

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS
DEPARTAMENTO DE COMPUTAÇÃO

Guilherme Bastos Guimarães

OTIMIZAÇÃO DA REDE DE TRANSPORTE PÚBLICO DE OURO PRETO

Ouro Preto, MG
2021

Guilherme Bastos Guimarães

OTIMIZAÇÃO DA REDE DE TRANSPORTE PÚBLICO DE OURO PRETO

Monografia II apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Gustavo Peixoto Silva

Ouro Preto, MG
2021

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

G963o Guimarães, Guilherme Bastos.
Otimização da rede de transporte público de Ouro Preto. [manuscrito]
/ Guilherme Bastos Guimarães. - 2021.
42 f.: il.: color., tab.. + Pseudocódigo.

Orientador: Prof. Dr. Gustavo Peixoto Silva.
Monografia (Bacharelado). Universidade Federal de Ouro Preto.
Instituto de Ciências Exatas e Biológicas. Graduação em Ciência da
Computação .

1. Otimização. 2. Transporte urbano. 3. Algoritmos. I. Silva, Gustavo
Peixoto. II. Universidade Federal de Ouro Preto. III. Título.

CDU 004

Bibliotecário(a) Responsável: Luciana De Oliveira - SIAPE: 1.937.800



FOLHA DE APROVAÇÃO

Guilherme Bastos Guimarães

Otimização da Rede de Transporte Público de Ouro Preto

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Ciência da Computação

Aprovada em 27 de Abril de 2021.

Membros da banca

Gustavo Peixoto Silva (Orientador) - Doutor - Universidade Federal de Ouro Preto
Alan Robert Resende de Freitas (Examinador) - Doutor - Universidade Federal de Ouro Preto
Rodolfo Ayala (Examinador) - Mestre - Universidade Federal de Ouro Preto

Gustavo Peixoto Silva, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 27/04/2021.



Documento assinado eletronicamente por **Gustavo Peixoto Silva, PROFESSOR DE MAGISTERIO SUPERIOR**, em 28/04/2021, às 14:17, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0160357** e o código CRC **D7777022**.

Agradecimentos

Primeiramente, gostaria de agradecer à minha família por sempre me apoiar no meu tempo de graduação e sempre estarem ao meu lado, principalmente à minha mãe que possibilitou que eu pudesse seguir esta carreira, e sempre trabalhou para que eu pudesse ter as melhores condições para estudo.

Gostaria de agradecer também aos meus amigos que estiveram junto de mim durante toda minha jornada, me ajudando a descontraír nos momentos de descanso. À minha namorada por estar sempre ao meu lado me apoiando.

Gostaria de agradecer à UFOP por todo o meu tempo de graduação e pelos ensinamentos passados a mim pelos professores da universidade. E principalmente ao professor Gustavo Peixoto Silva, pela orientação e ajuda durante todo o processo do trabalho.

Gostaria também de agradecer ao professor Alan Robert Resende de Freitas, da Universidade Federal de Ouro Preto, que participou da banca examinadora da disciplina BCC392 e fez algumas sugestões para alteração do algoritmo. Ao professor Rodolfo Ayala Lopes Costa por dispor de seu tempo para participar da banca de defesa.

Resumo

Este trabalho trata do Problema do Projeto de Redes de Transporte Público por ônibus urbano - PPRTP, que consiste em: i) estabelecer os trajetos das rotas (linhas) de ônibus urbano a serem operadas em uma cidade, ii) definir as tabelas com os horários de partida das viagens de cada rota e iii) definir a frota necessária para atender a demanda da cidade. A rede, composta pelo elementos citados, é criada de tal forma que sejam minimizados os custos dos operadores (empresas de transporta público) e dos usuários do sistema (passageiros), constituindo assim em um problema multiobjetivo. O custo dos operadores é representado tanto pela frota em operação quanto pela distância total percorrida por esta frota, para executar as viagens contidas nas tabelas de horários. Por outro lado, o custo dos usuários é representado pela soma dos tempos de espera, os tempos de viagem dentro do veículo e eventuais transferência realizadas entre diferentes linhas do sistema. Neste trabalho se propõe implementar dois Algoritmos Genéticos - AGs para resolver este problema, considerando uma única função objetivo com custos ponderados. Os métodos de solução basaram-se na construção inicial de um banco de rotas viáveis, sendo que cada solução é construída selecionando-se um subconjunto de rotas deste banco para formar uma solução viável para a rede. Foram aplicadas estratégias de busca por soluções viáveis durante os operadores dos AGs, devido à possibilidade de ocorrência de um grande número de indivíduos inviáveis. Os modelos foram avaliados com uma instância de teste da literatura e os resultados obtidos estiveram dentro do esperado. Uma vez calibrados os Algoritmos Genéticos, foi realizado um *estudo de caso* com os dados da cidade de Ouro Preto referentes ao ano de 2015. Desta forma, pode-se verificar se as soluções otimizadas são passíveis de implementação prática e comparar a eficiência das soluções empíricas, colocadas em prática no período do estudo.

Palavras-chave: Projeto de Redes de Transporte Público. Algoritmo Genético.

Abstract

This project is about the Problem of the Urban Bus Public Transport Network Project - PTNP, which consists of: i) establishing the routes of the bus lines to be operated in a city, ii) defining the timetables with the departure of the trips of each route and iii) define the necessary fleet to meet the demand of the city. The network is created in such a way that the costs of operators and system users are minimized, thus constituting a multiobjective problem. The cost of the operators is represented by both the operating fleet and the total mileage required to perform the journeys contained in the timesheets. On the other hand, the cost of users is represented by the sum of waiting times, travel times inside the vehicle and any transfer penalties between different system lines. In this project we propose to implement two Genetic Algorithms - AGs to solve this problem, considering only one objective function for initial tests. The solution methods are based on the initial construction of a viable route bank, where each solution is built by selecting a subset of routes from this bank to form a viable network solution. Search strategies for viable solutions will be applied during GA operators, due to the possibility of a large number of unviable individuals. Both models were evaluated with a test instance of the literature and the results obtained were as expected. Once the AGs were calibrated, was realized a *case study* with the datas from Ouro Preto city in the year of 2015.

Keywords: Public Transport Network Project. Genetic Algorithm.

Lista de Ilustrações

Figura 2.1 – Rede Inviável pois nem todos os nós são atendidos. Os nós destacados por um círculo maior não são atendidos.	8
Figura 2.2 – Rede Inviável pois as rotas não estão conectadas. A rota destacada à esquerda da figura está isolada do resto da rede.	9
Figura 3.1 – Fluxograma do Algoritmo 1	14
Figura 3.2 – Representação do processo de Cruzamento.	18
Figura 3.3 – Representação do processo de Mutação.	20
Figura 3.4 – Fluxograma do Algoritmo 2	23
Figura 3.5 – Rede de Mandl.	28
Figura 3.6 – Divisões de Ouro Preto	29
Figura 3.7 – Rede de Ouro Preto	30
Figura 4.1 – Resultados do AG1 para a rede de Mandl (1980)	33
Figura 4.2 – Resultados do AG1 para a rede de Ouro Preto	35
Figura 4.3 – Resultados do AG2 para a rede de Mandl (1980)	36
Figura 4.4 – Resultados do AG2 para a rede de Ouro Preto	38

Lista de Tabelas

Tabela 3.1 – Tabela de Origem / Destino da rede de Mandl (1980).	29
Tabela 3.2 – Tabela de Origem / Destino da rede de Ouro Preto.	31
Tabela 4.1 – Comparação entre nossos resultados do Algoritmo 1 e de Arbex e Cunha (2015).	33
Tabela 4.2 – Comparação entre nossos resultados do Algoritmo 1 e do Algoritmo 2 para a rede de Mandl (1980)	37
Tabela 4.3 – Comparação entre nossos resultados do Algoritmo 1 e do Algoritmo 2 para a rede de Ouro Preto	39

Lista de Abreviaturas e Siglas

PPRTP	Problema de Projeto de Rotas de Transporte Público
AG	Algoritmo Genético
O/D	Origem/Destino
FO	Função Objetivo

Sumário

1	Introdução	1
1.1	Justificativa	2
1.2	Objetivos	3
1.3	Organização do Trabalho	3
2	Revisão Bibliográfica	4
2.1	Introdução	4
2.1.1	Representação de uma Solução	4
2.1.2	Função Objetivo	4
2.1.3	Restrições	5
2.1.4	Variáveis de Decisão	8
2.1.5	Dados de Entrada	9
2.1.6	Métodos de Solução	10
2.2	Aplicação a Problemas Reais	11
2.3	Conclusão da Revisão	12
3	Desenvolvimento	13
3.1	Metodologia	13
3.2	Algoritmo 1	13
3.2.1	Banco de Rotas	15
3.2.2	População Inicial	16
3.2.3	Cruzamento	17
3.2.4	Mutação	20
3.2.5	Seleção dos Sobreviventes	21
3.2.6	Estrutura do Programa	22
3.3	Algoritmo 2	23
3.3.1	Cruzamento 2	24
3.3.2	Mutação 2	25
3.3.3	Seleção dos sobreviventes 2	26
3.3.4	Estrutura do Programa 2	27
3.4	Validação	28
3.5	Aplicação em Ouro Preto	29
4	Resultados	32
4.1	Algoritmo 1	32
4.1.1	Rede de Mandl	32
4.1.2	Rede de Ouro Preto	34
4.2	Algoritmo 2	35
4.2.1	Rede de Mandl	36

4.2.2 Rede de Ouro Preto	37
5 Considerações Finais	40
5.1 Conclusão	40
5.2 Trabalhos Futuros	40
Referências	41

1 Introdução

O transporte público por ônibus urbano é um dos componentes mais importantes dentro do sistema de transporte público nos grandes centros, e mesmo no interior, pelo grande número de usuários que transporta diariamente. De acordo com um estudo realizado pela BHTRANS de 2013, cerca de 54% do total de viagens realizadas na hora de pico da manhã ocorrem em transporte coletivo (BHTRANS, 2013). Assim, um projeto adequado para dimensionar a rede das linhas de ônibus urbano tem papel significativo na atratividade deste modo de transporte do ponto de vista dos usuários, e pode ainda, reduzir os custos do sistema como um todo.

O Problema do Projeto de Redes de Transporte Público por ônibus urbano - PP RTP tem como dados de entrada uma rede de transporte, sendo que os nós representam as diferentes regiões ou bairros da cidade em estudo. Os arcos correspondem às possíveis ligações entre os bairros, ou seja, as vias de acesso entre os bairros, e uma solução para o problema é um conjunto de rotas que conecta todos os pontos da cidade. Desta forma, um usuário pode partir de qualquer bairro e, por meio das rotas, acessar qualquer outro bairro da cidade. O custo de uma solução é dado pela distância total das rotas, o tempo total de deslocamento dos usuários, considerando os tempos de transbordo, e dos custos operacionais da frota necessária para realizar as viagens previstas nas rotas. Sendo assim, projetar uma rede de transporte público de ônibus envolve quatro etapas principais:

1. desenhar a rede de transporte, ou seja, a definir a trajetória de cada linha de ônibus (também denominada rota) que comporá a solução;
2. estabelecer a tabela de horários de partida dos ônibus de cada rota;
3. programar a frota (ônibus) mínima que atuará nas diferentes rotas e
4. programar os motoristas que conduzirão a frota em operação (CEDER; WILSON, 1986).

A primeira fase é a definição das rotas de ônibus que tem como base a matriz de demanda por viagens entre os diferentes bairros da cidade. A etapa seguinte envolve a determinação adequada da frequência de partida dos ônibus para cada rota, tendo em vista atender à quantidade de usuários prevista na matriz de demanda por viagem em cada período e tipo de dia: dia útil, sábado e domingo. A terceira etapa trata da programação da frota necessária para executar as viagens previstas em cada rota, com base na tabela de horários. Nesta etapa também são definidos os custos fixos e variáveis incorridos na operação dos veículos. Na quarta e última fase, são definidas as programações diárias das tripulações que conduzirão a frota, assim como a sua escala mensal de trabalho (FAN; MACHEMEHL, 2004). A quarta fase apresentada está fora do escopo deste trabalho.

É altamente desejável otimizar todos os quatro estágios simultaneamente a fim de explorar a capacidade do sistema e maximizar o desempenho e a eficiência do mesmo. No entanto, esta é uma abordagem extremamente complexa e de difícil resolução devido ao grande número de variáveis e de restrições. Por isso, o problema normalmente é tratado separadamente, sendo que o resultado de uma etapa é utilizado como dado de entrada para a etapa subsequente. Nas últimas décadas, houve um grande esforço para automatizar e otimizar as quatro etapas do problema, a fim de proporcionar uma operação mais eficiente, com horários controláveis e programações mais economicamente para ambas as partes envolvidas no sistema: o operador e o usuário (BHTRANS, 2013).

Os trabalhos da literatura relacionados com o Problema de Projeto de Rede de Transporte Público (PPRTP) por ônibus podem ser divididos em quatro classes: 1) os modelos que otimizam apenas a configuração da rede das rotas dos ônibus, 2) os modelos que primeiramente otimizam a configuração das rotas de ônibus e, em seguida, determinam as frequências ótimas de partida, 3) os modelos que otimizam simultaneamente a rede de rotas e suas respectivas frequências, e 4) os modelos que sequencialmente otimizam a configuração das rotas, as frequências das partidas e a programação dos ônibus.

