        k = k + 1
    Loop
    dist_rota1 = dist_rota1 + C 'Atualização da distância percorrida ao longo
da rota
    rota1.Add p, After:=q 'Atualização da rota
    Loop

```

Figura 8: Seccção do código para a construção da rota inicial 1

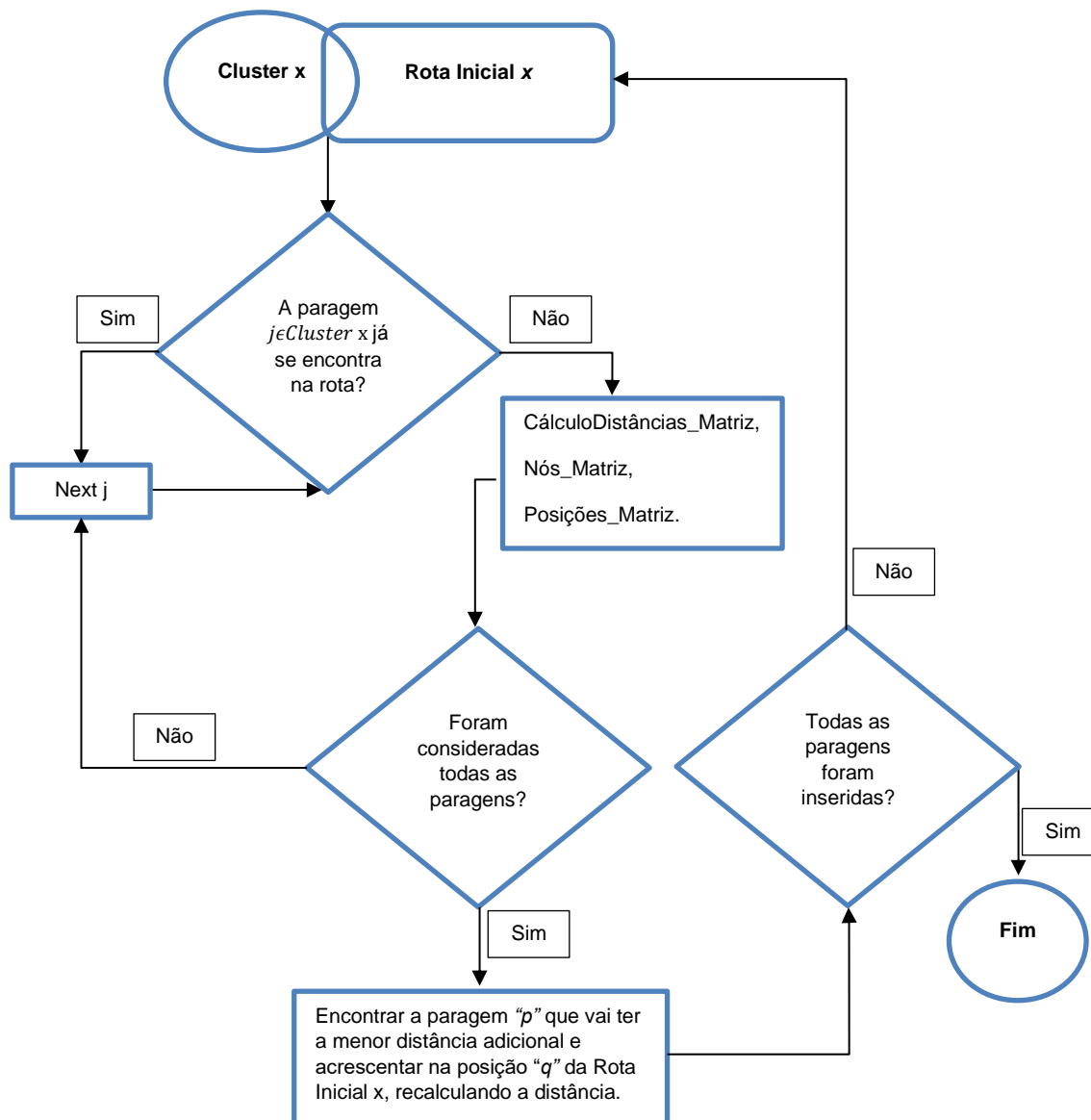


Figura 9: Fluxograma da criação das rotas iniciais

No final deste processo, serão obtidas as rotas iniciais, sobre as quais será aplicada a meta-heurística *Tabu Search*.

4.2.2. Meta-heurística *Tabu Search*

O código apresentado na Figura 10, que tem por base as rotas iniciais obtidas com o algoritmo *Cluster – First / Route - Second*, repete-se até que a vizinhança tenha o número de vizinhos pretendido.

Esta meta-heurística inicia a construção de cada vizinho com a seleção aleatória das duas rotas sobre as quais serão efetuadas alterações e as paragens a trocar entre estas. A partir daí passa-se à remoção das paragens a sair de cada uma das rotas seleccionadas e à atualização da distância percorrida.

```

i = 1
Do While i <= num_nós_a_sair(1) 'Substituição das paragens nas posições
seleccionadas por zero - não posso ir removendo porque altera as posições
    If r1.Item(posição_nós_a_sair(i)) <> 0 Then
        r1.Remove posição_nós_a_sair(i)
        r1.Add 0, Before:= posição_nós_a_sair(i)
    End If
    i = i + 1
Loop
i = 1
Do While i <= r1.Count 'Remoção dos zeros
    If r1.Item(i) = 0 Then
        r1.Remove i
    Else
        i = i + 1
    End If
Loop
dist_rotal = 0
For i = 1 To r1.Count - 1 'Atualização da distância percorrida
    dist_rotal = dist_rotal + matriz_distâncias(r1.Item(i) + 1,
r1.Item(i + 1) + 7)

```

Figura 10: Secção do código da remoção das paragens da rota 1 e atualização da distância percorrida

A inserção das paragens que saem de R1 para R2 e vice-versa, fez-se com base nas mesmas matrizes criadas no algoritmo *Cluster - First / Route – Second*. Após a atualização das duas rotas seleccionadas, passa-se à aplicação das trocas 2-otimais, cuja secção de código se apresenta na Figura 11.

```

'R1
loti(1) = r1.Item(trocas2ot(1)) 'i
loti(2) = r1.Item(trocas2ot (1) + 1) 'i+1
lotj(1) = r1.Item(trocas2ot (2)) 'j
lotj(2) = r1.Item(trocas2ot (2) + 1) 'j+1

custo = matriz_distâncias (loti(1) + 1, loti(2) + 1) + matriz_distâncias
(lotj(1) + 1, lotj(2) + 1) - (matriz_distâncias (loti(1) + 1, loptj (1) +
1) + matriz_distâncias (loti(2) + 1, lotj(2) + 1)) 'Diferença na distância
de i,i+1 e j,j+1 para i,j e i+1,j+1

If custo > 0 Then 'Se a distância é maior no primeiro, a troca acontece

```

```

r1.Remove (trocas2ot(1) + 1)
r1.Add lotj(1), Before:= trocas2ot(1) + 1
r1.Remove (trocas2ot(2))
r1.Add loti(2), Before:= trocas2ot(2)
dist_rotal = dist_rotal - custo
End If
    
```

Figura 11: Secção do código das trocas 2-otimais

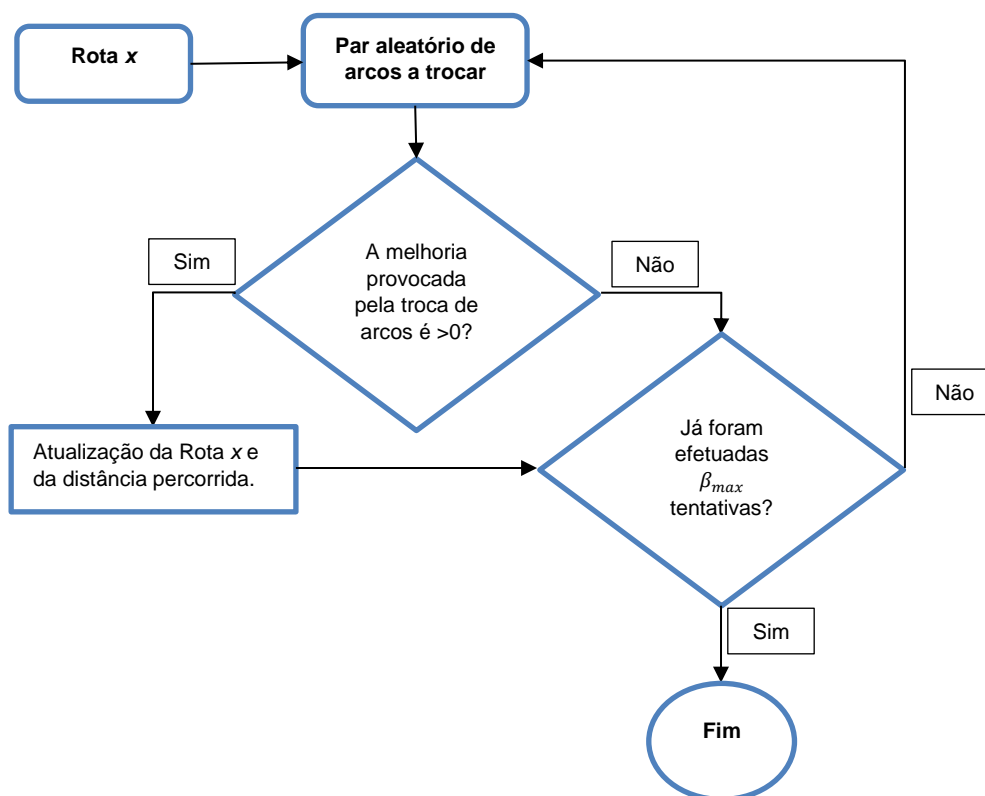


Figura 12: Fluxograma das trocas 2-otimais

O fluxograma da Figura 12 resume o processo das trocas 2-otimais, já com as rotas atualizadas após as trocas entre elas. O código na Figura 11 pertence ao processo de troca, e tem como base as posições selecionadas aleatoriamente na rota. Posteriormente, o vizinho resultante dos passos anteriormente expostos é guardado na vizinhança. O último passo do algoritmo seleciona, de entre o conjunto de vizinhos guardados, aquele que irá permitir obter a menor distância total, mesmo que seja superior à distância que serviu de base para o *Tabu Search*, tal como é possível observar pelo código na Figura 13.

```

dist = 1000000000
i = 1
vizinho = 0
Do While distânciatotal_vizinhos(i) <> "" 'Seleção da menor distância
percorrida entre os vizinhos
    If dist > distânciatotal_vizinhos(i) Then
        dist = distânciatotal_vizinhos(i)
        vizinho = i
    End If
    i = i + 1
Loop
'Guardar o melhor vizinho e respetivas distâncias percorridas
dist1 = distânciatotal_vizinhosr1(vizinho)
dist2 = distânciatotal_vizinhosr2 (vizinho)
dist3 = distânciatotal_vizinhosr2 (vizinho)
i = 1
Do While vizinhançar1(i, vizinho) <> ""
ReDim Preserve r1(1 To i)
    r1(i) = vizinhançar1(i, vizinho)
    i = i + 1
Loop
i = 1
Do While vizinhançar2(i, vizinho) <> ""
ReDim Preserve r2(1 To i)
    r2(i) = vizinhançar2(i, vizinho)
    i = i + 1
Loop
i = 1
Do While vizinhançar3(i, vizinho) <> ""
ReDim Preserve r3(1 To i)
    r3(i) = vizinhançar3(i, vizinho)
    i = i + 1
Loop

```

Figura 13: Secção de Código para a seleção do melhor vizinho na vizinhança

A *Lista Tabu*, que armazena as novas ligações, comparativamente à solução da iteração anterior, não foi integrada no código, tendo sido construída, iteração a iteração, como se pode observar nas Figuras presentes no Anexo ii.. Consideram-se *tabu* o conjunto de movimentos realizados em cada iteração, e não apenas cada par de arcos adicionado numa iteração. O facto de este processo ser efetuado de forma incremental, permitiu identificar erros de programação, em fases iniciais, e perceber o impacto que as trocas estavam a ter no processo.