Pode-se concluir, a partir de uma revisão na literatura, que a função objetivo utilizada na maior parte dos trabalhos inclui duas parcelas principais: o custo dos usuários e o custo do operador do sistema. A maioria dos trabalhos utiliza o tempo total de viagem dos passageiros como o custo do usuário, incluindo o tempo de espera, tempo de viagem, o tempo e o número de transferências entre as linhas. A quilometragem percorrida pelos ônibus ou o tempo de utilização do veículo, juntamente com o número de veículos reflete o custo do operador. As restrições mais encontradas nos modelos incluem a viabilidade da frequência de partida, o fator de carregamento do veículo, o tamanho (distância) de cada rota e a frota disponível. Como método de resolução do problema, fica clara a larga utilização dos Algoritmos Genéticos. Uma justificativa para a utilização dos Algoritmos Genéticos (AGs) pode ser pela representação de uma solução para o problema, que consiste em um conjunto de rotas e suas frequências (AFANDIZADEH; KHAKSAR; KALANTARI, 2013).

1.1 Justificativa

Visto que o transporte público é uma parte essencial na vida urbana, uma melhoria do mesmo seria de grande contribuição para a população das cidades já que grande parte dos brasileiros não possui veículo próprio, ou não dispõem de recursos para utilizar, de forma intensiva seus veículos. Uma melhoria no sistema também seria interessante para que as pessoas considerassem a possibilidade de migrar do transporte individual para o transporte público, diminuindo assim a poluição na cidade e o tráfego de veículos pelas vias.

1.2 Objetivos

Este trabalho tem como objetivo implementar dois algoritmos de otimização para definir o conjunto de linhas de um sistema de transporte público por ônibus urbano e determinar a frota em operação neste sistema. Os objetivos a serem otimizados pelo algoritmo visam a melhoria da qualidade do transporte para os usuários e também a diminuição dos custos com a frota para as empresas operadoras.

No Desenho de Redes de Transporte (*Transit Network Design*), além de definir a trajetória das linhas a serem operadas, também devem ser calculados os intervalos de partida dos veículos para cada linha. Finalmente, a partir da tabela de horários de cada linha, é definida a frota necessária para o cumprimento de todas as viagens programadas.

Para os usuários, busca-se a redução do tempo de espera do passageiro nos pontos de embarque e o tempo total da viagem. Por outro lado, para o operador, procura-se minimizar a frota e a sua distância total percorrida, necessária para atender a toda a demanda da cidade.

1.3 Organização do Trabalho

O restante do trabalho está organizado da seguinte forma: no Capítulo 2 é apresentada uma revisão da bibliografia com a descrição do problema e abordagens de solução dos principais trabalhos da literatura, com maior foco na estrutura que compõe um PPRTP. No Capítulo 3 estão descritos os algoritmos desenvolvidos neste trabalho para a resolução do problema apresentado, assim como a descrição das funções que os compõem. No Capítulo 4 estão as soluções geradas pelos algoritmos desenvolvidos, a comparação dessas soluções com outros trabalhos da literatura e as soluções geradas para a cidade de Ouro Preto, assim como discussões a respeito destas soluções. Por fim, no Capítulo 5.1 são apresentadas as conclusões deste trabalho e os trabalhos que podem ser realizados futuramente.

2 Revisão Bibliográfica

2.1 Introdução

Neste capítulo são apresentadas as principais etapas de um Problema de Projeto de Rede de Transporte Público (PPRTP) por ônibus a partir de diferentes trabalhos da literatura. Assim, são destacados os pontos cruciais de um método de resolução do problema, tendo em vista desenvolver um trabalho bem fundamentado e evitar desperdiçar tempo em questões menos relevantes.

2.1.1 Representação de uma Solução

Uma solução para o PPRTP é definida por: *i*) um conjunto de rotas ou linhas de ônibus e *ii*) a frota de ônibus urbano que deve percorrer as linhas, tal que toda a demanda prevista seja atendida. A cidade, ou região estudada, é dada por uma *rede* $R(N, A)$ composta pelos nós em N e os arcos em A representando os bairros e suas ligações respectivamente. Além da estrutura viária, ou seja, da rede R , deve ser informada a Matriz Origem-Destino (O/D), que contém a quantidade de passageiros que devem ser transportados entre cada um dos bairros considerados, como mostrado na tabela 3.1. Esta mesma matriz é também utilizada para determinar a frequência de saída de ônibus em cada uma das linhas.. A rede $R(N, A)$ também pode ser denominada um *grafo* e neste caso será denotada por $G(N, A)$.

2.1.2 Função Objetivo

A Função Objetivo (FO) é considerada pela maioria dos autores como multiobjetivo já que o problema visa minimizar tanto o custo do operador das rotas quanto o custo dos usuários, sendo estes dois, objetivos conflitantes entre si, conforme mencionado por [Fan e Machemehl \(2006 apud ARBEX; CUNHA, 2015\)](#).

Problemas do tipo multiobjetivo podem ser resolvidos de diversas formas. As mais comuns encontradas na literatura são através de modelos de otimização de Pareto, como em [Blum e Mathew \(2010 apud ARBEX; CUNHA, 2015\)](#), modelos de Soma Ponderada (*weighted sum*), como em [Afandizadeh, Khaksar e Kalantari \(2013\)](#) e [Arbex e Cunha \(2015\)](#). Neste trabalho foi adotada a Soma Ponderada na FO para representar o problema, que contempla, em uma única FO, os diferentes interesses envolvidos.

2.1.3 Restrições

Para que uma solução seja aplicável a uma dada rede, determinadas restrições devem ser atendidas, garantindo a viabilidade da solução encontrada. As restrições, em geral têm como função garantir que uma dada solução encontrada seja viável e esteja de acordo com a rede na qual a mesma está sendo aplicada. De acordo com [Chew e Lee \(2012 apud ARBEX; CUNHA, 2015\)](#), o conjunto de rotas escolhidas para compor uma solução somente é viável se atender às seguintes restrições:

- **Todos os nós da rede devem ser atendidos:** Para que uma solução seja viável, todo nó presente na rede deve ser atendido por pelo menos uma rota para que seja garantido que toda região possua uma linha que passe ou chegue a ela. Na [Figura 2.1](#) é mostrado um exemplo de uma solução que não respeita esta restrição.
- **Conectividade entre as rotas:** Toda rota de uma solução deve se conectar a uma ou mais rotas de forma a garantir que, partindo de um ponto da rede, seja possível chegar a qualquer outro ponto, independentemente do número de transferências que devem ser feitas. A [Figura 2.2](#) apresenta um exemplo de uma solução que não satisfaz a esta restrição.
- **A mesma rota não pode aparecer mais de uma vez em uma solução:** Isso pode ocorrer durante o processo de otimização. Devem ser tomadas precauções para eliminar a duplicidade de rotas, caso isto ocorra.
- **Nós devem ser únicos em uma rota, porém podem se repetir em diferentes rotas:** Um nó deve aparecer somente uma vez em uma rota, mas ele pode aparecer em diferentes rotas de uma solução, o que certamente ocorrerá em uma solução viável se for levada em consideração a restrição de Conectividade entre as rotas, apresentada anteriormente. Esta restrição pode ser flexibilizada em modelos mais complexos pois a mesma inibe a ocorrência de rotas conhecidas como circulares, onde o ponto final é o mesmo que o inicial de uma rota, como visto em [Kuo \(2014 apud ARBEX; CUNHA, 2015\)](#).

De acordo com [Kepaptsoglou e Karlaftis \(2009 apud ARBEX; CUNHA, 2015\)](#) as restrições podem ser divididas em dois tipos, as restrições de desempenho e as restrições de recurso. As restrições de desempenho com maior aplicabilidade a este trabalho são:

- **Viabilidade da frequência de cada rota da solução:** As frequências não podem ser muito baixas pois quanto maior o intervalo entre as partidas de ônibus, maior o tempo de espera do usuário. O intervalo também não podem ser muito reduzido pois considerando as paradas nos pontos intermediários das linhas e em semáforos, pode ocorrer o efeito de sanfonamento dos ônibus. No efeito de sanfonamento, ou comboio de dois ou mais

ônibus passam em um espaço de tempo muito curto fazendo com que o primeiro ônibus carregue a maioria dos passageiros, em detrimento dos outros que seguem no comboio, que podem viajar com capacidade ociosa.

- **Capacidade do veículo:** Na literatura é utilizado o termo *load factor*, ou fator de carregamento, que é a razão entre a quantidade de passageiros carregados e o número de assentos no ônibus. O valor do *load factor* não pode ultrapassar o limite determinado em nenhum trecho de nenhuma rota. O valor padrão encontrado na literatura foi de 1.25, de acordo com Mandl (1980) e Afandizadeh, Khaksar e Kalantari (2013).
- **Tamanho mínimo e máximo da rota:** As rotas de uma solução devem obedecer às restrições de tamanho impostas a elas para que certos tipos de problemas sejam evitados. Em rotas muito longas o controle do intervalo entre os veículos torna-se difícil, visto que quanto maior a rota, maior o número de semáforos e de pontos de parada de ônibus dentro da mesma. Em rotas muito curtas pode ser que a demanda atendida pela rota não seja rentável e o número de transferências necessárias para ir de um nó a outro seja elevado, deixando assim empresa e usuários insatisfeitos. Devido a isso, é estabelecido um limite inferior e um limite superior para o número de nós ou a distância total percorrida pelas rotas que compõem as soluções.
- **Área de cobertura:** É dada pela porcentagem da demanda que tem acesso ao transporte público levando em consideração a distância até o ponto de ônibus mais próximo. Guihaire e Hao (2008 apud ARBEX; CUNHA, 2015) consideram que pessoas vivendo a até 400-500 metros de um ponto de ônibus se enquadram nessa porcentagem de cobertura de área.

A cobertura de área indica apenas as pessoas que, de acordo com o que foi apresentado anteriormente, têm acesso ao transporte público. Neste trabalho a demanda fornecida como dado de entrada é tratada por completo, visto que os pontos de ônibus dentro de uma região representada por um nó não são analisados, mas fornecidos como dados de entrada fixos. Portanto a área de cobertura é predeterminada e tida como cem por cento dos dados de entrada.

- **Ligações diretas:** Dado um nó origem e um nó destino deve ser possível o usuário chegar de um nó ao outro andando o mínimo possível até um ponto de ônibus, percorrendo o caminho mais direto possível entre os nós e realizando um menor número de transferências, se possível nenhuma. Neste trabalho não serão consideradas as distâncias até os pontos de ônibus visto que todos os pontos de uma determinada região são representados por um único nó. A quantidade de transferências tem um custo elevado na FO, torna-se um fator importante a ser minimizado no processo de otimização.

- **Atendimento da demanda:** Neste trabalho é considerado que se o usuário necessitar de mais de duas transferências, é assumido que o mesmo terá preferência por outro meio de transporte, assim como em [Guihaire e Hao \(2008 apud ARBEX; CUNHA, 2015\)](#).

O segundo tipo de restrições descrito por [Kepaptsoglou e Karlaftis \(2009 apud ARBEX; CUNHA, 2015\)](#) são as restrições de recursos, ou seja, o tamanho da frota disponível e o orçamento da empresa responsável pela aquisição da frota. O orçamento é uma restrição pois limita a quantidade de ônibus que poderão ser adicionados à frota após a remodelagem da rede. Ou seja, a frota é a principal variável a ser reduzida visando o lucro do operador, mas nunca se esquecendo da demanda que deve ser atendida.

As restrições apresentadas acima são aquelas que aparecem com maior frequência na literatura. Entretanto, em [Guihaire e Hao \(2008 apud ARBEX; CUNHA, 2015\)](#) são mencionadas algumas outras restrições que influenciam significativamente no desenho de um novo sistema de rotas em uma região. As principais restrições citadas são:

- **Histórico da rede:** Se existir uma rede em vigência, pode ser que algumas rotas tenham que ser mantidas por alguma razão política ou mesmo por algumas rotas darem um grande lucro para a empresa responsável pelo transporte público, ou seja, a rede vigente deve ser analisada para que a partir das restrições estabelecidas pela mesma, sejam geradas soluções que atendam às exigências impostas.

Outro fator importante são as mudanças impostas no redimensionamento proposto para a rede. Os usuários já estão acostumados a uma determinada rota, a um determinado ponto de ônibus e a um determinado horário. Uma mudança drástica e repentina pode acarretar na insatisfação dos usuários e piorando assim a percepção de qualidade do transporte para quem faz uso dele [Jirón \(2012 apud ARBEX; CUNHA, 2015\)](#). Desta forma, as mudanças devem ser graduais e informadas aos usuários tendo em vista evitar descontentamentos com o transporte público.

- **Número de linhas:** Uma rede com um grande número de linhas geralmente está associado à existência de uma alta proporção de linhas com tempos de espera elevados. Cabe ressaltar que não existe uma definição do que vem a ser um número elevado de linhas, pois é um conceito relativo à extensão da área urbanizada da cidade ([ARBEX; CUNHA, 2015](#)).

Concluindo, as restrições mesmo sendo em grande número, são importantes para a criação de uma solução satisfatória para todos os envolvidos no sistema e viável de acordo com as limitações da infraestrutura de uma cidade.

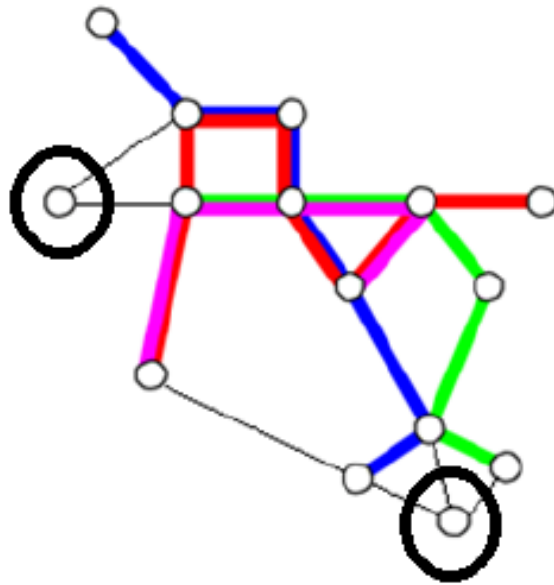


Figura 2.1 – Rede Inviável pois nem todos os nós são atendidos. Os nós destacados por um círculo maior não são atendidos.

Fonte: Figura retirada de Arbex e Cunha (2015)

2.1.4 Variáveis de Decisão

Em Kepaptsoglou e Karlaftis (2009 apud ARBEX; CUNHA, 2015) são apresentadas como principais variáveis de decisão para o PPRTP o traçado das rotas, a frequência de operação, o preço da passagem de ônibus, a localização dos pontos de ônibus e o tamanho dos veículos.

Em um cenário ideal, deveriam ser consideradas todas as cinco variáveis de decisão apresentadas. Porém, devido à complexidade do problema, os trabalhos estudados optam por considerar apenas duas, sendo elas a frequência de operação e o traçado das rotas. Conseqüentemente é obtida a frota necessária para realizar as viagens.

Para que o preço da passagem fosse considerado no trabalho, alterações deveriam ser feitas no domínio das demais variáveis, fazendo com que valores monetários fossem aplicados a elas. Outra opção seria transformar os valores das passagens para um valor numérico significativo dentro da FO, o que exigiria um estudo mais detalhado deste tema. Ambas as formas estão fora do escopo deste trabalho, por isso o preço da passagem não será levado em consideração.

A localização dos pontos dentro de uma determinada região, como discutido anteriormente, não será levada em consideração, visto que todos os pontos dentro dessa região serão representados por um único nó na rede, simplificando assim a análise da mesma.

O tamanho do veículo diz respeito a qual tipo de ônibus será utilizado em qual rota: padrão, articulado ou bi-articulado, por exemplo. De fato, cidades grandes têm à sua disposição uma variedade de tamanhos de veículos que podem ser usados. Porém, para tornar a análise mais

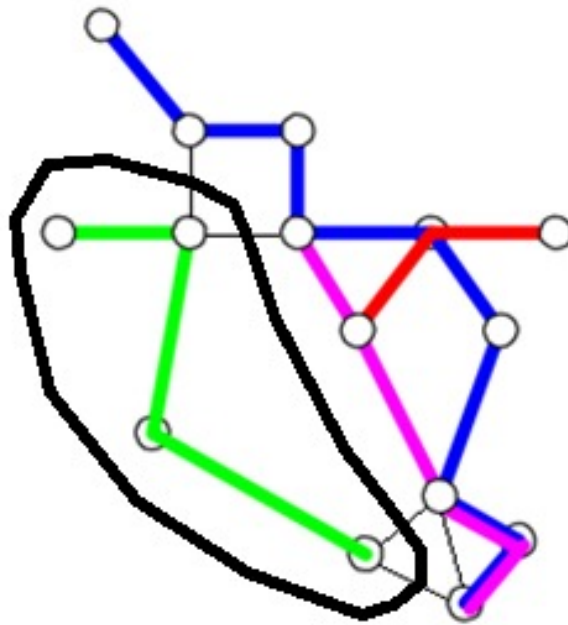


Figura 2.2 – Rede Inviável pois as rotas não estão conectadas. A rota destacada à esquerda da figura está isolada do resto da rede.

Fonte: Figura retirada de Arbex e Cunha (2015)

simples, serão considerados ônibus de um mesmo tamanho, tamanho este informado através dos dados de entrada do problema.

Analisando a literatura e o problema estudado, as variáveis de decisão descritas em Kepaptsoglou e Karlaftis (2009) e utilizadas neste trabalho foram: *i*) o traçado das rotas, *ii*) a frequência de operação, e *iii*) a frota necessária para cumprir as viagens. Para que o valor da passagem não fosse completamente ignorado, também foi avaliada a quantidade de transferências para os usuários se deslocarem entre todos os pares de pontos da rede, visando minimizar esta variável.

2.1.5 Dados de Entrada

De acordo como Arbex e Cunha (2015) os dados de entrada mais comumente encontrados em trabalhos relacionados a este problema são a demanda de transporte público, a demanda de transporte individual, a infraestrutura da cidade assim como o tempo de viagem entre suas regiões, e a rede de transporte público já existente. Em seu texto também é dito que a demanda de uma rede pode ser fixa ou variável. Em modelos com demanda variável, a demanda total de uma rede é dividida em demanda de transporte público e demanda de transporte individual, baseando seus valores na qualidade do transporte público. Assim, os dados de entrada de um modelo desse tipo necessita de ambas as demandas. Neste trabalho foi fornecida a demanda para a rede de transporte público, visando atender toda ela com as linhas e frequências de partidas contidas na

solução.

A infraestrutura de uma cidade é fornecida como dado de entrada na representação de uma rede onde as regiões da cidade são representadas pelos nós e as conexões entre as regiões, ou seja, as vias que conectam uma região à outra serão representadas como as arestas que ligam os nós. Caso existam faixas específicas para ônibus na cidade, estas também devem ser fornecidas como dados de entrada. Neste trabalho faixas de ônibus exclusivas são tratadas como conexões normais entre as regiões.

O tempo de viagem de um nó a outro deve também ser fornecido como um dado de entrada. Esse tempo de viagem é tratado como o custo das arestas da rede. Neste trabalho são fornecidas as distâncias entre os nós e a velocidade média dos ônibus na cidade analisada. As velocidades em diferentes rotas podem variar, porém neste trabalho foi utilizada a velocidade média em cada rota.

Para modelos mais complexos, também pode ser inserido o conjunto de rotas que já existe na cidade com o intuito de obter-se informações que possam auxiliar na construção de uma nova rede. Neste trabalho a análise deste conjunto de rotas foi realizada manualmente, já que a mesma irá auxiliar também na construção do grafo de representação.

Concluindo, os dados de entrada deste modelo são a representação da cidade, as demandas entre as regiões, as distâncias entre as regiões e a velocidade média dos ônibus na cidade.

2.1.6 Métodos de Solução

O PPRTP é um problema de otimização *NP-hard* complexo e difícil de resolver, sendo que formulações de programação matemática tradicionais não podem ser aplicados de forma eficiente (BAAJ; MAHMASSANI, 1991 apud ARBEX; CUNHA, 2015). Além disso, outros fatores podem contribuir para o aumento da complexidade do PPRTP como por exemplo o fato dele ser um problema multiobjetivo combinatorial. O PPRTP também é um problema não-linear, visto que adicionar vários ônibus a uma solução pode levar ao aumento do tempo de viagem decorrente de possíveis engarrafamentos gerados pela grande quantidade de veículos nas vias. Este fato pode aumentar o custo da solução.

Por se tratar de um problema combinatorial, a maioria dos trabalhos recentes utilizam heurísticas para resolvê-lo. Em Kepaptsoglou e Karlaftis (2009) essas heurísticas são divididas em dois grupos diferentes. No primeiro grupo é criado um banco de rotas fixo que contém diferentes rotas para diferentes origens e destinos e cada solução será formada por um subconjunto dessas rotas, como em Afandizadeh, Khaksar e Kalantari (2013). No segundo grupo, as rotas são criadas baseando-se em heurísticas e metaheurísticas e melhorias podem ser obtidas nessas rotas durante a execução do programa, como em Yang, Yu e Cheng (2007).

Em trabalhos do primeiro grupo, deve-se criar o banco de rotas no início do processo, respeitando as condições aplicadas às rotas descritas na seção 2.1.3. Os algoritmos mais utili-

zados para a construção desse banco de rotas são o de *caminho mínimo* de Dijkstra (1959) e o dos *k-caminhos mínimos* de Yen (1971). Já em trabalhos do segundo grupo, a utilização de metaheurísticas é feita diretamente na criação das rotas, como apresentado em Yang, Yu e Cheng (2007).

Em relação à escolha das rotas para formar as soluções iniciais, é muito utilizado o Algoritmo Genético quando se está trabalhando com o banco de rotas. Exemplos desta abordagem são os trabalhos de Chew e Lee (2012) e Afandizadeh, Khaksar e Kalantari (2013).

Após a formação das soluções iniciais, as demandas devem ser distribuídas entre as rotas que as compõem. Para isso, de acordo com Baaj e Mahmassani (1991 apud ARBEX; CUNHA, 2015) primeiramente é verificado se os nós origem e destino possuem uma conexão direta, ou seja, se essa viagem pode ocorrer sem a necessidade de transferências. Caso não seja possível, é verificado se é possível esse transporte com uma transferência, e em seguida com duas transferências até o limite permitido.

2.2 Aplicação a Problemas Reais

Nesta seção são apresentados estudos que fizeram a aplicação dos métodos de solução de um PPRTP em uma cidade real. A análise feita em um problema real pode mostrar características que não são perceptíveis ao analisar uma rede gerada aleatoriamente, principalmente por não contar com uma referência anterior com a qual comparar. Alguns trabalhos que realizaram aplicações reais são Mandl (1980) que realizou a aplicação em uma região da Suíça e Afandizadeh, Khaksar e Kalantari (2013) que por sua vez realizou a aplicação para a cidade de Mashad, no Irã.

A aplicação feita em uma rede de maior porte é encontrada no trabalho de Ciaffi, Cipriani e Petrelli (2012), que utiliza o método de banco de rotas para a cidade de Roma, na Itália. Uma comparação entre a rede encontrada pelos autores e a rede vigente à época apresentou um aumento de 3,7% no número de transferências em relação à rede em operação. Apresentou também reduções de 60% no número de linhas e 16,6% no tempo de viagem considerando os extremos das rotas. Com esses valores pode-se observar que a diferença entre o tamanho da rede em vigência no ano do estudo e da rede apresentada pelo método é significativa. Apesar de apresentar uma grande redução no número de linhas e uma redução considerável no tempo viagem, o número de transferências teve um aumento considerável. Portanto, em alguns casos podendo causar uma insatisfação dos usuários, uma vez que o custo é um fator de muito peso.

Outra aplicação do método é feita por Afandizadeh, Khaksar e Kalantari (2013), que realizou o estudo na cidade de Mashad, no Irã. A cidade conta com a população de cerca de 3,001 milhões de habitantes. Seus resultados apresentaram uma redução de 6% no número de linhas, 8,5% na frota e 46% no tempo de viagem.

Os trabalhos revisados realizaram a aplicação de seus métodos a uma rede menor citada

anteriormente, presente em Mandl (1980). Esta rede é considerada *benchmark* para o problema PRPT. A rede é composta por 15 nós, 42 duas arestas, com uma demanda total de 15.570 viagens. Esta rede foi utilizada para a validação do algoritmo desenvolvido neste trabalho.

2.3 Conclusão da Revisão

Com a revisão da literatura, foi possível perceber que grande parte dos trabalhos não mostram, ou não foram calculados, os valores da frota necessária para suprir as demandas apresentadas. A frota necessária é uma informação muito importante para não ser apresentada, dado que grande parte do custo do operador é proveniente da quantidade de ônibus que terão que estar em circulação.

Outro ponto muito importante que deve ser levado em consideração ao se trabalhar com um PPRPT são as restrições que serão aplicadas ao problema, pois elas que irão delimitar características importantes da região para que a rede gerada possa ser a mais otimizada possível. Portanto, grande atenção deve ser dada às restrições e seus valores, como por exemplo no número de linhas e no tamanho da rota.

3 Desenvolvimento

Neste capítulo é apresentada a metodologia adotada no trabalho assim como as etapas de desenvolvimento do mesmo. São descritos os algoritmos desenvolvidos para a resolução do problema PPRTP, assim como os estudos realizados para a aplicação dos algoritmos em um problema real. O primeiro algoritmo se baseia na literatura, com alguns métodos bastante conhecidos para este tipo de problema. O segundo algoritmo se baseia em sugestões da banca de qualificação da monografia.

3.1 Metodologia

A metodologia empregada neste trabalho tem natureza quantitativa, utilizando como dados de entrada para o método de resolução do problema as informações da operação da rede de transporte público de Ouro Preto-MG, no período de 2015. Os dados inerentes às soluções obtidas pelo método de resolução, assim como a solução adotada pelo órgão gestor à época, permitem mensurar as mesmas, e desta forma, realizar uma comparação numérica de eficiência entre elas. Este trabalho se caracteriza como uma pesquisa aplicada, uma vez que utiliza um método de otimização para conduzir um *estudo de caso* do sistema de transporte público do município de Ouro Preto-MG.