A seleção dos parâmetros usados nos diversos testes efetuados à meta-heurística foi feita de forma a procurar um equilíbrio entre a qualidade da pesquisa e a capacidade de processamento da máquina utilizada. Assim, mantendo constantes os seguintes parâmetros,

N_{max}	5 iterações
β_{max}	10 trocas
θ_{min}	4 iterações
θ_{max}	4 iterações

Tabela II: Parâmetros fixos ao longo dos testes da meta-heurística Tabu Search

foram efetuados os testes que se seguem:

	α_{max}	M, P
Teste 1	20	10,10
Teste 2	20	5, 5
Teste 3	30	5, 5
Teste 4	50	10,10
Teste 5	50	5, 5
Teste 6	20	2, 2
Teste 7	50	2, 2
Teste 8	50	1, 1
Teste 9	100	1, 1
Teste 10	150	1, 1
Teste 11	150	1, 1

Tabela III: Parâmetros variáveis ao longo dos testes efetuados

4.2.3. Microsoft Excel VRP Spreadsheet Solver

O *Microsoft Excel VRP Spreadsheet Solver* tem por base as coordenadas de todos os pontos da rede, bem como horários de trabalho, veículos disponíveis e respetiva velocidade média. Com localizações em Portugal existem ainda algumas limitações na ferramenta, que calcula apenas distâncias Euclidianas e não por estrada. Assim, este passo foi alterado, de forma a utilizar as distâncias obtidas no *Google Script*.

A geração de um conjunto de rotas nesta ferramenta inicia-se com o algoritmo de *Savings*, referido na Metodologia, para a criação da solução inicial, e de seguida é aplicada uma adaptação da meta-heurística *Adaptive Large Neighborhood Search* (Erdoğan, 2017).

Foram feitos três testes com base nesta ferramenta: o primeiro mantendo as três linhas da rede, de forma a avaliar a qualidade dos resultados obtidos, o segundo seguindo a sugestão da TiS (2020) e cobrindo todas as paragens com apenas duas rotas, e o terceiro voltando às três rotas e removendo as 15 paragens extra, de forma a perceber o impacto que estas tiveram nos resultados.

Nestas implementações, considerou-se que o limite máximo aceitável para percorrer cada linha de início ao fim seria 1h30min, que a hora de início de trabalho é às 7h45, tal como acontece atualmente no município de Bragança, e que os veículos se deslocam a uma velocidade média de 30km/h. O algoritmo foi corrido durante 12 minutos para os dois primeiros testes e 8 minutos para o terceiro, respeitando o tempo recomendado pela aplicação, tendo em conta a extensão do problema.

Capítulo 5. Resultados

4.1. Resultados do Algoritmo *Cluster-First / Route-Second*

O algoritmo selecionado para a construção das rotas iniciais gerou o seguinte conjunto, que percorre um total de 47 187 metros:

Rota 1

1 – 2 – 29 – 30 – 3 – 4 – 5 – 28 – 6 – 24 – 27 – 26 – 25 – 23 – 7 – 8 – 11 – 10 –
14 – 12 – 13 – 15 – 16 – 17 – 18 – 19 – 20 – 21 – 22 – 9 – 35 – 34 – 32 – 31 –
33 – 1,

com distância percorrida de 16 421 metros;

Rota 2

1 – 38 – 40 – 51 – 52 – 53 – 54 – 55 – 59 – 61 – 63 – 62 – 64 – 60 – 65 – 67 –
69 – 68 – 66 – 58 – 57 – 56 – 50 – 46 – 47 – 48 – 45 – 49 – 44 – 43 – 42 – 41 –
39 – 36 – 37 – 70 – 1,

com distância percorrida de 15 463 metros;

Rota 3

1 – 73 – 74 – 75 – 76 – 80 – 79 – 78 – 77 – 81 – 72 – 83 – 71 – 82 – 84 – 85 –
87 – 89 – 90 – 91 – 93 – 94 – 96 – 97 – 98 – 100 – 101 – 103 – 105 – 104 – 102
– 95 – 99 – 92 – 88 – 86 – 1,

com distância percorrida de 15 303 metros.

Não era esperado que o algoritmo selecionado gerasse um bom resultado. A esta solução será aplicada a meta-heurística *Tabu Search*, para procurar uma solução melhor que o conjunto de rotas atual.

4.2. Resultados da Meta-heurística *Tabu Search*

Foi aplicada a meta-heurística, inicializada com a solução apresentada em 5.1., e com cada um dos conjuntos de parâmetros definidos nas Tabelas II e III. De

seguida apresenta-se o resultado obtido em cada teste. Os resultados da totalidade das iterações de cada teste encontram-se no Anexo ii.

	Número de iterações	Solução da melhor iteração do teste (m)	Melhor solução encontrada até ao momento (m)
Teste 1	5	54 574	47 187
Teste 2	5	51 890	47 187
Teste 3	5	52 564	47 187
Teste 4	5	54 645	47 187
Teste 5	5	52 717	47 187
Teste 6	5	47 323	47 187
Teste 7	5	49 811	47 187
Teste 8	12	45 814	45 814
Teste 9	16	43 715	43 715
Teste 10	16	45 356	43 715
Teste 11	7	44 710	43 715

Tabela IV: *Melhor resultado obtido em cada teste efetuado*

Como é possível observar na tabela anterior, a melhor solução foi obtida no Teste 9, embora nenhuma delas permita melhorar as rotas atuais da cidade de Bragança.

De seguida apresenta-se, a título de exemplo, a melhor solução obtida nos testes efetuados e que permite percorrer todas as paragens num total de 43 715 metros:

Rota 1

1 – 71 – 3 – 4 – 5 – 28 – 6 – 24 – 27 – 26 – 25 – 23 – 12 – 14 – 11 – 10 – 8 – 7 – 13 – 15 – 16 – 17 – 18 – 19 – 20 – 21 – 22 – 9 – 62 – 64 – 66 – 58 – 36 – 35 – 32 – 1,

com distância percorrida de 14 386 metros;

Rota 2

1 – 2 – 30 – 73 – 74 – 75 – 76 – 80 – 79 – 78 – 77 – 81 – 72 – 83 – 82 – 84 – 85 – 87 – 89 – 90 – 91 – 93 – 94 – 96 – 97 – 98 – 100 – 101 – 103 – 104 – 102 – 95 – 99 – 92 – 88 – 86 – 1,

com distância percorrida de 14 341 metros;

Rota 3

1 – 31 – 33 – 105 – 38 – 40 – 51 – 52 – 53 – 54 – 56 – 59 – 61 – 63 – 60 – 65 –
67 – 68 – 57 – 55 – 50 – 46 – 47 – 48 – 45 – 49 – 44 – 43 – 42 – 41 – 39 – 37 –
34 – 69 – 29 – 70 – 1,

com distância percorrida de 14 988 metros.

Na Figura 14 é possível observar como se procedeu à construção da *Lista Tabu* em cada iteração e como se foi avaliando se os movimentos efetuados eram movimentos *Tabu* (as ligações a vermelho nas listas representam ligações que saíram das rotas e que fazem parte de movimentos *tabu*). No final de cada iteração do *Tabu Search* a solução foi analisada e os movimentos que a originaram, relativamente à solução da iteração anterior, foram manualmente inseridos na Lista Tabu. Nas quatro iterações seguintes, aquando da análise da solução obtida, passa a ser necessário garantir que este conjunto de movimentos continua a estar na solução.

Rota Inicial		2	29	30	3	4	5	28	6	24	27	26	25	23	7	8	11	10	14	12	13	15	16	17	18	19	20	21	22	9	35	34	32	31	33	1	70	1		
rota1	1	38	40	51	52	53	54	55	59	61	63	62	64	60	65	67	69	68	66	58	57	56	50	46	47	48	45	48	49	44	43	42	41	39	36	37	29	70	1	
rota2	1	38	40	51	52	53	54	55	59	61	63	62	64	60	65	67	69	68	66	58	57	56	50	46	47	48	45	48	49	44	43	42	41	39	36	37	29	70	1	
rota3	1	73	74	75	76	80	79	78	77	81	72	83	71	82	84	85	87	89	90	91	93	94	96	97	98	100	101	103	105	104	102	95	99	92	88	86	86	1	1	
distância		47	187																																					
		16	421																																					
		15	463																																					
		15	303																																					
Iteração 1		38	40	51	52	53	54	55	59	61	63	62	64	60	65	67	69	68	66	57	56	50	46	47	48	45	48	49	44	43	42	41	39	36	37	29	70	1		
rota1	1	38	40	51	52	53	54	55	59	61	63	62	64	60	65	67	69	68	66	57	56	50	46	47	48	45	48	49	44	43	42	41	39	36	37	29	70	1		
rota2	1	2	30	3	4	5	28	6	24	27	26	25	23	7	8	11	10	14	12	13	15	16	17	18	19	20	21	22	9	58	35	34	32	31	33	1	1	1		
rota3	1	73	74	75	76	80	79	78	77	81	72	83	71	82	84	85	87	89	90	91	93	94	96	97	98	100	101	103	105	104	102	95	99	92	88	86	86	1	1	
distância		47	186																																					
		16	106																																					
		15	777																																					
		15	303																																					
Lista Tabu		[[66,57],[37,29],[29,70],[2,30],[9,58],[58,35]]																																						
Iteração 2		38	40	51	52	53	54	55	59	61	63	62	64	60	65	67	69	68	66	57	56	50	46	47	48	45	48	49	44	43	42	41	39	36	37	28	29	70	1	
rota1	1	38	40	51	52	53	54	55	59	61	63	62	64	60	65	67	69	68	66	57	56	50	46	47	48	45	48	49	44	43	42	41	39	36	37	28	29	70	1	
rota2	1	2	30	69	3	4	5	6	24	27	26	25	23	7	8	11	10	14	12	13	15	16	17	18	19	20	21	22	9	58	35	34	32	31	33	1	1	1		
rota3	1	73	74	75	76	80	79	78	77	81	72	83	71	82	84	85	87	89	90	91	93	94	96	97	98	100	101	103	105	104	102	95	99	92	88	86	86	1	1	
distância		47	601																																					
		16	190																																					
		16	108																																					
		15	303																																					
Lista Tabu		[[66,57],[37,29],[29,70],[2,30],[9,58],[58,35]];																																						
Iteração 11		71	3	4	5	28	6	24	27	26	25	23	12	14	11	10	8	7	13	15	16	17	18	19	20	21	22	9	62	64	66	58	36	35	32	1	1	1		
rota1	1	71	3	4	5	28	6	24	27	26	25	23	12	14	11	10	8	7	13	15	16	17	18	19	20	21	22	9	62	64	66	58	36	35	32	1	1	1		
rota2	1	2	30	73	74	75	76	80	79	78	77	81	72	83	82	84	85	87	89	90	91	93	94	96	97	98	100	101	103	104	102	95	99	92	88	86	86	1	1	
rota3	1	31	33	105	38	40	51	52	53	54	56	59	61	63	60	65	67	68	57	55	50	46	47	48	45	49	44	43	42	41	39	37	34	69	29	70	1	1		
distância		43	715																																					
		14	386																																					
		14	341																																					
Lista Tabu		[[63,64],[37,34],[34,29],[9,62],[62,66],[35,32]];																																						
Iteração 15		31	33	105	38	40	51	52	53	54	56	59	61	63	60	65	67	68	57	55	50	46	47	48	45	49	44	43	42	41	39	37	34	29	82	85	1	1		
rota1	1	31	33	105	38	40	51	52	53	54	56	59	61	63	60	65	67	68	57	55	50	46	47	48	45	49	44	43	42	41	39	37	34	29	82	85	1	1		
rota2	1	2	69	30	70	72	74	75	76	80	79	78	77	81	81	72	83	84	87	89	90	91	93	94	96	97	98	100	101	103	104	102	95	99	92	88	86	1	1	
rota3	1	71	3	4	5	28	6	24	27	26	25	23	12	14	11	10	8	7	13	15	16	17	18	19	20	21	22	9	62	64	66	58	36	35	32	1	1	1		
distância		46	666																																					
		16	745																																					
		14	935																																					
Lista Tabu		[[29,82],[82,11],[30,70],[70,73],[83,84]];																																						
Iteração 16		2	69	30	70	74	75	76	80	79	78	77	81	72	83	84	85	87	89	90	91	93	94	96	97	98	100	101	103	104	102	95	99	92	88	86	1	1		
rota1	1	2	69	30	70	74	75	76	80	79	78	77	81	72	83	84	85	87	89	90	91	93	94	96	97	98	100	101	103	104	102	95	99	92	88	86	1	1		
rota2	1	31	33	105	38	40	51	52	53	54	56	59	61	63	60	65	67	68	57	55	50	46	47	48	45	49	44	43	42	41	39	37	34	29	73	82	1	1		
rota3	1	71	3	4	5	28	6	24	27	26	25	23	12	14	11	10	8	7	13	15	16	17	18	19	20	21	22	9	62	64	66	58	36	35	32	1	1	1		
distância		45	381																																					
		14	979																																					
		16	016																																					
		14	386																																					

5 iterações sem qualquer melhoria

Figura 14: Seleção dos resultados de iterações do Teste 9

4.3. Microsoft Excel VRP Spreadsheet Solver

De seguida, apresentam-se os resultados obtidos no primeiro teste efetuado no Microsoft Excel VRP Spreadsheet Solver.

Rota 1

1 – 31 – 33 – 105 – 104 – 103 – 102 – 101 – 100 – 99 – 92 – 88 – 86 – 85 – 87 – 89 – 90 – 91 – 93 – 94 – 95 – 96 – 97 – 98 – 1,

com distância percorrida de 8 043 metros;

Rota 2

1 – 57 – 61 – 63 – 62 – 64 – 60 – 65 – 66 – 58 – 59 – 15 – 17 – 18 – 19 – 16 – 20 – 14 – 13 – 12 – 11 – 10 – 21 – 22 – 9 – 8 – 7 – 6 – 23 – 24 – 25 – 26 – 27 – 28 – 67 – 68 – 69 – 1,

com distância percorrida de 16 187 metros;

Rota 3

1 – 38 – 40 – 42 – 43 – 44 – 45 – 46 – 47 – 48 – 49 – 50 – 51 – 52 – 53 – 54 – 55 – 56 – 41 – 39 – 37 – 36 – 35 – 34 – 32 – 2 – 3 – 4 – 5 – 29 – 73 – 74 – 75 – 76 – 82 – 84 – 83 – 72 – 70 – 71 – 80 – 79 – 78 – 77 – 81 – 30 – 1,

com distância percorrida de 16 953 metros.

O resultado obtido por esta combinação (total de 41 183 metros) é mais vantajoso do que qualquer um dos obtidos nos testes efetuados com os algoritmos selecionados, embora também não melhore as atuais rotas de transportes públicos da cidade de Bragança (em que se percorre um total de 34 896 metros).

Passando aos resultados do segundo teste, que gera apenas duas rotas:

Rota 1

1 – 31 – 33 – 104 – 103 – 101 – 102 – 95 – 96 – 99 – 92 – 93 – 94 – 97 – 98 – 100 – 35 – 34 – 32 – 2 – 3 – 4 – 5 – 29 – 70 – 71 – 82 – 83 – 72 – 73 – 74 – 75 – 76 – 80 – 79 – 78 – 77 – 81 – 84 – 85 – 87 – 89 – 90 – 91 – 88 – 86 – 1,

com distância percorrida de 16 869 metros;

Rota 2

1 – 69 – 68 – 66 – 58 – 50 – 47 – 48 – 46 – 45 – 49 – 44 – 43 – 42 – 41 – 39 – 37 – 36 – 105 – 38 – 40 – 51 – 52 – 53 – 54 – 55 – 56 – 57 – 65 – 67 – 6 – 62 –

64 – 60 – 59 – 61 – 63 – 15 – 17 – 18 – 19 – 16 – 20 – 14 – 13 – 12 – 11 – 10 –
21 – 22 – 9 – 8 – 7 – 23 – 24 – 25 – 26 – 27 – 28 – 30 – 1,

com distância percorrida de 23 663 metros.

A solução deste segundo teste tem um valor total de 40 532 metros mas esta não é admissível uma vez que uma das rotas obtida leva quase duas horas a ser percorrida, o que não cumpre a restrição de tempo dada ao problema.

No terceiro teste, removendo as 15 paragens adicionadas para o estudo, a solução obtida permite percorrer o conjunto das paragens num total de 38 694 metros. Além disto, embora a indicação dada seja de que se pretendem construir três rotas, a aplicação devolve apenas duas.

Os resultados destes três testes também se encontram no Anexo ii.

Conclusões

Este trabalho teve como objetivo a otimização de rotas da rede de transportes públicos do concelho de Bragança, com especial foco na zona urbana. Para tal, foram implementadas em VBA uma heurística construtiva *Clusters – First / Route – Second* e uma meta-heurística melhorativa *Tabu Search* para encontrar um conjunto de três rotas que permitisse percorrer a totalidade das paragens consideradas em menos tempo do que o das rotas atuais. Posteriormente, para validar e avaliar os resultados obtidos e a escolha de manter três linhas e adicionar as paragens extra, foram feitos três testes no *Microsoft Excel VRP Spreadsheet Solver*.

Com base nos resultados apresentados, é possível retirar três pontos essenciais:

- Os resultados obtidos com as aplicações no *Microsoft Excel VRP Spreadsheet Solver* foram melhores do que os gerados pelos algoritmos selecionados para o estudo.
- A construção de apenas duas rotas que cubram as 105 paragens não permitiu obter uma solução admissível, mas removendo as 15 paragens adicionais, a solução obtida considera apenas duas rotas, sendo a melhor solução encontrada.
- Nenhuma das soluções encontradas permitiu otimizar o atual conjunto de rotas dos transportes públicos da cidade de Bragança.

Quanto ao primeiro ponto, a limitação provocada pela falta de capacidade de processamento pode explicar a não obtenção de melhores resultados. Embora, pelos resultados obtidos, o aumento de 100 para 150 vizinhos gerados em cada vizinhança não traga melhorias à busca, uma maior capacidade de processamento teria permitido efetuar outras combinações de parâmetros e efetuar testes de forma menos morosa. Também o número de iterações realizadas poderia ter sido superior com uma maior capacidade de processamento. Após análise apropriada, a própria implementação do algoritmo também poderá ser melhorada.

Outro ponto que se tornou mais evidente após serem efetuadas as primeiras duas aplicações na ferramenta *Microsoft Excel VRP Spreadsheet Solver*, e que

pode explicar não só o facto de a solução atual não ter sido melhorada, mas também o facto de o melhor resultado encontrado ser significativamente pior do que o atual, são as 15 paragens extra que foram adicionadas. Estas paragens foram adicionadas para evitar que, em alguns casos específicos, a linha fosse forçada a seguir um determinado sentido por uma longa extensão, uma vez que isto iria enviesar a procura. No entanto, trata-se de 15 paragens que não se encontram na rede atual, que tiveram de ser cobertas por alguma linha e cujas ligações, por muito curtas que sejam, terão impacto no resultado final. Daí a execução do terceiro teste nesta ferramenta que permitiu, de facto, atingir a melhor solução encontrada, apesar de continuar a percorrer uma distância superior à das rotas atuais.

Assim, sendo claro que a tentativa de melhorar as rotas de transportes públicos no concelho de Bragança não foi bem-sucedida neste projeto, e com o último teste em *Microsoft Excel VRP Spreadsheet Solver* que demonstra a qualidade da solução atual, outro fator relevante de ser estudado é a frequência das rotas. Não deixa de ser curiosa a realidade de haver uma média de seis pessoas por circuito, existindo sobreocupação em duas alturas do dia. Um estudo junto da população, que permita perceber as razões que levam os utentes a não optar pelos transportes públicos, também deverá ser interessante.

Referências Bibliográficas

- Afandizadeh, S., Khaksar, H., & Kalantari, N. (2013). Bus fleet optimization using genetic algorithm a case study of Mashhad. *International Journal of Civil Engineering*, 11(1), 43–52.
- Arbex, R. O., & Cunha, C. B. (2015). Efficient transit network design and frequencies setting multi-objective optimization by alternating objective genetic algorithm. *Transportation Research Part B: Methodological*, 81, 355–376. <https://doi.org/10.1016/j.trb.2015.06.014>
- 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)
- Battiti, R., & Tecchiolli, G. (1994). Simulated annealing and Tabu search in the long run: A comparison on QAP tasks. *Computers and Mathematics with Applications*, 28(6), 1–8. [https://doi.org/10.1016/0898-1221\(94\)00147-2](https://doi.org/10.1016/0898-1221(94)00147-2)
- Beasley, J. E. (1983). Route first—cluster second methods for vehicle routing. *Omega - The International Journal of Management Science* volume 11, 403-408.
- Bento, R., Ribeiro, P., Martins, L., & Meirles, T. (2015). *Estudo de Mobilidade e Transportes das Terras de Trás-os-Montes*, CIM-TTM https://www.cim-ttm.pt/cimttm/uploads/document/file/99/estudo_de_mobilidade_e_transportes_da_comunidade_intermunicipal_das_terras_de_tras_os_montes.pdf.
- Burkhardt, J. E., Hedrick, J. L., & McGavock, A. T. (1998). *TCRP Report 34 Assessment of the Economic Impacts of Rural Public Transportation*, 1^a Ed. Washington, D.C.: National Academy Press.
- Câmara Municipal de Bragança. (2010). *STUB - Serviço de Transportes Urbanos de Bragança*, 1^a Ed. Bragança: Câmara Municipal de Bragança..
- Carpinteiro, A. S. (2007). *Transportes Públicos em Áreas Rurais - Perspectivas e Estudos-Caso*. VI Congresso da Geografia Portuguesa, Lisboa, 1-17.
- Clarke, G., & Wright, J. W. (1964). Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Operations Research*, 12(4), 568–581. <https://doi.org/10.1287/opre.12.4.568>
- Deitch, R., & Ladany, S. P. (2000). The Bus Touring Problem: Cluster - first, route - second heuristic. *Tourism Economics*, 6(3), 263–279.

<https://doi.org/10.5367/000000000101297631>

- Erdoğan, G. (2017). An open source Spreadsheet Solver for Vehicle Routing Problems. *Computers & Operations Research*, 84, 62–72.
<https://doi.org/10.1016/j.cor.2017.02.022>
- Fan, W., & Machemehl, R. (2006). Optimal Transit Route Network Design Problem with Variable Transit Demand: Genetic Algorithm Approach. *Journal of Transportation Engineering*, 132, 40–51.
[https://doi.org/10.1061/\(ASCE\)0733-947X\(2006\)132:1\(40\)](https://doi.org/10.1061/(ASCE)0733-947X(2006)132:1(40))
- Fan, W., & Machemehl, R. (2008). Tabu search strategies for the public transportation network optimizations with variable transit demand. *Computer-Aided Civil and Infrastructure Engineering*, 23, 502–520.
- Fisher, M. (1995). Chapter 1 Vehicle routing. *Handbooks in Operations Research and Management Science*, 8, 1–33.
[https://doi.org/10.1016/S0927-0507\(05\)80105-7](https://doi.org/10.1016/S0927-0507(05)80105-7)
- García-Archilla, B., Lozano, A. J., Mesa, J. A., & Perea, F. (2013). GRASP algorithms for the robust railway network design problem. *Journal of Heuristics*, 19(2), 399–422. <https://doi.org/10.1007/s10732-011-9185-z>
- Guihaire, V., & Hao, J. K. (2008). Transit network design and scheduling: A global review. *Transportation Research Part A: Policy and Practice*, 42(10), 1251–1273. <https://doi.org/10.1016/j.tra.2008.03.011>
- Jha, S. B., Jha, J. K., & Tiwari, M. K. (2019). A multi-objective meta-heuristic approach for transit network design and frequency setting problem in a bus transit system. *Computers and Industrial Engineering*, 130(February), 166–186. <https://doi.org/10.1016/j.cie.2019.02.025>
- Kepaptsoglou, K. & Karlaftis, M. (2009). Transit Route Network Design Problem : Review. *Journal of Transportation Engineering*, 135:(8), 174–182.
- Kuo, Y. (2014). *Design method using hybrid of line-type and circular-type routes for transit network system optimization*, Top 22, 600-613
<https://doi.org/10.1007/s11750-013-0273-0>
- Moseley, Malcom J. 2003. *Rural Development: Principles and Practice*, 3^aEd. London: SAGE Publications Ltd, 99-218.
- Social Exclusion Unit. (2003). Making the Connections: Final Report on

- Transport and Social Exclusion. In *Office of the Deputy Prime Minister* (Vol. 100, Issue 6). <https://doi.org/10.1002/j.1551-8833.2008.tb09646.x>
- Sousa, J., Ferreira, H., & Fernandes, A. (2009). *Transportes Colectivos em Espaço Rural e Áreas de Baixa Densidade Urbana: Abordagens e Soluções*, 1º Congresso de Desenvolvimento Regional de Cabo Verde, 2562-2588
- Sultana, T., Akhand, M. A. H., & Rahman, M. M. H. (2017). A variant Fisher and Jaikumar algorithm to solve capacitated vehicle routing problem. *ICIT 2017 - 8th International Conference on Information Technology, Proceedings, May*, 710–716. <https://doi.org/10.1109/ICITECH.2017.8079932>
- TiS. (2020). *Bragança: Uma nova energia para a mobilidade sustentável - Propostas de Intervenção*.
- Yen, J. Y. (1971). Finding the K Shortest Loopless Paths in a Network . *Management Science*, 17(11), 712–716. <https://doi.org/10.1287/mnsc.17.11.712>

Anexos

i. Tratamento de Dados



TRABALHO FINAL DE MESTRADO

OTIMIZAÇÃO DA REDE DE TRANSPORTES PÚBLICOS DO CONCELHO DE BRAGANÇA COM IMPLEMENTAÇÃO EM VBA

Dados Necessários:

1. Autocarros da frota que servem as linhas urbanas;
 - a. Capacidade (em lugares sentados e lugares em pé) destes autocarros.