O procedimento adotado para a resolução do PPRTP consiste em implementar dois Algoritmos Genéticos (AGs), um baseado no trabalho de [Arbex e Cunha \(2015\)](#), com modificações que permitem aprimorar este algoritmo, para otimizar uma rede de transporte público, e outro baseado em sugestões da banca de qualificação, como dito anteriormente. Os AGs implementados foram calibrados com dados *benchmark* da literatura e posteriormente foram aplicados aos mesmos dados de entrada da situação do município de Ouro Preto. As soluções obtidas pelos AGs foram comparadas com as soluções adotadas na prática, verificando assim se um método de otimização é aplicável à prática e se eles foram capazes de melhorar a operação do sistema de transporte público no caso estudado.

3.2 Algoritmo 1

Nesta seção é descrito o primeiro algoritmo utilizado para a resolução do PPRTP. Seu desenvolvimento foi baseado no Algoritmo Genético descrito por [Arbex e Cunha \(2015\)](#) com a adaptação dos pseudo-códigos apresentados pelo autor.

O método utilizado para a resolução do problema consiste no modelo de construção de um *banco de rotas* como descrito na subseção 2.1.6. O mesmo consiste na construção de um

conjunto de diferentes rotas para que, a partir dele sejam selecionadas as rotas que irão compor as soluções do problema. Neste trabalho, a construção do banco de rotas é feita a partir do algoritmo dos k-caminhos mínimos (YEN, 1971), aplicado em todos os pares possíveis de nós. Após a construção do banco de rotas é feita a geração da população inicial, que são as N primeiras soluções viáveis. Em seguida ocorre a aplicação do AG partindo destas soluções.

O Algoritmo Genético desenvolvido neste trabalho é dividido em três etapas: *i*) etapa de cruzamento, *ii*) etapa de mutação e *iii*) etapa de seleção dos sobreviventes para a próxima geração. Todas estas etapas são repetidas em um determinado número de gerações, na ordem em que foram apresentadas, até que o valor da melhor FO dentre as atuais soluções se mantenha constante por um dado número de gerações, como mostrado na Figura 3.1.

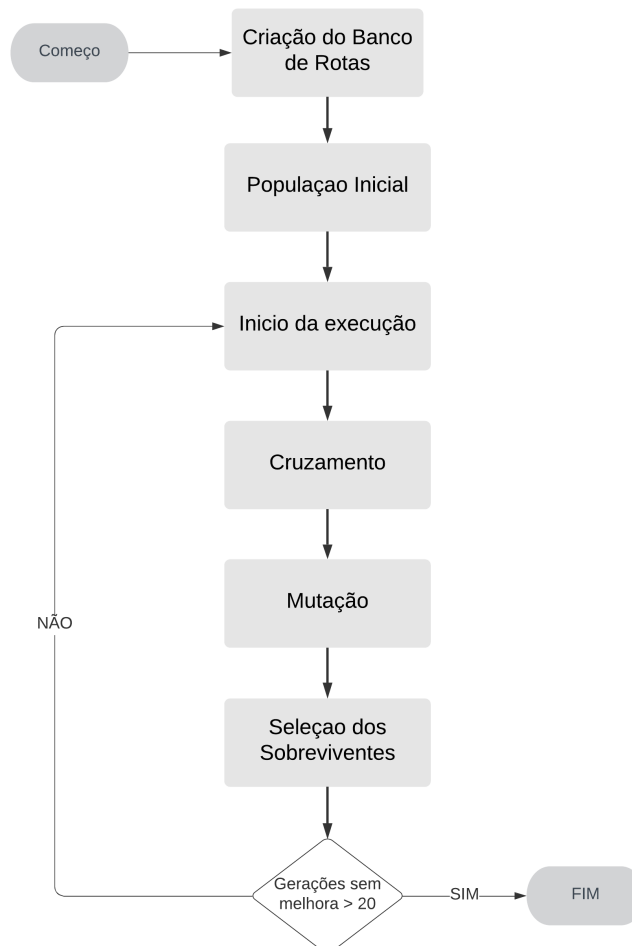


Figura 3.1 – Fluxograma do Algoritmo 1

Nas seguintes seções serão apresentadas mais detalhadamente as etapas que compõem o AG para a resolução do problema.

3.2.1 Banco de Rotas

Para a geração do banco de rotas foi desenvolvida uma função que recebe como entrada o grafo que representa a cidade, o tamanho mínimo de uma rota, uma taxa que define o tamanho máximo das rotas, o tempo máximo de uma rota e um vetor para o armazenamento das rotas, sendo esse o banco de rotas. Essa função é representada no Algoritmo 1.

Para cada nó da rede, são gerados k caminhos mínimos entre ele e todos os demais nós da rede. Esses caminhos são armazenados em um vetor, ordenado pelo tamanho das rotas, para que sejam selecionados como rotas possíveis de participar da solução ou não. Começando pelo primeiro elemento do vetor, é feita a verificação de tamanho mínimo de rota e tempo máximo. Caso a rota satisfaça as duas condições, ela é adicionada ao banco de rotas, caso contrário ela é descartada. As demais rotas são submetidas ao mesmo processo com acréscimo de uma condição: o tamanho da rota é limitado superiormente pelo tamanho da menor rota entre os vértices, multiplicado pela taxa dada como parâmetro de entrada. A criação das rotas é feita até que todos os pares possíveis de nós tenham sido considerados.

Algoritmo 1: Criação do Banco de Rotas

Entrada: Grafo G ; Limite inferior de tamanho LI ; Taxa para tamanho máximo TX ;

Limite Superior de tempo LS ; Vetor do Banco de Rotas BR

Saída: Quantidade de rotas no vetor BR .

```

1 begin
2   Cria um vetor Rotas
3   for cada  $V_i$  em  $G$  do
4     for cada  $V_j$  diferente de  $V_i$  em  $G$  do
5        $Rotas \leftarrow KShortestPaths(V_i, V_j, G)$ 
6       if  $Rotas[0].size > LI \ \& \ Rotas[0].time < LS$  then
7          $BR \leftarrow Rotas[0]$ 
8       end
9       for cada Rota em  $Rotas$  do
10        if  $Rota.size > LI \ \& \ Rota.time < LS \ \& \ Rota.size <$ 
11          $Rotas[0].size \times TX$  then
12            $BR \leftarrow Rota$ 
13         end
14       end
15     end
16   return  $BR.size$ 
17 end

```

3.2.2 População Inicial

A população inicial é composta pelas primeiras n soluções construídas pelo AG, sendo assim nosso passo inicial no Algoritmo Genético. A função de construção da população inicial recebe como entrada o banco de rotas previamente construído, o valor n , o tamanho mínimo das rotas, o grafo que representa a rede de transporte e um vetor para o armazenamento das soluções. A partir desses dados é executado o Algoritmo 2, apresentado abaixo.

Algoritmo 2: Construção da População Inicial

Entrada: Quantidade de soluções N , Banco de Rotas BR , Tamanho mínimo $TamMin$, Grafo G , Vetor de Soluções $VetSol$.

```

1 begin
2   Cria uma lista NosAtendidos
3   Cria uma Solução solAux
4   for  $i \leftarrow 0$  até  $N$  do
5     while solAux não for viável do
6       if  $(G.nos.size - NosAtendidos.size) > 1$  then
7         origem  $\leftarrow$  nó não atendido
8         destino  $\leftarrow$  nó não atendido
9         solAux  $\leftarrow$  SeleccionaRota(origem, destino,  $BR$ )
10      end
11     else
12       NoAux  $\leftarrow$  nó não atendido
13       solAux  $\leftarrow$  SeleccionaRota(NoAux,  $BR$ )
14     end
15     end
16     while solAux.rotas.size  $<$   $TamMin$  do
17       CalculaCarregamento(solAux)
18       arestaAux  $\leftarrow$  aresta de maior carregamento
19       solAux  $\leftarrow$  SeleccionaRota(arestaAux,  $BR$ )
20     end
21     vetSol  $\leftarrow$  solAux
22 end
23 end

```

A partir do grafo G , são identificados os nós presentes na rede. A criação de uma solução se dá acrescentando rotas até que se obtenha uma solução viável. Assim, são selecionados aleatoriamente dois nós que ainda não foram atendidos pela solução parcial. A partir desses nós é escolhida uma rota dentro do banco de rotas que conecte estes nós pelos seus pontos extremos. Este processo se repete até que todos os nós sejam atendidos pelas rotas na solução. Se, ao final, apenas um nó não for atendido pelas rotas que compõem a solução, é selecionada uma rota

que passe por este nó. A cada inserção de uma nova rota, é feita a verificação da viabilidade da solução.

Na maioria das vezes, após uma solução ser considerada viável e atingir todos os nós, a mesma não possui o número mínimo de rotas determinado. Para resolver este problema, primeiramente é calculada a distribuição da demanda pelas rotas, que será um fator importante para a decisão sobre o tipo de rota a ser adicionada à solução. Após o cálculo de distribuição, analisa-se qual o trecho de maior carregamento no grafo. Dentro de um sub-conjunto de rotas que passam por esse trecho, é feita uma seleção aleatória da rota que será adicionada à solução. Este processo é repetido até que o número mínimo de rotas em uma solução seja atingido. Posteriormente, a solução é adicionada ao vetor de soluções. Todo o processo é repetido até que sejam obtidas as n soluções da população inicial.

3.2.3 Cruzamento

Para que ocorra o cruzamento, são conhecidos: *i*) o vetor de soluções atuais $VetSol$, que na primeira execução do AG é o vetor com a população inicial, *ii*) o grafo G que representa a rede *iii*) o banco de rotas BR gerado anteriormente e, *iv*) a porcentagem de cruzamento $PercentCruz$ usada para determinar o número de filhos a serem gerados. Além desses parâmetros, é fornecida também a porcentagem de soluções pertencentes ao conjunto elite, $PercentElite$. Esta porcentagem determina a quantidade das melhores soluções, dentro da nova população, que irá compor o conjunto elite.

O cruzamento funciona da seguinte forma: são selecionados dois pais, um do conjunto elite e o outro fora do conjunto elite para que ocorra o cruzamento. Ambos são selecionados de forma pseudo-aleatória através do *Método de Seleção de Torneio* (TALBI, 2009), sendo que uma certa quantidade de soluções é selecionada de forma completamente aleatória e dentre as selecionadas, aquela com menor valor da FO é escolhida para o cruzamento.

Após a escolha dos pais, inicia de fato o procedimento do cruzamento. Baseando-se na quantidade de rotas de cada pai é sorteado aleatoriamente um *ponto de corte* onde as soluções serão divididas, como ilustrado na Figura 3.2. Ambos os pais sofrem o corte no mesmo ponto, iniciando a contagem a partir do primeiro gene da solução. Cada metade de cada solução é salva em uma lista de rotas auxiliar, que depois serão unidas para formarem novas soluções.

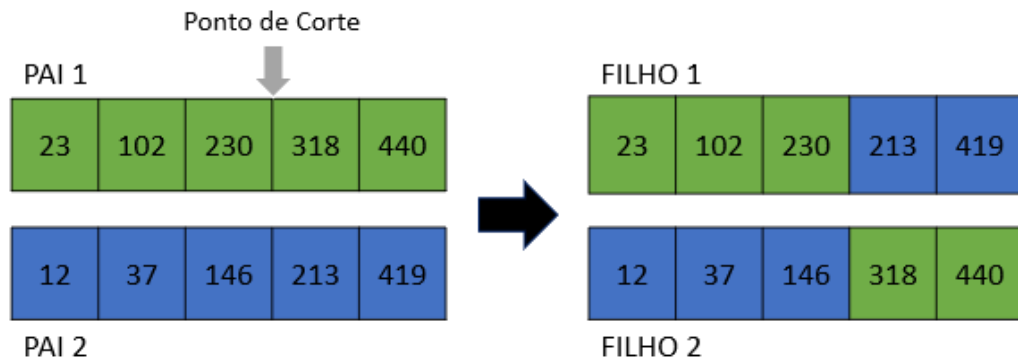


Figura 3.2 – Representação do processo de Cruzamento.

A primeira parte do primeiro pai é concatenada com a segunda parte do segundo pai, formando assim um novo indivíduo. O mesmo ocorre com a primeira parte do segundo pai e a segunda parte do primeiro pai. Podemos observar melhor este processo analisando a Figura 3.2. Os dois novos indivíduos são submetidos ao teste de validação da solução. Se ambos os filhos gerados forem viáveis, eles são salvos em uma lista de soluções construída apenas para os filhos. Caso apenas um filho gerado seja viável, o mesmo é salvo na lista de filhos e um novo ponto de corte é escolhido. O processo de seleção de pontos de corte ocorre até que dois filhos viáveis sejam gerados por esses pais ou os pontos de corte se esgotem. Caso isso aconteça, são selecionados novos pais a passarem pelo mesmo processo. Isso ocorre até o número de filhos gerados seja maior ou igual a $PercentCruz \times VetSol.size$.

Todo o processo do cruzamento descrito pode também ser visto no Algoritmo 3, apresentado abaixo em forma de pseudo-código.