2. Motoristas contratados para estas rotas¹.
3. Custos
 - a. Salários dos motoristas referidos no ponto 2.;
 - b. Gastos administrativos;
 - c. Gastos de manutenção e limpeza da frota;
 - d. Gasto/km de combustível (média) de cada autocarro da frota.
4. Mapa das linhas atuais;
 - a. Lista da localização exata de todas as paragens destas linhas (coordenadas).
5. Mapa das estradas da cidade (atualizado com o resultado dos trabalhos que estão a ser levados a cabo em algumas zonas).
6. Procura registada por dia e por estação durante um ano².

Peço ainda que me enviem estudos e relatórios recentes, que me permitam perceber melhor a realidade do STUB, como:

- Frequência nas linhas urbanas;
- Tempo médio que demora a percorrer cada uma das linhas;
- Evolução das linhas e especificidades a ter em conta.

¹ Os nomes dos motoristas não serão usados em momento algum deste projeto e, se assim o entenderem, podem mesmo substituir os nomes por letras ou códigos numéricos antes de os enviar.

² É preferível que o ano considerado não abranja meses de 2020, visto que não serão valores representativos da realidade de um ano "normal".

Figura 15: Pedido de disponibilização de dados à Câmara Municipal de Bragança

• Teste 2

5 iterações sem qualquer melhoria relativamente à rota inicial

Rota Inicial		1	2	29	30	3	4	5	28	6	24	27	26	25	23	7	8	11	10	14	12	13	15	16	17	18	19	20	21	22	9	35	34	32	31	33	1		
rota1	1	38	40	51	52	53	54	59	61	63	62	64	60	65	67	69	68	66	66	58	57	56	50	46	47	48	49	44	43	42	41	39	37	84	83	72	70	1	
rota2	1	73	74	75	76	80	79	78	77	81	71	82	85	87	89	90	91	93	94	96	97	95	101	103	38	45	55	36	105	104	102	100	99	92	88	86	1		
rota3	1	74	75	76	80	79	78	77	81	71	82	85	87	89	90	91	93	94	96	97	95	101	103	38	45	55	36	105	104	102	100	99	92	88	86	1			
distancia		47	107	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421

Iteração 1		1	94	98	40	51	52	53	54	59	61	63	62	64	60	65	67	69	68	66	58	57	56	50	46	47	48	49	44	43	42	41	39	37	84	83	72	70	1
rota1	1	73	74	75	76	80	79	78	77	81	71	82	85	87	89	90	91	93	94	96	97	95	101	103	38	45	55	36	105	104	102	100	99	92	88	86	1		
rota2	1	73	74	75	76	80	79	78	77	81	71	82	85	87	89	90	91	93	94	96	97	95	101	103	38	45	55	36	105	104	102	100	99	92	88	86	1		
rota3	1	73	74	75	76	80	79	78	77	81	71	82	85	87	89	90	91	93	94	96	97	95	101	103	38	45	55	36	105	104	102	100	99	92	88	86	1		
distancia		51	890	17	846	17	823	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421

Iteração 2		1	73	74	75	76	80	78	77	81	71	82	85	87	89	90	91	93	94	96	97	101	103	38	46	47	49	45	42	37	36	104	102	99	92	88	86	1	
rota1	1	95	96	98	100	105	52	53	54	55	59	61	63	62	64	60	65	67	69	68	66	58	57	56	50	46	47	48	45	44	43	41	39	84	83	79	72	70	1
rota2	1	95	96	98	100	105	52	53	54	55	59	61	63	62	64	60	65	67	69	68	66	58	57	56	50	46	47	48	45	44	43	41	39	84	83	79	72	70	1
rota3	1	2	29	30	3	4	5	28	6	24	27	26	25	23	7	8	11	10	14	12	13	15	16	17	18	19	20	21	22	9	35	34	32	31	33	1			
distancia		54	802	19	442	19	418	19	418	19	418	19	418	19	418	19	418	19	418	19	418	19	418	19	418	19	418	19	418	19	418	19	418	19	418	19	418	19	418

Iteração 3		1	72	73	74	75	76	80	77	81	71	85	87	89	90	91	93	94	96	97	101	103	38	56	56	48	45	42	37	36	104	102	99	92	88	86	1		
rota1	1	95	98	100	105	51	52	53	54	55	59	61	63	62	64	60	65	67	69	68	66	58	57	50	40	46	47	49	44	43	41	39	84	83	79	72	70	1	
rota2	1	95	98	100	105	51	52	53	54	55	59	61	63	62	64	60	65	67	69	68	66	58	57	50	40	46	47	49	44	43	41	39	84	83	79	72	70	1	
rota3	1	2	29	30	3	4	5	28	6	24	27	26	25	23	7	8	11	10	14	12	13	15	16	17	18	19	20	21	22	9	35	34	32	31	33	1			
distancia		58	101	19	884	21	956	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421

Iteração 4		1	72	73	74	75	76	80	77	81	71	85	87	89	90	91	93	94	96	97	98	101	103	40	56	50	45	47	42	39	36	104	102	99	92	88	86	1	
rota1	1	95	100	105	38	51	52	53	54	55	59	61	63	62	64	60	65	67	69	68	66	58	57	46	46	48	48	44	43	41	37	84	83	79	72	70	1		
rota2	1	95	100	105	38	51	52	53	54	55	59	61	63	62	64	60	65	67	69	68	66	58	57	46	46	48	48	44	43	41	37	84	83	79	72	70	1		
rota3	1	2	29	30	3	4	5	28	6	24	27	26	25	23	7	8	11	10	14	12	13	15	16	17	18	19	20	21	22	9	35	34	32	31	33	1			
distancia		56	988	19	709	20	488	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421	16	421

Iteração 5		1	72	73	75	76	80	79	78	77	81	71	84	85	87	89	90	91	93	94	96	97	98	101	103	40	50	45	51	58	42	39	36	104	102	100	92	88	86	1	
rota1	1	99	95	96	103	105	38	52	53	54	55	59	61	63	62	64	60	65	67	69	68	66	57	56	50	46	47	48	45	49	44	43	41	37	84	83	83	83	70	1	
rota2	1	99	95	96	103	105	38	52	53	54	55	59	61	63	62	64	60	65	67	69	68	66	57	56	50	46	47	48	45	49	44	43	41	37	84	83	83	83	70	1	
rota3	1	2	29	30	3	4	5	28	6	24	27	26	25	23	7	8	11	10	14	12	13	15	16	17	18	19	20	21	22	9	35	34	34	34	32	31	33	1			
distancia		55	988	20	808	19	916	19	916	19	916	19	916	19	916	19	916	19	916	19	916	19	916	19	916	19	916	19	916	19	916	19	916	19	916	19	916	19	916	19	916

Figura 18: Resultados das iterações do Teste 2

• Teste 3

5 iterações sem qualquer melhoria relativamente à rota inicial

Rota Inicial	
rota1	1 2 29 30 3 4 5 28 6 24 27 26 25 23 7 8 11 10 14 12 13 15 16 17 18 19 20 21 22 9 62 60 42 35 34 31 1
rota2	1 38 40 51 52 53 54 55 59 61 63 62 64 60 65 67 69 68 66 58 57 50 46 47 48 45 49 44 43 42 41 39 36 37 32 70 1
rota3	1 73 74 75 76 80 79 78 77 81 72 83 71 82 84 85 87 89 90 91 93 94 96 97 98 100 101 103 105 104 102 95 99 92 88 86 1
distância	47 187
rota1	16 421
rota2	16 421
rota3	16 421
distância	15 303

Iteração 1	
rota1	1 38 40 51 52 53 54 55 63 18 24 26 64 60 65 67 69 68 66 58 57 50 46 47 48 45 49 44 43 42 41 39 36 37 32 70 1
rota2	1 2 29 30 3 4 5 28 6 27 25 23 7 8 59 61 11 10 14 12 13 15 16 17 18 19 20 21 22 9 62 56 35 34 31 33 1
rota3	1 73 74 75 76 80 79 78 77 81 72 83 71 82 84 85 87 89 90 91 93 94 96 97 98 100 101 103 105 104 102 95 99 92 88 86 1
distância	55 315
rota1	21 223
rota2	18 789
rota3	15 303

Iteração 2	
rota1	1 2 29 69 3 4 5 28 6 27 25 23 7 8 50 59 11 10 12 13 15 16 17 18 19 20 21 22 9 62 60 42 37 35 34 31 1
rota2	1 33 38 40 51 52 53 54 55 61 63 14 24 26 64 65 67 68 66 58 57 56 46 47 48 45 49 44 43 42 41 39 36 37 32 70 30 1
rota3	1 73 74 75 76 80 79 78 77 81 72 83 71 82 84 85 87 89 90 91 93 94 96 97 98 100 101 103 105 104 102 95 99 92 88 86 1
distância	52 644
rota1	19 312
rota2	19 312
rota3	15 303

Iteração 3	
rota1	1 2 29 69 4 5 28 27 26 25 23 7 8 64 50 49 59 11 12 13 15 16 17 18 19 20 22 9 62 60 42 37 35 34 32 31 1
rota2	1 33 38 40 51 52 53 54 55 61 63 14 20 21 24 26 65 67 68 6 58 57 56 46 47 48 45 44 43 41 39 36 3 70 30 1
rota3	1 73 74 75 76 80 79 78 77 81 72 83 71 82 84 85 87 89 90 91 93 94 96 97 98 100 101 103 105 104 102 95 99 92 88 86 1
distância	54 906
rota1	19 929
rota2	19 274
rota3	15 303

Iteração 4	
rota1	1 33 40 51 52 53 54 55 59 61 14 12 10 24 66 65 68 58 57 56 49 46 47 48 45 44 43 41 39 37 36 69 3 5 70 30 1
rota2	1 2 29 4 28 27 26 25 23 6 7 8 64 67 50 63 11 13 15 16 17 18 19 20 21 22 9 62 60 42 38 37 34 35 32 31 1
rota3	1 73 74 75 76 80 79 78 77 81 72 83 71 82 84 85 87 89 90 91 93 94 96 97 98 100 101 103 105 104 102 95 99 92 88 86 1
distância	56 100
rota1	20 557
rota2	20 240
rota3	15 303

Iteração 5	
rota1	1 33 40 51 52 53 54 55 59 61 12 10 20 9 7 24 66 65 68 57 50 46 47 48 45 44 43 41 39 37 36 35 69 3 5 70 30 1
rota2	1 2 29 4 28 27 26 25 23 6 7 8 64 67 63 11 13 14 15 16 17 18 19 20 21 22 22 60 60 56 49 49 42 38 34 34 32 31 1
rota3	1 73 74 75 76 80 79 78 77 81 72 83 71 82 84 84 83 87 89 90 91 93 94 96 97 98 100 101 103 105 104 102 95 99 92 88 86 1
distância	55 052
rota1	20 388
rota2	19 388
rota3	15 303

Figura 19: Resultados das iterações do Teste 3

• Teste 7

Rota Inicial		Rota Final																																					
rota1	1	2	29	30	3	4	5	28	6	24	27	26	25	23	7	8	11	10	14	12	13	15	16	17	18	19	20	21	22	9	35	34	32	31	33	1			
rota2	1	38	40	51	52	53	54	55	59	61	63	62	64	60	65	67	69	68	66	65	57	56	50	46	47	48	45	49	44	43	42	41	39	36	37	70	1		
rota3	1	73	74	75	76	80	79	78	77	81	72	83	71	82	84	85	87	89	90	91	93	94	96	97	98	100	101	103	105	104	102	95	99	92	88	86	1		
distancia	47	187																																					
rota1	1	99	97	101	36	56	51	52	53	54	55	59	61	63	62	64	60	65	67	69	68	66	58	57	105	50	46	47	48	45	49	44	43	42	41	38	70	74	1
rota2	1	99	97	101	36	56	51	52	53	54	55	59	61	63	62	64	60	65	67	69	68	66	58	57	100	103	40	42	39	37	104	102	95	92	88	86	1		
rota3	1	70	73	74	75	76	79	78	77	81	72	82	84	85	87	89	90	91	93	94	96	97	98	100	103	40	43	42	39	104	102	95	92	88	86	1			
distancia	53	699																																					
rota1	1	2	29	30	3	4	5	28	6	24	27	26	25	23	7	8	11	10	14	12	13	15	16	17	18	19	20	21	22	9	35	34	32	31	33	1			
rota2	1	99	97	101	36	56	51	52	53	54	55	59	61	63	62	64	60	65	67	69	68	66	58	57	105	50	46	47	48	45	49	44	43	42	41	38	74	1	
rota3	1	70	73	75	76	80	79	78	77	81	72	82	84	85	87	89	90	91	93	94	96	97	98	100	103	40	43	42	39	104	102	95	92	88	86	1			
distancia	53	114																																					
rota1	1	99	97	101	36	56	51	52	53	54	55	59	61	63	62	64	60	65	67	69	68	66	58	57	105	50	46	47	48	45	49	44	43	42	41	38	74	1	
rota2	1	99	97	101	36	56	51	52	53	54	55	59	61	63	62	64	60	65	67	69	68	66	58	57	100	103	40	42	39	37	104	102	95	92	88	86	1		
rota3	1	70	73	75	76	80	79	78	77	81	72	82	84	85	87	89	90	91	93	94	96	97	98	100	103	40	43	42	39	104	102	95	92	88	86	1			
distancia	53	114																																					
rota1	1	73	74	75	76	80	79	78	77	81	72	83	71	82	84	85	87	89	90	91	93	94	96	97	98	100	103	40	42	105	104	102	95	92	88	86	1		
rota2	1	99	101	36	51	52	53	54	55	59	61	63	62	64	60	65	67	69	68	66	58	57	56	50	46	47	48	45	49	44	43	42	41	39	38	70	1		
rota3	1	2	29	30	3	4	5	28	6	24	27	26	25	23	7	8	11	10	14	12	13	15	16	17	18	19	20	21	22	9	35	34	32	31	33	1			
distancia	49	811																																					
rota1	1	73	74	75	76	80	79	78	77	81	72	83	71	82	84	85	87	89	90	91	93	94	96	97	98	100	103	40	42	105	104	102	95	92	88	86	1		
rota2	1	99	101	36	51	52	53	54	55	59	61	63	62	64	60	65	67	69	68	66	58	57	56	50	46	47	48	45	49	44	43	42	41	39	38	70	1		
rota3	1	2	29	30	3	4	5	28	6	24	27	26	25	23	7	8	11	10	14	12	13	15	16	17	18	19	20	21	22	9	35	34	32	31	33	1			
distancia	16	716																																					
rota1	1	99	97	101	36	56	51	52	53	54	55	59	61	63	62	64	60	65	67	69	68	66	58	57	105	50	46	47	48	45	49	44	43	42	41	38	74	1	
rota2	1	99	97	101	36	56	51	52	53	54	55	59	61	63	62	64	60	65	67	69	68	66	58	57	100	103	40	42	39	37	104	102	95	92	88	86	1		
rota3	1	70	73	75	76	80	79	78	77	81	72	82	84	85	87	89	90	91	93	94	96	97	98	100	103	40	43	42	39	104	102	95	92	88	86	1			
distancia	16	421																																					
rota1	1	99	97	101	36	56	51	52	53	54	55	59	61	63	62	64	60	65	67	69	68	66	58	57	105	50	46	47	48	45	49	44	43	42	41	38	74	1	
rota2	1	99	97	101	36	56	51	52	53	54	55	59	61	63	62	64	60	65	67	69	68	66	58	57	100	103	40	42	39	37	104	102	95	92	88	86	1		
rota3	1	70	73	75	76	80	79	78	77	81	72	82	84	85	87	89	90	91	93	94	96	97	98	100	103	40	43	42	39	104	102	95	92	88	86	1			
distancia	16	421																																					
rota1	1	99	97	101	36	56	51	52	53	54	55	59	61	63	62	64	60	65	67	69	68	66	58	57	105	50	46	47	48	45	49	44	43	42	41	38	74	1	
rota2	1	99	97	101	36	56	51	52	53	54	55	59	61	63	62	64	60	65	67	69	68	66	58	57	100	103	40	42	39	37	104	102	95	92	88	86	1		
rota3	1	70	73	75	76	80	79	78	77	81	72	82	84	85	87	89	90	91	93	94	96	97	98	100	103	40	43	42	39	104	102	95	92	88	86	1			
distancia	16	421																																					

5 iterações sem qualquer melhoria relativamente à rota inicial

Figura 23: Resultados das iterações do Teste 7

Iteração 11	
rota1	1 2 29 3 4 5 7 23 24 27 26 25 6 8 11 10 14 12 13 15 16 17 18 19 20 21 22 9 64 60 58 37 105 34 32 1
rota2	1 31 33 38 40 47 52 53 54 51 59 61 63 62 65 67 28 30 69 68 66 58 57 56 55 50 46 48 45 49 44 43 42 41 39 36 35 1
rota3	1 70 73 74 75 76 80 79 78 77 81 72 83 71 82 84 85 87 89 90 91 93 94 96 97 98 100 101 103 104 102 95 99 92 88 86 1
distância	47 468
	16 322
	16 805
	14 341
Letra Tabu	[(60,60),(46,57),(56,55),(5,50),(50,97),(42,39),(33,100),(101,11),(100,103),(104,41),(41,100)]; [(42,41),(41,39),(33,11),(100,101),(101,103),(104,102)]; [(63,62),(67,28),(28,30),(63,61),(63,62),(61,11),(11,30)]; [(69,60),(60,58),(52,11),(11,13),(13,55),(40,97),(47,57),(42,57),(42,55),(50,49),(46,48)];
Iteração 12	
rota1	1 2 29 3 4 5 28 7 23 24 27 26 25 6 8 11 10 14 12 13 15 16 17 18 19 20 21 22 9 64 60 58 37 105 34 32 1
rota2	1 31 33 38 40 47 52 53 54 51 59 61 63 62 65 67 30 69 68 66 58 57 56 55 50 46 48 45 49 44 43 42 41 39 36 35 1
rota3	1 70 73 74 75 76 80 79 78 77 81 72 83 71 82 84 85 87 89 90 91 93 94 96 97 98 100 101 103 104 102 95 99 92 88 86 1
distância	48 301
	16 427
	17 333
	14 341

5 iterações sem qualquer melhoria

Figura 26: Resultados das iterações do Teste 8 (parte III)

Iteração 10		1	30	3	4	5	28	6	24	27	26	25	23	7	8	11	10	14	12	13	15	16	17	18	19	20	21	22	9	62	64	66	58	36	35	32	86	1	
rota1	1	31	33	105	38	40	51	52	53	54	56	59	61	63	60	65	67	68	57	55	50	46	47	48	45	49	44	43	42	41	39	37	34	69	29	88	70	1	
rota2	1	2	73	74	75	76	80	79	78	77	81	72	83	82	84	85	87	89	90	91	93	94	96	97	98	100	101	103	104	102	95	99	92	88	86	1	1	1	
rota3	1	2	73	74	75	76	80	79	78	77	81	72	83	82	84	85	87	89	90	91	93	94	96	97	98	100	101	103	104	102	95	99	92	88	86	1	1	1	
distância	44	988	15	659	14	988	14	341																															
Lista Tabu		[(33,105),(105,38),(37,29),(1,2),(2,73),(103,104)]; [(63,64),(37,34),(34,29),(9,62),(62,66),(35,32)]; [(1,31),(31,33),(70,1),(1,30),(30,69),(32,1)]; [(30,3),(62,64),(64,66),(63,60),(34,69),(59,29)];																																					
Iteração 11		1	71	3	4	5	28	6	24	27	26	25	23	12	14	11	10	8	7	13	15	16	17	18	19	20	21	22	9	62	64	66	58	36	35	32	86	1	
rota1	1	2	30	73	74	75	76	80	79	78	77	81	72	83	82	84	85	87	89	90	91	93	94	96	97	98	100	101	103	104	102	95	99	92	88	86	1	1	
rota2	1	31	33	105	38	40	51	52	53	54	56	59	61	63	60	65	67	68	57	55	50	46	47	48	45	49	44	43	42	41	39	37	34	69	29	88	86	1	
rota3	1	31	33	105	38	40	51	52	53	54	56	59	61	63	60	65	67	68	57	55	50	46	47	48	45	49	44	43	42	41	39	37	34	69	29	88	86	1	
distância	43	715	14	386	14	341																																	
Lista Tabu		[(63,64),(37,34),(34,29),(9,62),(62,66),(35,32)]; [(1,31),(31,33),(70,1),(1,30),(30,69),(32,1)]; [(30,3),(62,64),(64,66),(63,60),(34,69),(59,29)]; [(1,71),(71,3),(23,12),(12,14),(14,11),(10,8),(8,7),(7,13),(2,30),(30,73),(83,82)];																																					
Iteração 12		1	31	33	105	38	40	51	52	53	54	56	59	61	63	60	65	67	68	57	55	50	46	47	48	45	49	44	43	42	41	39	37	34	69	29	88	86	1
rota1	1	2	30	70	73	74	75	76	80	79	78	77	81	72	83	84	85	87	89	90	91	93	94	96	97	98	100	101	103	104	102	95	99	92	88	86	1	1	
rota2	1	2	30	70	73	74	75	76	80	79	78	77	81	72	83	84	85	87	89	90	91	93	94	96	97	98	100	101	103	104	102	95	99	92	88	86	1	1	
rota3	1	71	3	4	5	28	6	24	27	26	25	23	12	14	11	10	8	7	13	15	16	17	18	19	20	21	22	9	62	64	66	58	36	35	32	86	1		
distância	44	917	16	190	14	341																																	
Lista Tabu		[(1,31),(31,33),(70,1),(1,30),(30,69),(32,1)]; [(30,3),(62,64),(64,66),(63,60),(34,69),(59,29)]; [(1,71),(71,3),(23,12),(12,14),(14,11),(10,8),(8,7),(7,13),(2,30),(30,73),(83,82)]; [(29,82),(82,1),(30,70),(70,73),(83,84)];																																					
Iteração 13		1	2	69	30	70	73	74	75	76	80	79	78	77	81	72	83	84	85	87	89	90	91	93	94	96	98	100	101	103	104	102	95	99	92	88	86	1	
rota1	1	31	33	105	38	40	51	52	53	54	56	59	61	63	60	65	67	68	57	55	50	46	47	48	45	49	44	43	42	41	39	37	34	29	82	97	1	1	
rota2	1	31	33	105	38	40	51	52	53	54	56	59	61	63	60	65	67	68	57	55	50	46	47	48	45	49	44	43	42	41	39	37	34	29	82	97	1	1	
rota3	1	71	3	4	5	28	6	24	27	26	25	23	12	14	11	10	8	7	13	15	16	17	18	19	20	21	22	9	62	64	66	58	36	35	32	86	1		
distância	46	123	14	894	16	843																																	
Lista Tabu		[(30,3),(62,64),(64,66),(63,60),(34,69),(59,29)]; [(1,71),(71,3),(23,12),(12,14),(14,11),(10,8),(8,7),(7,13),(2,30),(30,73),(83,82)]; [(29,82),(82,1),(30,70),(70,73),(83,84)]; [(2,69),(69,30),(95,98),(34,29),(82,97),(97,1)];																																					

Figura 29: Resultados das iterações do este 9 (parte III)

Iteração 14	
rota1	1 2 69 30 70 73 74 75 76 80 78 77 81 72 83 84 85 87 89 90 91 93 94 96 97 98 100 101 103 104 102 95 99 92 88 86 1
rota2	1 31 33 105 38 40 51 52 53 54 56 59 61 63 60 65 67 68 57 55 50 46 47 48 45 49 44 43 42 41 39 37 34 29 79 82 1
rota3	1 71 3 4 5 28 6 24 27 26 25 23 12 14 11 10 8 7 13 15 16 17 18 19 20 21 22 9 62 64 66 58 36 35 32 1
distância	46 064
	14 940