Algoritmo 3: Cruzamento

Entrada: Vetor de Soluções $VetSol$, Grafo G , Banco de Rotas BR , Porcentagem de cruzamento $PercentCruz$, Porcentagem de soluções no conjunto elite $PercentElite$.

Saída: Lista de soluções $Filhos$ com as soluções geradas pela Cruzamento.

```

1 begin
2   Cria uma lista de soluções  $Filhos$ 
3   Cria uma lista de pontos de corte  $PontosPossiveis$ 
4   while  $Filhos.size < (PercentCruz \times VetSol.size)$  do
5      $Pai1 \leftarrow SelecionaSolucao(Elite)$ 
6      $Pai2 \leftarrow SelecionaSolucao(ForaElite)$ 
7      $FilhosGerados \leftarrow 0$ 
8      $PontoDeCorte \leftarrow 0$ 
9      $PontosTestados.clear$ 
10     $MenorPai \leftarrow Pai1$ 
11    if  $Pai1.size > Pai2.size$  then
12      |  $MenorPai \leftarrow Pai2$ 
13    end
14    while  $FilhosGerados < 2 \ \& \ PontosTestados < PontosPossiveis$  do
15      |  $PontoDeCorte \leftarrow SelecionaPontoN\~{a}oTestado(MenorPai,$ 
16      |    $PontosTestados)$ 
17      |  $Filho1 \leftarrow Pai1.begin \text{ at\~{e}} PontoDeCorte + PontoDeCorte \text{ at\~{e}} Pai2.end$ 
18      |  $Filho2 \leftarrow Pai2.begin \text{ at\~{e}} PontoDeCorte + PontoDeCorte \text{ at\~{e}} Pai1.end$ 
19      | if  $Filho1$  for viável then
20      |   |  $Filhos \leftarrow Filho1$ 
21      |   |  $FilhosGerados++$ 
22      |   end
23      |   if  $Filho2$  for viável then
24      |     |  $Filhos \leftarrow Filho2$ 
25      |     |  $FilhosGerados++$ 
26      |   end
27    end
28  return  $Filhos$ 
29 end

```

3.2.4 Mutação

Após o cruzamento, as novas soluções geradas, os filhos, são passados para a função de mutação para verificar se sofrerão uma mutação ou não. Esta função recebe como entrada: *i*) a lista de filhos gerados pelo cruzamento, *ii*) o percentual de mutação *PercentMut* que determina a porcentagem da população que deve sofrer a mutação, *iii*) a porcentagem de mutação dos genes *PercentGene*, e *iv*) o banco de rotas *BR*.

O método de seleção de torneio é utilizado para escolher os filhos que sofrerão mutação até que o número de soluções mutadas seja maior ou igual a $PercentMut \times Filhos.size$. Cada filho selecionado para a mutação passa pelo seguinte processo: para cada uma das rotas que compõem a solução, é gerado um número aleatório entre 0 e 100. Caso este número seja menor do que *PercentGene*, aquela rota sofrerá a mutação, caso contrário analisa-se a próxima rota como mostrado na figura 3.3.



Figura 3.3 – Representação do processo de Mutação.

Se uma rota for escolhida para sofrer uma mutação, outra rota com os mesmos extremos (nós inicial e final iguais) será colocada em seu lugar, para assim aumentar as chances da solução continuar viável após a mutação daquele gene. Rotas com os mesmos extremos que aquela escolhida para sofrer a mutação são testadas na solução até que se encontre uma solução viável, ou até que as rotas de mesmos extremos se esgotem. E então analisa-se o próximo gene, até que todos os genes daquela solução tenham sido verificados.

O Algoritmo 4 mostra em formato de pseudo-código o funcionamento da mutação como foi descrita acima.

Algoritmo 4: Mutação

Entrada: Vetor de Soluções $VetSol$, Grafo G , Banco de Rotas BR , Porcentagem de cruzamento $PercentCruz$, Porcentagem de soluções no conjunto elite $PercentElite$.

```

1 begin
2   Cria uma solução auxiliar  $SolucAux$ 
3   Cria uma variável para o número de soluções mutadas  $SolucMut = 0$ 
4   while  $SolucMut < (PercentMut \times Filhos.size)$  do
5      $SolucAux \leftarrow SelecSoluc(Filhos)$ 
6     for cada Rota pertencente à  $SolucAux$  do
7        $aux \leftarrow rand(0,100)$ 
8       if  $aux < PercentGene$  then
9          $Extremo1 \leftarrow Rota.begin$ 
10         $Extremo2 \leftarrow Rota.end$ 
11         $QtdJaTestadas \leftarrow 0$ 
12         $NaoMutada \leftarrow TRUE$ 
13        while  $NaoMutada \ \& \ (QtdJaTestadas <$ 
14           $Rotas(Extremo1, Extremo2).size)$  do
15           $RotaAux \leftarrow Rotas(Extremo1, Extremo2)$ 
16          Remove Rota de  $SolucAux$ 
17          Adiciona  $RotaAux$  em  $SolucAux$ 
18          if  $SolucAux$  é viável then
19            Troca solução selecionada por  $SolucAux$  em  $Filhos$ 
20             $NaoMudata \leftarrow FALSE$ 
21          end
22        end
23      end
24 end

```

3.2.5 Seleção dos Sobreviventes

A seleção dos sobreviventes ocorre logo após terminar a fase de mutação. Nesta etapa do algoritmo são escolhidos aqueles indivíduos que irão compor a próxima geração e passarão pelos processos de cruzamento e mutação novamente. Todos os indivíduos, tanto pais quanto filhos, são armazenados em uma única estrutura de dados para que ocorra a seleção, e então é feita a remoção de soluções idênticas.

Uma parte dos indivíduos selecionados para a próxima geração consiste em uma porcenta-

gem das soluções com menor valor da FO, ou seja, o conjunto das soluções Elite. A porcentagem utilizada neste trabalho foi de 30% das melhores soluções. Depois de serem selecionados e retirados os 30% melhores indivíduos dentre os pais e os filhos, são selecionados 50% de forma pseudo-aleatória das soluções restantes. A forma pseudo-aleatória implementada neste trabalho foi a *Roulette Wheel Selection* (TALBI, 2009) que atribui maior probabilidade de escolha àquelas soluções que possuem um menor valor da FO. Por fim, os outros 20% são selecionados de forma completamente aleatória. Todas as porcentagens são aplicadas ao tamanho do vetor de soluções composto pelos pais, ou seja, caso 50 pais gerassem 50 filhos, apenas 50 indivíduos de um total de 100 são selecionados para a próxima geração, sendo estes 50 compostos por 15 indivíduos com o menor valor da FO entre os 100, 25 selecionados de forma pseudo-aleatória entre os 85 restantes e 10 selecionados de forma completamente aleatória a partir dos 60 indivíduos que restaram.

3.2.6 Estrutura do Programa

Algoritmo 5: Main

```

1 begin
2   Cria um vetor de rotas BR
3   BR  $\leftarrow$  BancodeRotas()
4   VetSol  $\leftarrow$  PopulacaoInicial(BR)
5   Cria variavel GeracoesSemMelhoras  $\leftarrow$  0
6   Cria variavel MenorFO  $\leftarrow$  VetSol[0].FO
7   while GeracoesSemMelhoras < MaxIter do
8     GeracoesSemMelhoras++
9     Filhos  $\leftarrow$  Cruzamento(VetSol)
10    Mutacao(Filhos)
11    ListaAux  $\leftarrow$  VetSol + Filhos
12    VetSol  $\leftarrow$  SelecaoSobreviventes(ListaAux)
13    VetSol.sort(FO)
14    if VetSol[0].FO < MenorFO then
15      | GeracoesSemMelhoras  $\leftarrow$  0
16    end
17    MenorFO  $\leftarrow$  VetSol[0].FO
18  end
19 end

```

Durante um número n de gerações, ocorrem as etapas do AG até que os critérios de parada sejam satisfeitos. Neste trabalho foi utilizado o critério de gerações sem melhoria na solução de menor valor da FO. Ou seja, se em, por exemplo, 20 gerações ($MaxIter = 20$) não for gerada uma solução com um menor valor da FO do que da geração anterior, o algoritmo é interrompido

e a solução com menor valor da FO será considerada a melhor. A execução é apresentada no Algoritmo 5 que utiliza as funções descritas anteriormente em Banco de Rotas (3.2.1), População Inicial (3.2.2), Cruzamento (3.2.3), Mutação (3.2.4) e Seleção dos Sobreviventes (3.2.5). Esta mesma estrutura pode ser vista no fluxograma apresentado na Figura 3.1

3.3 Algoritmo 2

Nesta seção será descrito o algoritmo modificado neste trabalho utilizado para a resolução do PPRTP. Seu desenvolvimento foi baseado em sugestões da banca de qualificação da Monografia.

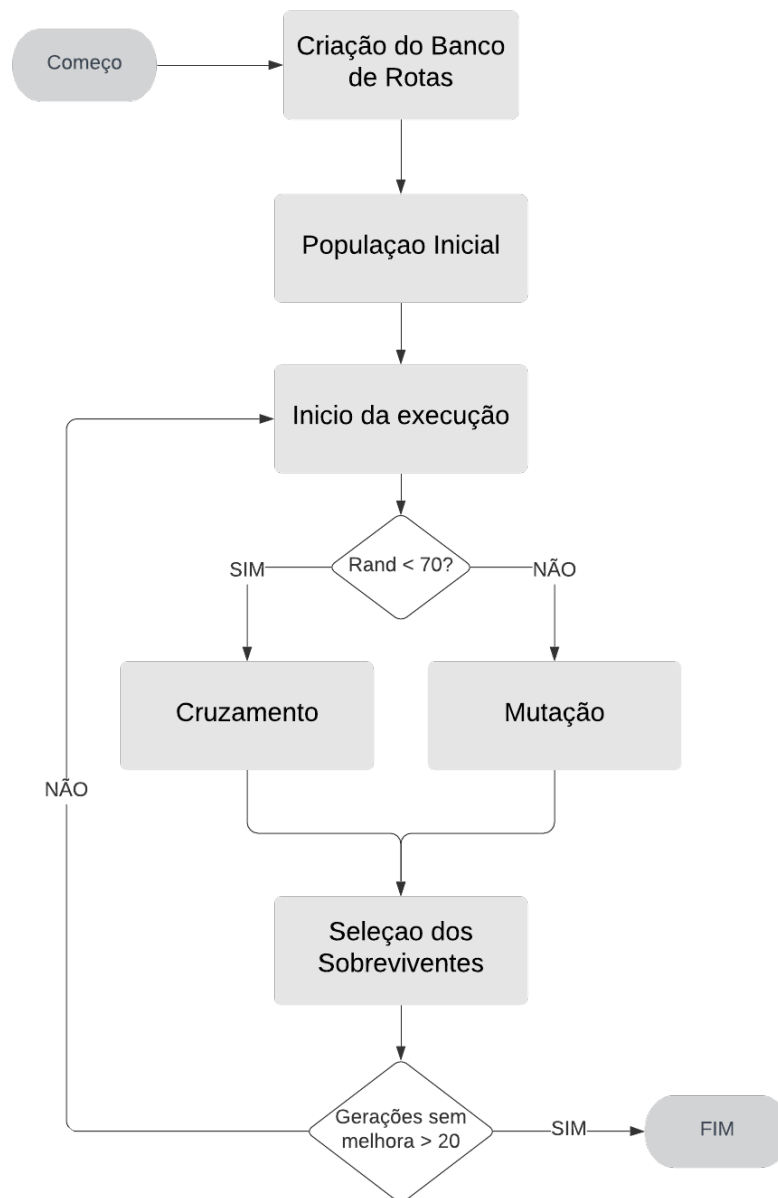


Figura 3.4 – Fluxograma do Algoritmo 2

Os primeiros passos deste algoritmo são exatamente os mesmos utilizados no algoritmo apresentado na seção 3.2, diferindo apenas no Algoritmo Genético, no procedimento de geração de novas soluções.

Diferentemente do AG da seção 3.2, em uma mesma geração não ocorrerão ambas as fases de Cruzamento e de Mutação. No início de cada geração será decidido se ocorrerá o Cruzamento ou a Mutação através de uma seleção randômica. Logo após os novos indivíduos serem gerados, acontecerá a seleção dos sobreviventes, que também é realizada de forma diferente da apresentada na seção anterior. A nova estrutura do programa pode ser vista no fluxograma apresentado na Figura 3.4. As novas formas de Cruzamento, Mutação e de Seleção de Sobreviventes serão apresentadas a seguir.

3.3.1 Cruzamento 2

Assim como o Cruzamento da seção 3, este também tem como dados de entrada o vetor de soluções atuais *VetSol* e o banco de rotas *BR* gerado anteriormente. Neste novo cruzamento, é fornecido também um *numCruzamentos*, que define o número mínimo de vezes que um pai participará de um cruzamento.

Os resultados obtidos pelo algoritmo anterior (3.2) mostram que a geração de novos filhos a partir da metodologia de pontos de corte é bastante limitada, pois a mesma gera muitos filhos inviáveis, o que acabava sendo um gargalo muito grande para se encontrar novas soluções depois de um certo número de gerações. Então, buscando aumentar as possibilidades de se obter filhos viáveis em grande quantidade, diversificando a busca, resolveu-se modificar a forma de geração de novos filhos a partir dos pais.