	17 088
	14 386
Lista Tabu	[(1,71),(71,3),(23,12),(12,14),(14,11),(10,8),(8,7),(7,13),(2,20),(30,73),(83,82)]; [(29,82),(82,1),(30,70),(70,73),(83,84)]; [(2,69),(69,30),(96,58),(34,29),(82,97),(97,11)]; [(80,78),(96,97),(97,98),(29,79),(79,82)];
Iteração 15	
rota1	1 31 33 105 38 40 51 52 53 54 56 59 61 63 60 65 67 68 87 89 90 91 93 94 48 45 49 44 43 42 41 39 37 34 29 82 85 1
rota2	1 2 69 30 70 73 74 75 76 80 79 78 77 81 72 83 84 84 87 89 90 91 93 94 96 97 98 100 101 103 104 102 95 99 92 88 86 1
rota3	1 71 3 4 5 28 6 24 27 26 25 23 12 14 11 10 8 7 13 15 16 17 18 19 20 21 22 9 62 64 66 58 36 35 32 1
distância	46 066
	16 745
	14 935
	14 386
Lista Tabu	[(29,82),(82,1),(30,70),(70,73),(83,84)]; [(2,69),(69,30),(96,58),(34,29),(82,97),(97,11)]; [(80,78),(96,97),(97,98),(29,79),(79,82)]; [(29,82),(82,85),(85,1),(80,79),(79,78),(84,87)];
Iteração 16	
rota1	1 2 69 30 70 74 75 76 80 79 78 77 81 72 83 84 85 87 89 90 91 93 94 96 97 98 100 101 103 104 102 95 99 92 88 86 1
rota2	1 31 33 105 38 40 51 52 53 54 56 59 61 63 60 65 67 68 57 55 50 46 47 48 45 49 44 43 42 41 39 37 34 29 73 82 1
rota3	1 71 3 4 5 28 6 24 27 26 25 23 12 14 11 10 8 7 13 15 16 17 18 19 20 21 22 9 62 64 66 58 36 35 32 1
distância	45 381
	14 979
	16 016
	14 386

5 iterações sem qualquer melhoria

Figura 30: Resultados das iterações do este 9 (parte IV)

• Teste 10

Iteração	rot1	rot2	rot3	distância
Iteração 1	1 33 38 40 51 52 53 54 55 59 61 63 62 64 60 65 67 69 66 58 57 56 50 46 47 48 45 49 44 43 42 41 39 36 37 70 1	1 2 29 30 3 4 5 28 6 24 27 26 25 23 7 8 11 10 14 12 13 15 16 17 18 19 20 21 22 9 57 36 35 32 31 1	1 70 73 74 75 76 80 79 78 77 81 72 83 71 82 84 85 87 89 90 91 93 94 96 97 98 100 101 103 105 102 95 99 92 88 86 1	46 938
Iteração 2	1 33 104 38 40 51 52 53 54 55 59 61 63 62 64 60 65 67 69 66 58 57 56 50 46 47 48 45 49 44 43 42 41 39 36 37 1	1 70 73 74 75 76 80 79 78 77 81 72 83 71 82 84 85 87 89 90 91 93 94 96 97 98 100 101 103 105 102 95 99 92 88 86 1	1 2 29 30 3 4 5 28 6 24 27 26 25 23 7 8 11 10 14 12 13 15 16 17 18 19 20 21 22 9 68 35 34 32 31 1	46 520
Iteração 3	1 33 104 38 40 51 52 53 54 55 59 61 63 62 64 60 65 67 69 66 58 57 56 50 46 47 48 45 49 44 43 42 41 39 37 1	1 2 29 30 3 4 5 28 6 24 27 26 25 23 7 8 11 10 14 12 13 15 16 17 18 19 20 21 22 9 56 35 34 32 31 1	1 70 73 74 75 76 80 79 78 77 81 72 83 71 82 84 85 87 89 90 91 93 94 96 97 98 100 101 103 105 102 95 99 92 88 86 1	46 903
Iteração 4	1 33 104 38 40 51 52 53 54 55 59 61 63 62 64 60 65 67 69 66 58 57 56 50 46 47 48 45 49 44 43 42 41 39 37 1	1 2 29 30 3 4 5 28 6 24 27 26 25 23 7 8 11 10 14 12 13 15 16 17 18 19 20 21 22 9 57 36 35 32 31 1	1 70 73 74 75 76 80 79 78 77 81 72 83 71 82 84 85 87 89 90 91 93 94 96 97 98 100 101 103 105 102 95 99 92 88 86 1	46 356
Iteração 5	1 2 29 30 69 3 4 5 28 23 24 27 26 25 6 7 11 10 14 12 13 15 16 17 18 19 20 21 22 9 57 36 35 32 31 1	1 33 104 38 40 51 52 53 54 55 59 61 63 62 64 60 65 67 69 66 58 57 56 50 46 47 48 45 49 44 43 42 41 39 37 1	1 70 73 74 75 76 80 79 78 77 81 72 83 71 82 84 85 87 89 90 91 93 94 96 97 98 100 101 103 105 102 95 99 92 88 86 1	46 390

Iteração	rot1	rot2	rot3	distância
Iteração 1	1 38 40 51 52 53 54 55 59 61 63 62 64 60 65 67 69 66 58 57 56 50 46 47 48 45 49 44 43 42 41 39 36 37 70 1	1 2 29 30 3 4 5 28 6 24 27 26 25 23 7 8 11 10 14 12 13 15 16 17 18 19 20 21 22 9 68 35 34 32 31 1	1 73 74 75 76 80 79 78 77 81 72 83 71 82 84 85 87 89 90 91 93 94 96 97 98 100 101 103 105 102 95 99 92 88 86 1	47 177
Iteração 2	1 38 40 51 52 53 54 55 59 61 63 62 64 60 65 67 69 66 58 57 56 50 46 47 48 45 49 44 43 42 41 39 36 37 70 1	1 2 29 30 3 4 5 28 6 24 27 26 25 23 7 8 11 10 14 12 13 15 16 17 18 19 20 21 22 9 68 35 34 32 31 1	1 73 74 75 76 80 79 78 77 81 72 83 71 82 84 85 87 89 90 91 93 94 96 97 98 100 101 103 105 102 95 99 92 88 86 1	16 421
Iteração 3	1 38 40 51 52 53 54 55 59 61 63 62 64 60 65 67 69 66 58 57 56 50 46 47 48 45 49 44 43 42 41 39 36 37 70 1	1 2 29 30 3 4 5 28 6 24 27 26 25 23 7 8 11 10 14 12 13 15 16 17 18 19 20 21 22 9 68 35 34 32 31 1	1 73 74 75 76 80 79 78 77 81 72 83 71 82 84 85 87 89 90 91 93 94 96 97 98 100 101 103 105 102 95 99 92 88 86 1	15 463
Iteração 4	1 38 40 51 52 53 54 55 59 61 63 62 64 60 65 67 69 66 58 57 56 50 46 47 48 45 49 44 43 42 41 39 36 37 70 1	1 2 29 30 3 4 5 28 6 24 27 26 25 23 7 8 11 10 14 12 13 15 16 17 18 19 20 21 22 9 68 35 34 32 31 1	1 73 74 75 76 80 79 78 77 81 72 83 71 82 84 85 87 89 90 91 93 94 96 97 98 100 101 103 105 102 95 99 92 88 86 1	15 303

Iteração	rot1	rot2	rot3	distância
Iteração 1	1 33 38 40 51 52 53 54 55 59 61 63 62 64 60 65 67 69 66 58 57 56 50 46 47 48 45 49 44 43 42 41 39 36 37 1	1 2 29 30 3 4 5 28 6 24 27 26 25 23 7 8 11 10 14 12 13 15 16 17 18 19 20 21 22 9 68 35 34 32 31 1	1 73 74 75 76 80 79 78 77 81 72 83 71 82 84 85 87 89 90 91 93 94 96 97 98 100 101 103 105 102 95 99 92 88 86 1	46 938
Iteração 2	1 33 104 38 40 51 52 53 54 55 59 61 63 62 64 60 65 67 69 66 58 57 56 50 46 47 48 45 49 44 43 42 41 39 36 37 1	1 70 73 74 75 76 80 79 78 77 81 72 83 71 82 84 85 87 89 90 91 93 94 96 97 98 100 101 103 105 102 95 99 92 88 86 1	1 2 29 30 3 4 5 28 6 24 27 26 25 23 7 8 11 10 14 12 13 15 16 17 18 19 20 21 22 9 68 35 34 32 31 1	46 520
Iteração 3	1 33 104 38 40 51 52 53 54 55 59 61 63 62 64 60 65 67 69 66 58 57 56 50 46 47 48 45 49 44 43 42 41 39 37 1	1 2 29 30 3 4 5 28 6 24 27 26 25 23 7 8 11 10 14 12 13 15 16 17 18 19 20 21 22 9 56 35 34 32 31 1	1 70 73 74 75 76 80 79 78 77 81 72 83 71 82 84 85 87 89 90 91 93 94 96 97 98 100 101 103 105 102 95 99 92 88 86 1	46 903
Iteração 4	1 33 104 38 40 51 52 53 54 55 59 61 63 62 64 60 65 67 69 66 58 57 56 50 46 47 48 45 49 44 43 42 41 39 37 1	1 2 29 30 3 4 5 28 6 24 27 26 25 23 7 8 11 10 14 12 13 15 16 17 18 19 20 21 22 9 57 36 35 32 31 1	1 70 73 74 75 76 80 79 78 77 81 72 83 71 82 84 85 87 89 90 91 93 94 96 97 98 100 101 103 105 102 95 99 92 88 86 1	46 356
Iteração 5	1 2 29 30 69 3 4 5 28 23 24 27 26 25 6 7 11 10 14 12 13 15 16 17 18 19 20 21 22 9 57 36 35 32 31 1	1 33 104 38 40 51 52 53 54 55 59 61 63 62 64 60 65 67 69 66 58 57 56 50 46 47 48 45 49 44 43 42 41 39 37 1	1 70 73 74 75 76 80 79 78 77 81 72 83 71 82 84 85 87 89 90 91 93 94 96 97 98 100 101 103 105 102 95 99 92 88 86 1	46 390

Figura 31: Resultados das iterações do Teste 10 (parte I)

Iteração 6	
rota1	1 70 73 74 75 76 80 79 78 77 81 72 83 71 82 84 85 87 89 90 91 93 94 96 97 98 100 103 104 105 102 95 99 92 88 86 1
rota2	1 33 101 38 40 51 52 53 54 55 59 61 63 8 62 64 60 65 67 68 66 58 56 50 46 47 48 45 49 44 43 42 41 39 37 34 1
rota3	1 2 29 30 69 3 4 5 28 23 24 27 26 25 6 7 11 10 14 12 13 15 16 17 18 19 20 21 22 9 57 36 35 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
distância	45 979 15 110 14 724 16 145
Lista Tabu	
[(67,68),(68,69),(39,37),(9,36),(36,35)];	
[(28,23),(23,24),(25,6),(7,11),(9,57),(57,36),(63,8),(6,62),(58,56)];	
[(100,103),(103,104),(104,105),(33,101),(101,38)];	
Iteração 7	
rota1	1 101 38 40 51 52 53 54 56 59 61 63 8 62 64 60 65 67 68 66 58 55 50 46 47 48 45 49 44 43 42 41 39 37 34 32 1
rota2	1 2 29 30 69 3 4 5 28 23 24 27 26 25 6 7 11 10 14 12 13 15 16 17 18 19 20 21 22 9 57 36 35 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
rota3	1 70 73 74 75 76 80 79 78 77 81 72 83 71 82 84 85 87 89 90 91 93 94 96 97 98 100 103 104 105 102 95 99 92 88 86 1
distância	46 609 14 211 17 288 15 110
Lista Tabu	
[(68,66),(37,34),(34,1),(30,69),(69,3),(35,32)];	
[(28,23),(23,24),(25,6),(7,11),(9,57),(57,36),(63,8),(6,62),(58,56)];	
[(100,103),(103,104),(104,105),(33,101),(101,38)];	
[(1,101),(54,56),(56,59),(58,55),(55,50),(34,32),(32,1),(35,31),(31,33),(33,1)];	
Iteração 8	
rota1	1 2 29 30 69 3 4 5 28 27 24 23 26 25 6 7 11 10 14 12 13 15 16 17 18 19 20 21 22 9 36 35 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
rota2	1 101 38 40 51 52 53 54 56 57 59 61 63 8 62 64 60 65 67 68 66 58 55 50 46 47 48 45 49 44 43 42 41 39 37 34 1
rota3	1 70 73 74 75 76 80 79 78 77 81 72 83 71 82 84 85 87 89 90 91 93 94 96 97 98 100 103 104 105 102 95 99 92 88 86 1
distância	46 407 16 786 14 511 15 110
Lista Tabu	
[(28,23),(23,24),(25,6),(7,11),(9,57),(57,36),(63,8),(6,62),(58,56)];	
[(100,103),(103,104),(104,105),(33,101),(101,38)];	
[(1,101),(54,56),(56,59),(58,55),(55,50),(34,32),(32,1),(35,31),(31,33),(33,1)];	
[(28,27),(27,24),(24,23),(23,26),(9,36),(35,32),(32,31),(56,57),(57,59),(34,1)];	
Iteração 9	
rota1	1 29 30 69 3 4 5 28 27 24 23 26 25 6 7 11 10 14 12 13 15 16 17 18 19 20 21 22 9 105 36 35 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
rota2	1 2 70 73 74 75 76 80 79 78 77 81 72 83 71 82 84 85 87 89 90 91 93 94 96 97 98 100 103 104 102 95 99 92 88 86 1
rota3	1 101 38 40 51 52 53 54 56 57 59 61 63 8 62 64 60 65 67 68 66 58 55 50 46 47 48 45 49 44 43 42 41 39 37 34 1
distância	46 291 17 440 14 340 14 511

5 iterações sem qualquer melhoria

Figura 32: Resultados das iterações do Teste 10 (parte II)

Iteração 5		1	33	38	40	51	52	53	54	56	59	61	63	62	64	60	65	67	68	66	58	57	55	50	46	47	48	45	49	44	43	42	41	39	37	102	96	1			
rotal1	1	69	73	74	75	76	80	79	78	77	81	72	83	71	82	84	85	87	89	90	91	93	94	93	94	97	98	100	101	104	36	105	103	95	99	92	88	86	1		
rotal2	1	84	33	38	40	51	52	53	54	56	59	61	63	62	64	60	65	67	68	66	58	57	55	50	46	47	48	45	49	44	43	42	41	39	37	101	96	1			
rotal3	1	2	29	70	30	3	4	5	28	6	35	27	26	25	23	7	8	11	10	14	12	13	15	16	17	18	19	20	21	22	9	24	34	32	31	1	1				
distância		47 025																																							
		14 997																																							
		16 183																																							
		15 845																																							
Lista Tabu		[[33,103],[105,38],[67,68],[1,69],[69,73],[103,104]];																																							
		[[103,105],[105,104],[104,95],[33,38],[37,102],[102,1]];																																							
		[[100,104],[104,36],[105,103],[103,95],[39,37],[102,101],[101,1]];																																							
		[[37,101],[101,96],[100,102],[102,104]];																																							
		[[37,101],[101,96],[100,102],[102,104]];																																							
Iteração 6		1	69	73	74	75	76	80	79	78	77	81	72	83	71	82	84	85	87	89	90	91	93	94	97	98	100	102	104	36	105	103	95	99	92	88	86	1			
rotal1	1	33	38	40	51	52	53	54	56	59	61	63	62	64	60	65	67	68	66	58	57	55	50	46	47	48	45	49	44	43	42	41	39	37	101	96	1				
rotal2	1	2	29	70	30	3	4	5	28	6	35	27	26	25	23	7	8	11	10	14	12	13	15	16	17	18	19	20	21	22	9	24	34	32	31	1	1				
rotal3	1	1	2	29	70	30	3	4	5	28	6	35	27	26	25	23	7	8	11	10	14	12	13	15	16	17	18	19	20	21	22	9	24	34	32	31	1	1			
distância		47 311																																							
		16 198																																							
		15 268																																							
		15 845																																							
Lista Tabu		[[103,105],[105,104],[104,95],[33,38],[37,102],[102,1]];																																							
		[[100,104],[104,36],[105,103],[103,95],[39,37],[102,101],[101,1]];																																							
		[[37,101],[101,96],[100,102],[102,104]];																																							
		[[100,102],[102,104],[37,101],[101,96],[37,101]];																																							
Iteração 7		1	69	73	74	75	76	80	79	78	77	81	72	83	71	82	84	85	87	89	90	91	93	94	97	98	100	102	104	36	105	103	95	99	92	88	86	1			
rotal1	1	84	33	38	40	51	52	53	54	56	59	61	63	62	64	60	65	67	68	66	58	57	55	50	46	47	48	45	49	44	43	42	41	39	37	101	96	1			
rotal2	1	2	29	70	30	3	4	5	28	6	35	27	26	25	23	7	8	11	10	14	12	13	15	16	17	18	19	20	21	22	9	24	34	32	31	1	1				
rotal3	1	1	2	29	70	30	3	4	5	28	6	35	27	26	25	23	7	8	11	10	14	12	13	15	16	17	18	19	20	21	22	9	24	34	32	31	1	1			
distância		47 894																																							
		16 198																																							
		15 841																																							
		15 845																																							

Figura 34: Resultados das iterações do Teste 11 (parte II)

• Teste 1 VRP Spreadsheet Solver

Stop count	Location name	Distance travelled	Driving time	Arrival time	Departure time	Working time	Profit collected	Load	Stop count	Location name	Distance travelled	Driving time	Arrival time	Departure time	Working time	Profit collected	Load	Stop count	Location name	Distance travelled	Driving time	Arrival time	Departure time	Working time	Profit collected	Load
0	1	0.00	0.00	07:45	07:45	0.00	0.00	0	0	1	0.00	0.00	07:45	07:45	0.00	0.00	0	0	1	0.00	0.00	07:45	07:45	0.00	0.00	0
1	31	0.20	0.00	07:45	07:47	0.02	0.00	0	1	57	1.41	0.00	07:45	07:47	0.02	0.00	0	1	57	1.41	0.00	07:45	07:47	0.02	0.00	0
2	105	0.09	0.00	07:48	07:51	0.06	0.00	0	2	42	1.69	0.00	07:48	07:51	0.06	0.00	0	2	42	1.69	0.00	07:48	07:51	0.06	0.00	0
3	105	0.09	0.00	07:48	07:51	0.06	0.00	0	3	63	2.68	0.00	07:48	07:51	0.06	0.00	0	3	63	2.68	0.00	07:48	07:51	0.06	0.00	0
4	104	1.38	0.00	07:51	07:53	0.08	0.00	0	4	43	1.73	0.00	07:51	07:53	0.08	0.00	0	4	43	1.73	0.00	07:51	07:53	0.08	0.00	0
5	103	1.55	0.00	07:53	07:55	0.10	0.00	0	5	44	2.18	0.00	07:53	07:55	0.10	0.00	0	5	44	2.18	0.00	07:53	07:55	0.10	0.00	0
6	102	1.86	0.00	07:55	07:57	0.12	0.00	0	6	45	2.32	0.00	07:55	07:57	0.12	0.00	0	6	45	2.32	0.00	07:55	07:57	0.12	0.00	0
7	101	2.07	0.00	07:57	07:59	0.14	0.00	0	7	46	2.64	0.00	07:57	07:59	0.14	0.00	0	7	46	2.64	0.00	07:57	07:59	0.14	0.00	0
8	100	2.20	0.00	07:59	08:01	0.16	0.00	0	8	47	3.12	0.00	07:59	08:01	0.16	0.00	0	8	47	3.12	0.00	07:59	08:01	0.16	0.00	0
9	100	2.20	0.00	07:59	08:01	0.16	0.00	0	9	48	3.22	0.00	07:59	08:01	0.16	0.00	0	9	48	3.22	0.00	07:59	08:01	0.16	0.00	0
10	92	3.31	0.00	08:03	08:05	0.20	0.00	0	10	49	3.91	0.00	08:03	08:05	0.20	0.00	0	10	49	3.91	0.00	08:03	08:05	0.20	0.00	0
11	88	3.72	0.00	08:05	08:07	0.22	0.00	0	11	50	4.26	0.00	08:05	08:07	0.22	0.00	0	11	50	4.26	0.00	08:05	08:07	0.22	0.00	0
12	86	4.03	0.00	08:07	08:09	0.24	0.00	0	12	51	4.76	0.00	08:07	08:09	0.24	0.00	0	12	51	4.76	0.00	08:07	08:09	0.24	0.00	0
13	85	4.15	0.00	08:09	08:11	0.26	0.00	0	13	52	5.09	0.00	08:09	08:11	0.26	0.00	0	13	52	5.09	0.00	08:09	08:11	0.26	0.00	0
14	82	4.46	0.00	08:11	08:13	0.28	0.00	0	14	53	5.21	0.00	08:11	08:13	0.28	0.00	0	14	53	5.21	0.00	08:11	08:13	0.28	0.00	0
15	80	4.69	0.00	08:13	08:15	0.30	0.00	0	15	54	5.47	0.00	08:13	08:15	0.30	0.00	0	15	54	5.47	0.00	08:13	08:15	0.30	0.00	0
16	80	4.69	0.00	08:13	08:15	0.30	0.00	0	16	55	5.77	0.00	08:13	08:15	0.30	0.00	0	16	55	5.77	0.00	08:13	08:15	0.30	0.00	0
17	91	5.17	0.00	08:17	08:19	0.34	0.00	0	17	56	6.08	0.00	08:17	08:19	0.34	0.00	0	17	56	6.08	0.00	08:17	08:19	0.34	0.00	0
18	93	5.88	0.00	08:19	08:21	0.36	0.00	0	18	41	6.95	0.00	08:19	08:21	0.36	0.00	0	18	41	6.95	0.00	08:19	08:21	0.36	0.00	0
19	94	6.13	0.00	08:21	08:23	0.38	0.00	0	19	39	7.15	0.00	08:21	08:23	0.38	0.00	0	19	39	7.15	0.00	08:21	08:23	0.38	0.00	0
20	95	6.64	0.00	08:23	08:25	0.40	0.00	0	20	37	7.30	0.00	08:23	08:25	0.40	0.00	0	20	37	7.30	0.00	08:23	08:25	0.40	0.00	0
21	96	6.87	0.00	08:25	08:27	0.42	0.00	0	21	36	7.57	0.00	08:25	08:27	0.42	0.00	0	21	36	7.57	0.00	08:25	08:27	0.42	0.00	0
22	92	7.12	0.00	08:27	08:29	0.44	0.00	0	22	35	7.85	0.00	08:27	08:29	0.44	0.00	0	22	35	7.85	0.00	08:27	08:29	0.44	0.00	0
23	92	7.12	0.00	08:27	08:29	0.44	0.00	0	23	32	8.48	0.00	08:31	08:33	0.48	0.00	0	23	32	8.48	0.00	08:31	08:33	0.48	0.00	0
24	1	6.04	0.00	08:31	08:31	0.46	0.00	0	24	2	8.88	0.00	08:33	08:35	0.50	0.00	0	24	2	8.88	0.00	08:33	08:35	0.50	0.00	0
25	26							0	25	3	8.88	0.00	08:35	08:35	0.52	0.00	0	25	3	8.88	0.00	08:35	08:35	0.52	0.00	0
26	27							0	26	4	9.03	0.00	08:37	08:39	0.54	0.00	0	26	4	9.03	0.00	08:37	08:39	0.54	0.00	0
27	27							0	27	5	9.26	0.00	08:39	08:41	0.56	0.00	0	27	5	9.26	0.00	08:39	08:41	0.56	0.00	0
28	28							0	28	6	9.59	0.00	08:41	08:43	0.58	0.00	0	28	6	9.59	0.00	08:41	08:43	0.58	0.00	0
29	29							0	29	7	10.27	0.00	08:45	08:47	1.02	0.00	0	29	7	10.27	0.00	08:45	08:47	1.02	0.00	0
30	31							0	30	7	10.27	0.00	08:45	08:47	1.02	0.00	0	30	7	10.27	0.00	08:45	08:47	1.02	0.00	0
31	28							0	31	7	11.02	0.00	08:47	08:49	1.04	0.00	0	31	7	11.02	0.00	08:47	08:49	1.04	0.00	0
32	27							0	32	7	11.50	0.00	08:49	08:51	1.06	0.00	0	32	7	11.50	0.00	08:49	08:51	1.06	0.00	0
33	28							0	33	7	12.20	0.00	08:51	08:53	1.08	0.00	0	33	7	12.20	0.00	08:51	08:53	1.08	0.00	0
34	67							0	34	84	12.53	0.00	08:53	08:55	1.10	0.00	0	34	84	12.53	0.00	08:53	08:55	1.10	0.00	0
35	35							0	35	84	12.86	0.00	08:55	08:57	1.12	0.00	0	35	84	12.86	0.00	08:55	08:57	1.12	0.00	0
36	35							0	36	72	13.49	0.00	08:59	09:01	1.16	0.00	0	36	72	13.49	0.00	08:59	09:01	1.16	0.00	0