O vetor de soluções atual é percorrido, tratando cada solução como o primeiro pai. Para cada solução presente no vetor, são selecionadas outras k soluções, com $k = \text{numCruzamentos}$, de forma randômica, dentre as soluções que ainda não fizeram par com aquele pai, para que ocorra o Cruzamento entre a solução atual e cada uma destas outras soluções. Após escolher dois pais, é criado um banco de rotas auxiliar contendo apenas as rotas presentes nas duas soluções. A partir desse banco auxiliar, são selecionadas de forma aleatória rotas para compor a nova solução até que a mesma seja viável. Desta forma garantimos que a nova solução gerada seja viável, sendo no pior caso uma solução idêntica a um dos pais, o que é tratado posteriormente na seleção de sobreviventes. Esta mesma execução é apresentada no Algoritmo 6.

Algoritmo 6: Cruzamento 2

Entrada: Vetor de Soluções $VetSol$, Grafo G , Banco de Rotas BR , Numero de cruzamentos $numCruzamentos$.

Saída: Lista de soluções $Filhos$ com as soluções geradas pela Cruzamento.

```

1 begin
2   Cria uma lista de soluções  $Filhos$ 
3   Cria uma lista de soluções  $SolucoesRestantes$ 
4   Cria uma solução  $FilhoProv$ 
5   Cria uma solução  $SegundoPai$ 
6   Cria uma lista de Rotas  $BancoAux$ 
7   for cada  $Solucao$  em  $VetSol$  do
8      $SolucoesRestantes \leftarrow VetSol$ 
9      $SolucoesRestantes$  removes  $Solucao$ 
10    for  $i \leftarrow 0$  até  $numCruzamentos$  do
11       $BancoAux.clear()$ 
12       $BancoAux \leftarrow Solucao.Rotas$ 
13       $SegundoPai \leftarrow SelecionaRand(SolucoesRestantes)$ 
14       $SolucoesRestantes$  removes  $SegundoPai$ 
15       $BancoAux \leftarrow SegundoPai.Rotas$ 
16       $BancoAux.RemoveEquals()$ 
17      while  $FilhoProv$  não é viável do
18         $FilhoProv \leftarrow SelecionaRand(BancoAux)$ 
19      end
20       $Filhos \leftarrow FilhoProv$ 
21    end
22  end
23  return  $Filhos$ 
24 end

```

3.3.2 Mutaç o 2

A nova Mutaç o ocorre de forma semelhante   apresentada na seç o 4, por m diferentemente da anterior, os indiv duos a sofrerem mutaç o n o s o selecionados de forma aleat ria, e sim sistematicamente, de modo que cada soluç o presente no vetor de soluç o gere no m nimo um novo filho diferente dele, ou at  que as possibilidades para mutaç o daquele indiv duo se esgote. Ou seja, ocorre uma Busca Local em cada rota das soluç es atuais.

Para cada rota presente na soluç o, s o testadas em seu lugar todas as rotas do Banco de Rotas com os mesmos pontos extremos, visando assim obter um maior n mero de sucessos em gerar novas rotas vi veis por meio da mutaç o. Assim, ap s esta verificaç o para cada rota de

cada solução, são selecionados os indivíduos para a próxima geração, como detalhado na seção seguinte.

Algoritmo 7: Mutação 2

Entrada: Vetor de Soluções *VetSol*, Grafo *G*, Banco de Rotas *BR*.

Saída: Lista de Soluções *NovasSolucoes* com as soluções geradas pela Mutação

```

1 begin
2   Cria uma lista de Soluções NovasSolucoes
3   Cria uma Solução SolucaoAux
4   for cada Solucao em VetSol do
5     for cada Rota em Solucao do
6       Extremo1 ← Rota.begin
7       Extremo2 ← Rota.end
8       if Rotas(Extremo1, Extremo2).size > 1 then
9         for cada RotaAux em Rotas(Extremo1, Extremo2) do
10          SolucaoAux ← Solucao
11          SolucaoAux.remove(Rota)
12          SolucaoAux.Rotas ← RotaAux
13          if SolucaoAux é viável then
14            NovasSolucoes ← SolucaoAux
15          end
16        end
17      end
18    end
19  end
20  return NovasSolucoes
21 end

```

3.3.3 Seleção dos sobreviventes 2

Após ser executada a etapa de Cruzamento ou de Mutação, segue a etapa de Seleção de Sobreviventes, na qual são selecionadas as soluções que irão compor a próxima geração. Como na seleção da seção 3.2.5, todos os indivíduos, tanto pais quanto filhos, são agrupados em uma única estrutura de dados para que ocorra a seleção, e então é feita a remoção de soluções idênticas.

A seleção ocorre de forma similar à da Seção 3.2.5, porém o número de soluções selecionadas para o conjunto elite é menor, visando evitar o mínimo possível o novo algoritmo, atrasando assim a estagnação do mesmo. Foi utilizado o valor de 7% da nova geração, seguindo sugestão da banca de qualificação do trabalho. Considerando-se 100 soluções por geração, apenas 7 das melhores soluções vão obrigatoriamente para a próxima geração, os outros 93% são selecionados de forma pseudo-aleatória, utilizando *Método de Seleção de Torneio* (TALBI, 2009),

onde são selecionadas varias soluções de forma aleatória, e a de menor valor de FO é escolhida para a próxima geração.

3.3.4 Estrutura do Programa 2

Algoritmo 8: Main

```

1 begin
2   Cria um vetor de rotas BR
3   BR ← BancodeRotas()
4   VetSol ← PopulacaoInicial(BR)
5   Cria variavel GeracoesSemMelhoras ← 0
6   Cria variavel MenorFO ← VetSol[0].FO
7   while GeracoesSemMelhoras < MaxIter do
8     GeracoesSemMelhoras++
9     Cria variavel Decisao ← rand(1, 100)
10    if Decisao ≤ 70 then
11      | Filhos ← Cruzamento(VetSol)
12    end
13    else
14      | Filhos ← Mutacao(VetSol)
15    end
16    ListaAux ← VetSol + Filhos
17    VetSol ← SelecaoSobreviventes(ListaAux)
18    VetSol.sort(FO)
19    if VetSol[0].FO < MenorFO then
20      | GeracoesSemMelhoras ← 0
21    end
22    MenorFO ← VetSol[0].FO
23  end
24 end

```

Durante um número n de gerações, ocorrem as etapas do AG até que os critérios de parada sejam satisfeitos. Assim como apresentado na seção 3.2.6 o critério escolhido foi o de gerações sem melhoria na solução de menor valor da FO. Ou seja, se em, por exemplo, 20 gerações ($MaxIter = 20$) não for gerada uma solução com um menor valor da FO do que da geração anterior, o algoritmo é interrompido e a solução com menor valor da FO será considerada a melhor. A execução é apresentada no Algoritmo 8 que utiliza as funções descritas anteriormente em Banco de Rotas (3.2.1), População Inicial (3.2.2), Cruzamento (3.3.1), Mutação (3.3.2) e Seleção dos Sobreviventes (3.3.3). Esta mesma estrutura pode ser vista no fluxograma apresentado na Figura 3.4

3.4 Validação

Ao final da construção dos algoritmos apresentados, foram realizados testes com a rede Mandl (MANDL, 1980), considerada uma rede *Benchmark* da literatura, para validar o algoritmo implementado. A rede Mandl, apresentada na Figura 3.5, é composta por 15 nós e 42 arestas contendo pesos que representam o tempo em minutos para percorrê-las de um extremo ao outro.

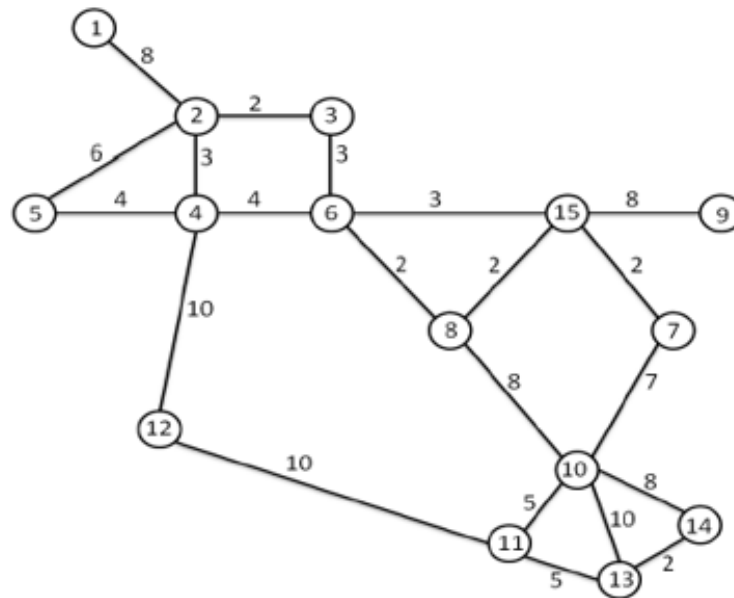


Figura 3.5 – Rede de Mandl.

Fonte: Figura retirada de Mandl (1980)

A rede também conta com uma matriz origem-destino (O/D), representada na Tabela 3.1, onde é informado o número de pessoas que desejam ir de um nó origem (linha) a cada nó destino (coluna) em um determinado intervalo de tempo. Por exemplo, a linha 1 coluna 2 da Tabela 3.1 mostra que há uma demanda de 400 pessoas a serem transportadas do nó 1 para o nó 2 no intervalo de tempo estudado. Esta matriz é simétrica pois considera que os usuários devem retornar para o mesmo ponto de origem. Neste trabalho, o intervalo de tempo analisado foi de uma hora. A partir da rede e da matriz O/D apresentadas, foram geradas soluções pelo AG1 e AG2 para que estas possam ser comparadas com soluções de outros trabalhos da literatura para essa mesma rede. As soluções encontradas e a comparação com soluções da literatura serão apresentadas no Capítulo 4.

Origem/ Destino	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	400	200	60	80	150	75	75	30	160	30	25	35	0	0
2	400	0	50	120	20	180	90	90	15	130	20	10	10	5	0
3	200	50	0	40	60	180	90	90	15	45	20	10	10	5	0
4	60	120	40	0	50	100	50	50	15	240	40	25	10	5	0
5	80	20	60	50	0	50	25	25	10	120	20	15	5	0	0
6	150	180	180	100	50	0	100	100	30	880	60	15	15	10	0
7	75	90	90	50	25	100	0	50	15	440	35	10	10	5	0
8	75	90	90	50	25	100	50	0	15	440	35	10	10	5	0
9	30	15	15	15	10	30	15	15	0	140	20	5	0	0	0
10	160	130	45	240	120	880	440	440	140	0	600	250	500	200	0
11	30	20	20	40	20	60	35	35	20	600	0	75	95	15	0
12	25	10	10	25	15	15	10	10	5	250	75	0	70	0	0
13	35	10	10	10	5	15	10	10	0	500	95	70	0	45	0
14	0	5	5	5	0	10	5	5	0	200	15	0	45	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tabela 3.1 – Tabela de Origem / Destino da rede de Mandl (1980).

3.5 Aplicação em Ouro Preto

Nesta seção iremos apresentar os dados utilizados para fazer o estudo do caso da cidade de Ouro Preto - MG, sendo: *i*) um mapa mostrando a divisão das regiões do município de Ouro Preto (Figura 3.6), *ii*) a Matriz de demandas O/D (Tabela 3.2), e *iii*) o grafo que representa a cidade (Figura 3.7).

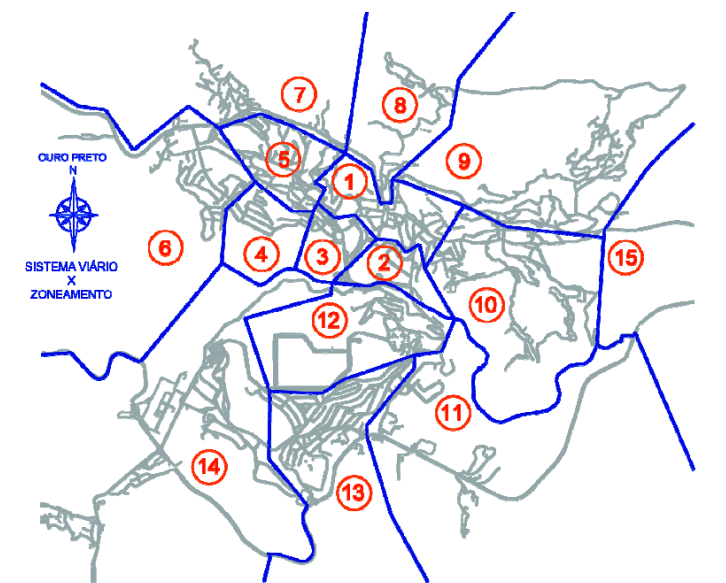


Figura 3.6 – Divisões de Ouro Preto

Na Figura 3.6 mostra uma divisão da cidade de forma que cada nó no grafo corresponde a uma região e não a um ponto específico dentro daquela região, visto que se uma região é alcançável através de outra, qualquer ponto dentro da mesma também será alcançável, sendo desnecessária a determinação de um ponto específico.

Na Figura 3.7 pode-se observar um grafo correspondente às conexões existentes na infraestrutura viária de Ouro Preto. Os nós do grafo foram definidos a partir da Figura 3.6, retirada do Diagnóstico do Sistema de Mobilidade de Ouro Preto (2015), que foi cedido gentilmente pela prefeitura da cidade.

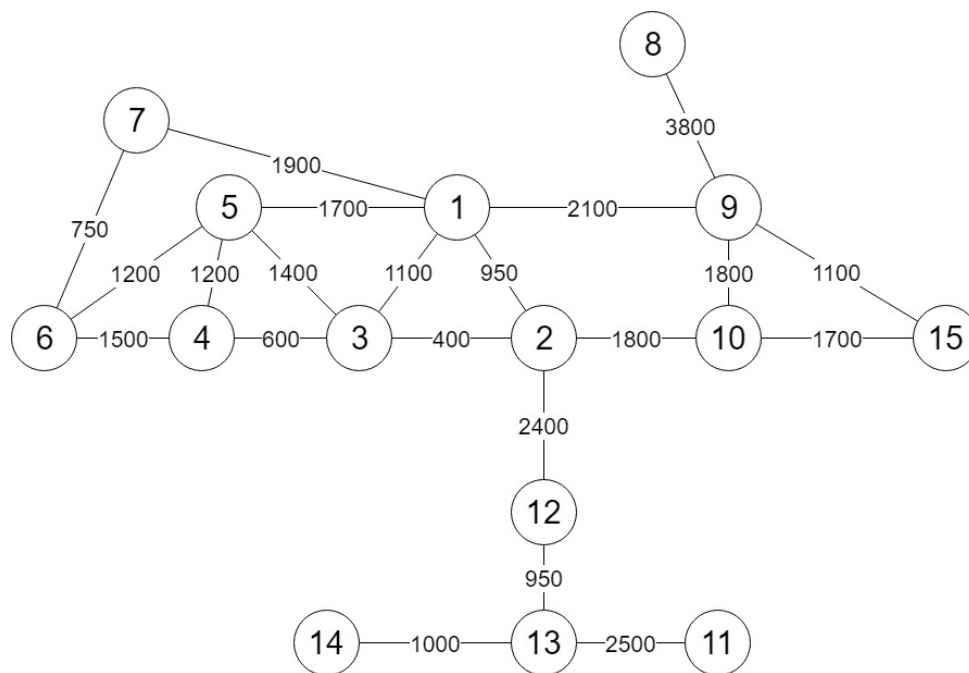


Figura 3.7 – Rede de Ouro Preto

As arestas foram incluídas baseadas em conjuntos de rotas já aplicadas na cidade no ano de 2007, Edital de Concorrência Pública n.º 04/2007 (OURO PRETO, 2008), para assim garantir que todas as conexões no grafo existem na infraestrutura da cidade. Os números em cada aresta são as distâncias, em metros, de um ponto ao outro. Estas distâncias não são o peso final as arestas, visto que o peso usado pelo programa está em minutos. Os pesos de cada aresta são calculados internamente na aplicação.

A aplicação recebe de um arquivo texto a quantidade de nós, a quantidade de arcos e a velocidade média dos ônibus na cidade. Após isso, são descritas as conexões de cada nó juntamente com a distância correspondente a cada uma, e através das distâncias e da velocidade são calculados os tempos (pesos das arestas) necessários para ir de um nó até outro.

Assim como apresentado na Seção 3.4 temos uma tabela de Origem/Destino 3.2 para nos informar a quantidades de pessoas que desejam ir de um nó a outro.

Origem/ Destino	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1782	167	56	4	50	16	101	0	291	119	27	130	466	84	3
2	214	77	96	0	8	29	93	0	59	23	45	63	262	46	2
3	525	67	87	0	0	27	86	0	160	122	2	67	257	32	0
4	118	0	0	0	0	13	0	0	31	7	0	29	86	0	4
5	324	24	6	3	81	6	51	0	22	50	1	56	170	12	11
6	238	15	19	0	5	62	34	0	87	29	0	12	98	50	0
7	850	141	43	4	12	0	138	0	396	93	2	182	536	55	18
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	667	50	55	3	15	16	313	0	488	2	49	120	347	11	7
10	439	77	62	33	49	20	409	0	15	202	1	58	218	0	5
11	212	64	33	0	3	0	6	0	36	2	50	12	13	0	2
12	510	82	179	33	58	53	106	0	117	7	66	0	226	1	13
13	473	79	102	30	90	129	246	0	149	32	20	11	294	44	7
14	615	63	130	0	14	219	222	0	11	0	0	1	90	80	0
15	110	0	0	0	21	0	52	0	21	0	0	75	106	0	105

Tabela 3.2 – Tabela de Origem / Destino da rede de Ouro Preto.

4 Resultados

Neste capítulo são apresentados e analisados os resultados obtidos pelos algoritmos descritos nas Seções 3.2 e 3.3 e previamente mencionados na Seção 3.4. Para cada um dos algoritmos foram feitas 30 execuções de cada um para que pudessem ser coletados os resultados e calculadas as médias apresentadas nas seções seguintes. Não feita a medição de tempo de execução de cada algoritmo, visto que os mesmos não necessitam de estar sempre exibindo um resultado para o usuário, e sim estar buscando o melhor resultado independente do tempo gasto.

4.1 Algoritmo 1

Nesta seção serão apresentados os resultados do Algoritmo 1 aplicados na rede de Mandl (1980) e na rede de Ouro Preto. Neste algoritmo, tanto para a rede de Mandl (1980) quanto para a rede de Ouro Preto, o mesmo executava durante cerca de 50 gerações até que atingisse o critério de parada determinado.

4.1.1 Rede de Mandl

Na Figura 4.1 são apresentadas algumas soluções para a rede de Mandl (1980) encontradas pelo Algoritmo 1 implementado neste trabalho. A seguir, estas soluções são comparadas com os resultados de Arbex e Cunha (2015) obtidos para esta mesma rede.

As soluções da Figura 4.1 foram escolhidas por conterem rotas bem distintas umas das outras. Assim pode-se verificar a diversidade das rotas geradas pelo AG. Cada uma destas soluções foi a melhor solução encontrada pelo algoritmo em uma determinada execução. Ao comparar as soluções do AG com as soluções de Arbex e Cunha (2015) são obtidos os resultados apresentados na Tabela 4.1.

Ao analisar a Tabela 4.1 podemos observar que os valores obtidos pelo AG1 são bastante parecidos tanto para o número de rotas na solução, o número de nós por rota quanto a frota utilizada. Ao analisarmos o *Headway* médio, que é o intervalo entre as partidas, vemos que o mesmo teve um tempo mais baixo, pois mesmo que as rotas sejam menores que as apresentadas em Arbex e Cunha (2015), elas necessitam suprir uma mesma demanda de passageiros transportados.

Um ponto que pode ser observado nas soluções apresentadas é que as consideradas melhores soluções pelo algoritmo são aquelas que distribuem a maior parte da demanda por poucas rotas, possibilitando assim que as rotas sejam menores, apesar de estarem sobrecarregando algumas. Isso se deve muito ao fato de como são definidos os custos na FO. Neste trabalho, a expressão da FO atribui um peso grande ao tamanho das rotas, fazendo com que sejam selecionadas soluções com rotas menores. Com isso, analisando do ponto de vista operacional,

Nº da Linha	Nº de Origem	Nº de Destino	Sequência de nós	# nós	Tempo Viagem (minutos)	Frota	Headway (minutos)	Valor Crítico
Solução 1								
1	1	6	1-2-3-6	4	13	9	3	1320
2	3	7	3-6-8-15-7	5	9	3	6	510
3	3	12	3-2-4-12	4	15	3	10	270
4	5	12	5-2-4-12	4	19	7	6	480
5	9	4	9-15-6-4	4	15	5	7	425
6	12	14	12-11-13-14	4	17	6	6	465
7	14	10	14-13-11-10	4	12	12	2	2500
					14,3	45	5,8	
Solução 2								
1	1	7	1-2-3-6-15-7	6	18	9	4	939
2	3	6	3-2-4-6	4	9	3	8	380
3	4	15	4-6-8-15	4	8	6	3	995
4	5	4	5-2-3-6-4	5	15	5	6	480
5	9	10	9-15-7-10	4	17	12	3	1290
6	12	3	12-4-2-3	4	15	6	5	620
7	14	10	14-13-11-10	4	12	12	2	1745
					13,7	41	4,9	
Solução 3								
1	1	5	1-2-4-5	4	15	10	3	980
2	4	6	4-2-3-6	4	8	2	10	275
3	6	2	6-4-5-2	4	14	10	3	1027
4	7	14	7-15-8-10-14	5	20	20	2	2640
5	9	4	9-15-6-4	4	15	3	12	233
6	14	5	14-13-11-12-4-5	6	31	31	2	1750
					14,8	76	4,6	

Figura 4.1 – Resultados do AG1 para a rede de Mandl (1980)

o algoritmo gera soluções sem valor prático. Porém, com alguns ajustes, o mesmo pode gerar soluções interessantes. Entre as soluções obtidas, aquela que apresentou um melhor equilíbrio entre os fatores foi a Solução 2, pois a mesma possui um menor valor da frota necessária e ainda conta com baixo tempo médio.

Comparação	AG	Arbex e Cunha (2015)
Nº de linhas na solução	7	8
Nº médio de nós por rota	4,42	4,8
Frota	51	54
Tempo Médio das rotas	14,2	18,25
Headway Médio (Minutos)	5,1	5,55

Tabela 4.1 – Comparação entre nossos resultados do Algoritmo 1 e de Arbex e Cunha (2015).

4.1.2 Rede de Ouro Preto

Nesta seção serão apresentados os resultados do Algoritmo 1 para a rede de Ouro Preto. Como podemos ver na Figura 4.2, os resultados se assemelham com os obtidos pelo AG1 para a rede de Mandl (1980). Em parte, isso se deve ao fato de que as duas redes são muito parecidas em termos de quantidade de nós e quantidade de arestas na rede.

Um fato que difere as duas redes é a existência de nós que são muito isolados do restante da rede no caso de Ouro Preto. Estes nós se ligam à rede através de um único caminho, que seria passando pelo nó 12, como apresentado na Figura 3.7. Isso faz com que as demandas desses nós sejam distribuídas em poucas rotas, fazendo com que fiquem com um carregamento maior, consequentemente aumentando a frota nestas rotas.

Outra diferença notável entre as redes é a quantidade de usuários que desejam utilizar a rede para se locomoverem dentre do mesmo nó que se encontram, o que não acontecia na rede de Mandl (1980). Com isso o carregamento em rotas que são as únicas a atender um determinado nó, crescem de forma significativa. Podemos observar isso nas soluções apresentadas na Figura 4.2, onde na primeira solução apenas uma rota está atendendo ao nó 1, fazendo com que o valor crítico da mesma cresça significativamente. Cada uma destas soluções foi a melhor solução encontrada pelo algoritmo em uma determinada execução.

Nº da Linha	Nó de Origem	Nó de Destino	Sequência de nós	# nós	Tempo Viagem (minutos)	Frota	Headway (minutos)	Valor Crítico
Solução 1								
1	4	6	4-3-5-6	4	14	1	30	87
2	7	2	7-6-4-3-2	5	19	3	15	188
3	8	10	8-9-15-10	4	32	4	20	116
4	10	3	10-2-1-3	4	16	16	2	2342
5	11	3	11-13-12-2-3	5	32	13	5	614
6	14	2	14-13-12-2	4	22	15	3	971
7	15	3	15-10-2-3	4	14	3	10	281
					21,3	55	12,2	
Solução 2								
1	1	2	1-5-3-2	4	15	10	3	1413
2	6	15	6-4-3-2-10-15	6	23	4	15	185
3	8	7	8-9-1-7	4	31	21	3	1092
4	10	4	10-2-3-4	4	12	1	30	84
5	11	10	11-13-12-2-10	5	37	19	4	854
6	14	2	14-13-12-2	4	22	11	4	926
					23,4	66	9,9	
Solução 3								
1	2	6	2-3-4-6	4	11	1	30	85
2	7	3	7-6-5-3	4	19	2	20	135
3	8	2	8-9-1-2	4	27	18	3	1326
4	9	4	9-1-3-4	4	16	11	3	1326
5	11	3	11-13-12-2-13	5	32	22	3	984
6	14	4	14-13-12-2-3-4	6	27	18	3	1037
7	15	3	15-10-2-3	4	14	2	20	139
					20,9	74	11,8	

Figura 4.2 – Resultados do AG1 para a rede de Ouro Preto

Na última linha de cada solução, são apresentados o tempo médio das rotas que compõem a solução, a frota total daquela solução e o *Headway* médio dentre as rotas. Podemos observar que em comparação com as soluções da rede de Mandl (1980), estes fatores se assemelham bastante, exceto em relação à frota, sendo que para rede de Ouro Preto, a frota necessária para suprir a demanda foi maior.

4.2 Algoritmo 2

Nesta seção serão apresentados os resultados referentes ao Algoritmo 2 aplicado na rede de Mandl (1980) e na rede de Ouro Preto.