37	39							0	37	70	13.16	0.00	08:59	09:01	1.16	0.00	0	37	70	13.16	0.00	08:59	09:01	1.16	0.00	0
38	38							0	38	71	13.37	0.00	09:03	09:05	1.18	0.00	0	38	71	13.37	0.00	09:03	09:05	1.18	0.00	0
39	39							0	39	80	13.70	0.00	09:03	09:05	1.20	0.00	0	39	80	13.70	0.00	09:03	09:05	1.20	0.00	0
40	40							0	40	79	14.09	0.00	09:05	09:07	1.22	0.00	0	40	79	14.09	0.00	09:05	09:07	1.22	0.00	0
41	41							0	41	78	14.53	0.00	09:07	09:09	1.24	0.00	0	41	78	14.53	0.00	09:07	09:09	1.24	0.00	0
42	42							0	42	77	14.99	0.00	09:09	09:11	1.26	0.00	0	42	77	14.99	0.00	09:09	09:11	1.26	0.00	0
43	43							0	43	76	15.48	0.00	09:11	09:13	1.28	0.00	0	43	76	15.48	0.00	09:11	09:13	1.28	0.00	0
44	44							0	44	30	16.56	0.00	09:13	09:15	1.30	0.00	0	44	30	16.56	0.00	09:13	09:15	1.30	0.00	0
45	45							0	45	30	16.56	0.00	09:13	09:15	1.30	0.00	0	45	30	16.56	0.00	09:13	09:15	1.30	0.00	0
46	46							0	46	1	16.95	0.00	09:15	09:15	1.30	0.00	0	46	1	16.95	0.00	09:15	09:15	1.30	0.00	0

Figura 35: Resultados do teste com 3 linhas no Microsoft Excel VRP Spreadsheet Solver

• Teste 2 VRP Spreadsheet Solver

Stop count	Location name	Distance travelled	Driving time	Arrival time	Departure time	Working time	Profit collected	Load	Stop count	Location name	Distance travelled	Driving time	Arrival time	Departure time	Working time	Profit collected	Load
0	1	0,00	0,00	07:45	07:45	0,00	0,00	0	0	1	0,00	0,00	07:45	07:45	0,00	0,00	0
1	31	0,20	0,00	07:45	07:47	0,02	0,00	0	1	69	0,72	0,00	07:47	07:49	0,02	0,00	0
2	33	0,47	0,00	07:47	07:49	0,04	0,00	0	2	68	1,07	0,00	07:49	07:51	0,04	0,00	0
3	104	1,39	0,00	07:49	07:51	0,06	0,00	0	3	66	1,78	0,00	07:49	07:51	0,06	0,00	0
4	103	1,56	0,00	07:51	07:53	0,08	0,00	0	4	58	1,86	0,00	07:51	07:53	0,08	0,00	0
5	101	1,95	0,00	07:53	07:55	0,10	0,00	0	5	50	3,08	0,00	07:53	07:55	0,10	0,00	0
6	102	2,16	0,00	07:55	07:57	0,12	0,00	0	6	47	4,23	0,00	07:55	07:57	0,12	0,00	0
7	95	2,67	0,00	07:57	07:59	0,14	0,00	0	7	48	4,56	0,00	07:57	07:59	0,14	0,00	0
8	96	2,90	0,00	07:59	08:01	0,16	0,00	0	8	46	4,70	0,00	07:59	08:01	0,16	0,00	0
9	99	3,33	0,00	08:01	08:03	0,18	0,00	0	9	45	5,02	0,00	08:01	08:03	0,18	0,00	0
10	92	4,02	0,00	08:03	08:05	0,20	0,00	0	10	49	5,02	0,00	08:03	08:05	0,20	0,00	0
11	93	4,32	0,00	08:05	08:07	0,22	0,00	0	11	44	5,18	0,00	08:05	08:07	0,22	0,00	0
12	94	4,56	0,00	08:07	08:09	0,24	0,00	0	12	43	5,59	0,00	08:07	08:09	0,24	0,00	0
13	97	4,91	0,00	08:09	08:11	0,26	0,00	0	13	42	5,69	0,00	08:09	08:11	0,26	0,00	0
14	98	5,10	0,00	08:11	08:13	0,28	0,00	0	14	41	5,95	0,00	08:11	08:13	0,28	0,00	0
15	100	5,32	0,00	08:13	08:15	0,30	0,00	0	15	39	6,15	0,00	08:13	08:15	0,30	0,00	0
16	35	6,04	0,00	08:15	08:17	0,32	0,00	0	16	37	6,31	0,00	08:15	08:17	0,32	0,00	0
17	34	6,43	0,00	08:17	08:19	0,34	0,00	0	17	36	6,57	0,00	08:17	08:19	0,34	0,00	0
18	32	6,70	0,00	08:19	08:21	0,36	0,00	0	18	105	6,90	0,00	08:19	08:21	0,36	0,00	0
19	2	6,99	0,00	08:21	08:23	0,38	0,00	0	19	38	7,20	0,00	08:21	08:23	0,38	0,00	0
20	3	7,25	0,00	08:23	08:25	0,40	0,00	0	20	40	7,21	0,00	08:23	08:25	0,40	0,00	0
21	4	7,47	0,00	08:25	08:27	0,42	0,00	0	21	51	8,36	0,00	08:25	08:27	0,42	0,00	0
22	5	7,69	0,00	08:27	08:29	0,44	0,00	0	22	52	8,69	0,00	08:27	08:29	0,44	0,00	0
23	29	7,80	0,00	08:29	08:31	0,46	0,00	0	23	53	8,81	0,00	08:29	08:31	0,46	0,00	0
24	70	8,26	0,00	08:31	08:33	0,48	0,00	0	24	54	9,07	0,00	08:31	08:33	0,48	0,00	0
25	71	8,47	0,00	08:33	08:35	0,50	0,00	0	25	54	9,36	0,00	08:33	08:35	0,50	0,00	0
26	82	8,58	0,00	08:35	08:37	0,52	0,00	0	26	56	9,68	0,00	08:35	08:37	0,52	0,00	0
27	83	8,92	0,00	08:37	08:39	0,54	0,00	0	27	57	10,04	0,00	08:37	08:39	0,54	0,00	0
28	72	9,12	0,00	08:39	08:41	0,56	0,00	0	28	63	10,51	0,00	08:39	08:41	0,56	0,00	0
29	73	9,44	0,00	08:41	08:43	0,58	0,00	0	29	67	10,84	0,00	08:41	08:43	0,58	0,00	0
30	74	9,64	0,00	08:43	08:45	1,00	0,00	0	30	6	12,81	0,00	08:43	08:45	1,02	0,00	0
31	75	10,09	0,00	08:45	08:47	1,04	0,00	0	31	62	12,81	0,00	08:45	08:47	1,04	0,00	0
32	80	10,57	0,00	08:47	08:49	1,06	0,00	0	32	64	13,09	0,00	08:47	08:49	1,06	0,00	0
33	80	11,20	0,00	08:49	08:51	1,06	0,00	0	33	60	13,29	0,00	08:49	08:51	1,06	0,00	0
34	78	11,27	0,00	08:51	08:52	1,10	0,00	0	34	60	14,15	0,00	08:51	08:52	1,06	0,00	0
35	32	11,27	0,00	08:52	08:52	1,10	0,00	0	35	27	14,15	0,00	08:52	08:52	1,10	0,00	0
36	77	12,46	0,00	08:52	08:52	1,12	0,00	0	36	63	14,62	0,00	08:52	08:52	1,12	0,00	0
37	84	13,20	0,00	08:52	08:52	1,14	0,00	0	37	16	15,60	0,00	08:52	08:52	1,14	0,00	0
38	84	13,80	0,00	08:52	08:52	1,14	0,00	0	38	17	15,82	0,00	08:52	08:52	1,14	0,00	0
39	86	14,32	0,00	08:52	08:52	1,16	0,00	0	39	18	16,69	0,00	08:52	08:52	1,16	0,00	0
40	87	14,64	0,00	08:52	08:52	1,20	0,00	0	40	16	16,99	0,00	08:52	08:52	1,20	0,00	0
41	89	15,06	0,00	08:52	08:52	1,22	0,00	0	41	16	17,11	0,00	08:52	08:52	1,22	0,00	0
42	90	15,21	0,00	08:52	08:52	1,24	0,00	0	42	20	17,39	0,00	08:52	08:52	1,24	0,00	0
43	91	15,35	0,00	08:52	08:52	1,26	0,00	0	43	14	17,78	0,00	08:52	08:52	1,26	0,00	0
44	88	15,77	0,00	08:52	08:52	1,28	0,00	0	44	13	17,98	0,00	08:52	08:52	1,28	0,00	0
45	86	16,08	0,00	08:52	08:52	1,30	0,00	0	45	12	18,44	0,00	08:52	08:52	1,30	0,00	0
46	1	16,87	0,00	08:52	08:52	1,30	0,00	0	46	11	18,93	0,00	08:52	08:52	1,30	0,00	0
47									47	10	19,30	0,00	08:52	08:52	1,30	0,00	0
48									48	21	19,56	0,00	08:52	08:52	1,30	0,00	0
49									49	22	19,86	0,00	08:52	08:52	1,30	0,00	0
50									50	9	19,92	0,00	08:52	08:52	1,42	0,00	0
51									51	8	20,22	0,00	08:52	08:52	1,42	0,00	0
52									52	7	20,57	0,00	08:52	08:52	1,46	0,00	0
53									53	23	20,94	0,00	08:52	08:52	1,46	0,00	0
54									54	24	21,21	0,00	08:52	08:52	1,46	0,00	0
55									55	25	21,40	0,00	08:52	08:52	1,50	0,00	0
56									56	26	21,58	0,00	08:52	08:52	1,52	0,00	0
57									57	27	21,93	0,00	08:52	08:52	1,52	0,00	0
58									58	28	22,30	0,00	08:52	08:52	1,56	0,00	0
59									59	30	23,27	0,00	08:52	08:52	1,56	0,00	0
60									60	1	23,66	0,00	08:52	08:52	1,58	0,00	0

Figura 36: Resultados do teste com 2 linhas no Microsoft Excel VRP Spreadsheet Solver

• Teste 3 VRP Spreadsheet Solver

Stop count	Location as	Distance	Travel	Driving	Arrival	Departure	Working	Profit	Collec	Load
0	1	0,00	0,00	0,00	07:45	0,00	0,00	0	0	0
1	2	0,43	0,00	0,00	07:45	0,00	0,00	0	0	0
2	3	0,43	0,00	0,00	07:45	0,00	0,00	0	0	0
3	4	0,14	0,00	0,00	07:43	0,00	0,00	0	0	0
4	5	0,31	0,00	0,00	07:51	0,00	0,00	0	0	0
5	6	1,83	0,00	0,00	07:55	0,00	0,00	0	0	0
6	7	2,17	0,00	0,00	07:55	0,00	0,00	0	0	0
7	8	2,54	0,00	0,00	07:55	0,00	0,00	0	0	0
8	9	3,00	0,00	0,00	08:01	0,00	0,00	0	0	0
9	10	3,18	0,00	0,00	08:03	0,00	0,00	0	0	0
10	11	3,53	0,00	0,00	08:05	0,00	0,00	0	0	0
11	12	3,90	0,00	0,00	08:07	0,00	0,00	0	0	0
12	13	5,32	0,00	0,00	08:09	0,00	0,00	0	0	0
13	14	6,48	0,00	0,00	08:11	0,00	0,00	0	0	0
14	15	6,32	0,00	0,00	08:13	0,00	0,00	0	0	0
15	16	6,53	0,00	0,00	08:15	0,00	0,00	0	0	0
16	17	6,67	0,00	0,00	08:17	0,00	0,00	0	0	0
17	18	6,39	0,00	0,00	08:19	0,00	0,00	0	0	0
18	19	7,91	0,00	0,00	08:21	0,00	0,00	0	0	0
19	20	7,42	0,00	0,00	08:23	0,00	0,00	0	0	0
20	21	6,27	0,00	0,00	08:25	0,00	0,00	0	0	0
21	22	3,91	0,00	0,00	08:27	0,00	0,00	0	0	0
22	23	3,44	0,00	0,00	08:29	0,00	0,00	0	0	0
23	24	3,57	0,00	0,00	08:31	0,00	0,00	0	0	0
24	25	3,93	0,00	0,00	08:33	0,00	0,00	0	0	0
25	26	4,48	0,00	0,00	08:35	0,00	0,00	0	0	0
26	27	10,97	0,00	0,00	08:37	0,00	0,00	0	0	0
27	28	11,30	0,00	0,00	08:39	0,00	0,00	0	0	0
28	29	11,93	0,00	0,00	08:41	0,00	0,00	0	0	0
29	30	12,05	0,00	0,00	08:43	0,00	0,00	0	0	0
30	31	12,78	0,00	0,00	08:45	0,00	0,00	0	0	0
31	32	13,44	0,00	0,00	08:47	0,00	0,00	0	0	0
32	33	13,56	0,00	0,00	08:49	0,00	0,00	0	0	0
33	34	13,96	0,00	0,00	08:51	0,00	0,00	0	0	0
34	35	14,21	0,00	0,00	08:53	0,00	0,00	0	0	0
35	36	14,63	0,00	0,00	08:55	0,00	0,00	0	0	0
36	37	15,22	0,00	0,00	08:57	0,00	0,00	0	0	0
37	38	15,98	0,00	0,00	08:59	0,00	0,00	0	0	0
38	39	16,14	0,00	0,00	09:01	0,00	0,00	0	0	0
39	40	16,61	0,00	0,00	09:03	0,00	0,00	0	0	0
40	41	17,70	0,00	0,00	09:05	0,00	0,00	0	0	0
41	42	18,03	0,00	0,00	09:07	0,00	0,00	0	0	0
42	43	18,43	0,00	0,00	09:09	0,00	0,00	0	0	0
43	44	18,90	0,00	0,00	09:11	0,00	0,00	0	0	0
44	45	19,42	0,00	0,00	09:13	0,00	0,00	0	0	0
45	46	19,42	0,00	0,00	09:13	0,00	0,00	0	0	0
46	47	19,42	0,00	0,00	09:13	0,00	0,00	0	0	0

Figura 37: Resultados do teste sem as 15 paragens adicionais no Microsoft Excel VRP Spreadsheet Solver