Além das mudanças feitas nas funções de Cruzamento, Mutação e Seleção dos sobreviventes, também foi feita uma alteração em alguns parâmetros da criação do banco de rotas, para que a mesma permitisse que rotas maiores também fizessem parte do banco, além de um

pequeno ajuste de pesos na FO, para que o tamanho das rotas deixasse de ter uma predominância tão grande.

Este algoritmo, diferentemente do apresentado na seção anterior, teve uma média de 70 gerações, ou seja, ele executou todos os passos do AG cerca de 70 vezes até que fosse atingido o critério de parada. Grande parte dessa melhoria, se dá à nova mutação, que é responsável por realizar uma busca mais minuciosa entre as soluções.

4.2.1 Rede de Mandl

Na Figura 4.3 estão apresentadas algumas soluções para a rede de Mandl (1980) encontradas pelo Algoritmo 2 implementado neste trabalho.

Nº da Linha	Nó de Origem	Nó de Destino	Sequência de nós	# nós	Tempo Viagem (minutos)	Frota	Headway (minutos)	Valor Crítico
Solução 1								
1	1	15	1-2-3-6-15	5	16	11	3	1320
2	2	10	2-3-6-15-7-10	6	17	17	2	2595
3	9	3	9-15-8-6-4-2-3	7	21	3	15	179
4	9	4	9-15-8-6-4	5	16	2	20	149
5	9	12	9-15-8-6-4-12	6	26	3	20	150
6	5	9	5-4-6-8-15-9	6	25	3	20	149
7	14	2	14-13-11-12-4-2	6	30	30	2	1678
					21,6	69	11,8	
Solução 2								
1	5	2	5-4-6-3-2	5	13	7	4	939
2	8	1	8-6-3-2-1	5	15	2	15	200
3	8	1	8-15-6-3-2-1	6	18	3	15	200
4	9	3	9-15-8-6-3	5	15	2	20	134
5	11	7	11-12-4-6-8-15-7	7	30	5	12	245
6	14	7	14-13-11-10-7	5	19	19	2	1746
7	14	10	14-13-11-10	4	12	8	3	1298
					18,4	38	11,4	
Solução 3								
1	2	6	5-2-4-12	4	19	2	30	85
2	7	3	5-4-12-11-13	5	29	3	20	135
3	8	2	7-15-6-4-12-11-13-14	8	36	24	3	1027
4	9	4	8-6-3-2-1	5	15	3	12	233
5	11	3	8-15-6-3-2-1	6	18	3	12	233
6	14	4	9-15-8-6-3	5	15	4	9	310
7	15	3	11-13-14-10	4	15	15	2	2595
					21	54	12,6	

Figura 4.3 – Resultados do AG2 para a rede de Mandl (1980)

Estas soluções foram escolhidas por conter algumas diferenças entre elas, uma vez que a maioria das soluções geradas pelo algoritmo são muito parecidas em vários aspectos. Analisando a Solução 2 apresentada na Figura 4.3, podemos ver que a frota necessária para atender a demanda

foi menor, pois nesta solução os nós que possuem uma demanda maior estão presentes em mais de uma rota, fazendo com que a demanda seja melhor distribuída entre as rotas. Cada uma destas soluções foi a melhor solução encontrada pelo algoritmo em uma determinada execução.

Como podemos ver, analisando a Tabela 4.2, as soluções obtidas pelo Algoritmo 2 possuem rotas maiores em relação ao Algoritmo 1 e mesmo assim a frota necessária se manteve bem próxima à média. O *headway* médio é maior, pois nestas soluções a demanda estava sendo distribuída muito mal, fazendo com que a maioria das rotas tivessem um *headway* elevado, aumentando assim o valor médio.

Comparação	AG 1	AG 2
Nº de linhas na solução	7	7
Nº de nós por rota	4,42	5,6
Frota	51	54
Tempo Médio das rotas	14,4	20,6
Headway Médio (Minutos)	5,1	11,36

Tabela 4.2 – Comparação entre nossos resultados do Algoritmo 1 e do Algoritmo 2 para a rede de Mandl (1980)

4.2.2 Rede de Ouro Preto

Nesta seção serão apresentados resultados do Algoritmo 2 para a rede de Ouro Preto. Como podemos observar nas soluções apresentadas na Figura 4.4 os tempos médios das rotas se assemelham bastante com as soluções encontradas pelo Algoritmo 1. No entanto, as frotas estão com valores maiores. Cada uma destas soluções foi a melhor solução encontrada pelo algoritmo em uma determinada execução.

Podemos ver também que, apesar de diminuir o peso do tamanho das rotas na FO, diferentemente pra rede de Mandl (1980), as rotas se mantiveram muito parecidas, ao invés de serem selecionadas rotas maiores.

Apesar da rede de Mandl (1980) e de Ouro Preto serem muito parecidas em questão de número de nós e número de arestas, a forma com que estas arestas conectam os nós são diferentes. Na rede de Ouro Preto temos uma sub-rede que pode ser acessada apenas através de uma aresta, fazendo com que qualquer rota que acessa esta sub-rede conta com um valor de tempo médio elevado, por se tratar de um caminho longo. Outro ponto é que esta mesma sub-rede possui dois extremos, fazendo com que o mínimo de rotas necessárias para atender esta sub-rede sejam duas, aumentando ainda mais o tempo médio das rotas.

Uma grande diferença que podemos perceber, analisando a Tabela 4.3, é que o *headway* médio das soluções diminuiu consideravelmente. Isso se deve ao fato de que o Algoritmo 2 está

Nº da Linha	Nó de Origem	Nó de Destino	Sequência de nós	# nós	Tempo Viagem (minutos)	Frota	Headway (minutos)	Valor Crítico
Solução 1								
1	1	15	1-2-10-15	4	16	11	3	1221
2	6	2	6-4-3-2	4	11	2	20	126
3	7	2	7-6-5-3-2	5	21	5	9	342
4	8	3	8-9-10-2-3	5	32	5	15	177
5	10	1	10-2-3-1	4	14	10	3	1418
6	11	2	11-13-12-2	4	30	15	4	883
7	14	4	14-13-12-2-3-4	6	27	18	3	1013
					21,6	66	8,2	
Solução 2								
1	1	6	1-3-4-6	4	14	14	2	2122
2	4	6	4-3-5-6	4	14	4	7	428
3	7	3	7-6-4-3	4	17	9	4	820
4	8	2	8-9-10-2	4	30	5	12	211
5	11	3	11-13-12-2-3	5	32	16	4	943
6	14	5	14-13-12-2-3-5	6	30	20	3	1194
7	15	4	15-10-2-3-4	5	17	5	8	380
					22	73	5,8	
Solução 3								
1	4	10	4-3-2-10	4	12	4	6	516
2	7	3	7-1-2-3	4	14	14	2	1626
3	7	6	7-1-5-6	4	20	20	2	1740
4	8	15	8-9-10-15	4	28	4	15	176
5	11	2	11-13-12-2	4	30	20	3	973
6	14	2	14-13-12-2	4	22	15	3	1291
					18	77	4,5	

Figura 4.4 – Resultados do AG2 para a rede de Ouro Preto

um pouco mais equilibrado, encontre assim, soluções que apresentam uma demanda mais bem distribuída.

Como consequência de *headway* menor e de haver rotas com um tempo médio maior, a frota necessária para suprir a demanda cresce bastante, pois o tempo necessário para que um veículo conclua seu ciclo e possa atender a um novo horário de saída, é maior.

A Solução 3, apresentada na Figura 4.2, se mostrou um pouco diferente das demais. Além de possuir rotas de um tamanho menor que a média, ela possui um número menor de rotas, levando a uma relação maior de demanda por número de rotas, aumentando a demanda em cada rota. Além disso, todas as rotas da solução possuem o número mínimo de nós aceitável. Como podemos ver, esta combinação resultou em uma solução com uma demanda melhor distribuída, porém com um *headway* baixo, e uma frota elevada.

Comparação	AG 1	AG 2
Nº de linhas na solução	7	7
Nº de nós por rota	4,72	5,6
Frota	63	72
Tempo Médio das rotas	21,5	21,3
Headway Médio (Minutos)	11,13	7,4

Tabela 4.3 – Comparação entre nossos resultados do Algoritmo 1 e do Algoritmo 2 para a rede de Ouro Preto

5 Considerações Finais

5.1 Conclusão

Este trabalho teve como objetivos: *i*) o estudo da literatura no que diz respeito ao Problema do Projeto de Rotas de Transporte Público (PPRTP), *ii*) a implementação de dois Algoritmos Genéticos que resolvam este problema de modo que eles definam as rotas que irão compor a solução, definam os horários de partida dos ônibus e determinem a quantidade necessária de ônibus para suprir toda a demanda fornecida e *iii*) realizar um estudo de caso da cidade de Ouro Preto - MG.

Ao fim deste trabalho, pôde ser observado que existem muitos trabalhos relacionados a este tema e que os pontos estudando por estes trabalhos são de grande relevância para a construção de um algoritmo que supra todas as necessidades de uma rede de forma eficiente, considerando principalmente suas restrições, que são o guia para se obter um bom resultado.

A respeito dos algoritmos implementados, baseando-se no processo de validação, concluímos que os mesmos apresentaram boas soluções para o problema, porém um pouco fora da realidade que presenciamos, se considerarmos os conjuntos de rotas que já estão em vigor na cidade de Ouro Preto. Se compararmos os os dois Algoritmos Genéticos, AG1 e AG2 implementados, podemos notar algumas diferenças em relação às soluções. O algoritmo AG2 faz uma busca local mais detalhada do que o AG1, tornando mais diversificada a geração das soluções a cada iteração. Por outro lado, o Algoritmo 2 se mostra menos enviesado a selecionar soluções com rotas que possuem um número menor de nós do que o Algoritmo 1. Acredito que melhorias possam ser feitas em ambos os algoritmos no que se trata de cálculo do valor da FO e uma melhor distribuição da demanda, visto que cada um destes tópicos por si só já é uma área de estudo bastante complexa do PPRTP.

5.2 Trabalhos Futuros

Para futuros trabalhos temos como objetivo a melhoria dos algoritmos como dito anteriormente, estudando melhor a composição da FO e da distribuição de demandas, que são partes essenciais do PPRTP.

Referências

- AFANDIZADEH, S.; KHAKSAR, H.; KALANTARI, N. Bus fleet optimization using genetic algorithm a case study of mashhad. *International Journal of Civil Engineering*, International Journal of Civil Engineering, v. 11, n. 1, p. 43–52, 2013.
- ARBEX, R. O.; CUNHA, C. B. da. Efficient transit network design and frequencies setting multi-objective optimization by alternating objective genetic algorithm. *Transportation Research Part B: Methodological*, Elsevier, v. 81, p. 355–376, 2015.
- BAAJ, M. H.; MAHMASSANI, H. S. An ai-based approach for transit route system planning and design. *Journal of advanced transportation*, Wiley Online Library, v. 25, n. 2, p. 187–209, 1991.
- BHTRANS. 2013. Disponível em: <<http://www.bhtrans.pbh.gov.br/portal/page/portal/portalpublico/Temas/Observatorio/PLANMOB-diagnostico-2013>>. Acesso em: 1 fev. 2019.
- BLUM, J. J.; MATHEW, T. V. Intelligent agent optimization of urban bus transit system design. *Journal of Computing in Civil Engineering*, American Society of Civil Engineers, v. 25, n. 5, p. 357–369, 2010.
- CEDER, A.; WILSON, N. H. Bus network design. *Transportation Research Part B: Methodological*, Elsevier, v. 20, n. 4, p. 331–344, 1986.
- CHEW, J. S. C.; LEE, L. S. A genetic algorithm for urban transit routing problem. In: WORLD SCIENTIFIC. *International Journal of Modern Physics: Conference Series*. [S.l.], 2012. v. 9, p. 411–421.
- CIAFFI, F.; CIPRIANI, E.; PETRELLI, M. Feeder bus network design problem: A new metaheuristic procedure and real size applications. *Procedia-Social and Behavioral Sciences*, Elsevier, v. 54, p. 798–807, 2012.
- DIJKSTRA, E. W. A note on two problems in connexion with graphs. *Numerische mathematik*, Springer, v. 1, n. 1, p. 269–271, 1959.
- FAN, W.; MACHEMEHL, R. Optimal transit route network design problem: Algorithms. *Implementations, and Numerical Results*, 2004.
- FAN, W.; MACHEMEHL, R. B. Optimal transit route network design problem with variable transit demand: genetic algorithm approach. *Journal of transportation engineering*, American Society of Civil Engineers, v. 132, n. 1, p. 40–51, 2006.
- GUIHAIRE, V.; HAO, J.-K. Transit network design and scheduling: A global review. *Transportation Research Part A: Policy and Practice*, Elsevier, v. 42, n. 10, p. 1251–1273, 2008.
- JIRÓN, P. Technology and technicians out of control: The implementation of transantiago from a daily mobility point of view. L. Vannini, C. Budd, P. Fisker, Jiron, P. & O. Jensen, *Technologies of mobility in the Americas. Series Intersections in communications and culture: Global approaches and transdisciplinary perspectives*, v. 29, p. 255–277, 2012.
- KEPAPTSOGLU, K.; KARLAFTIS, M. Transit route network design problem. *Journal of transportation engineering*, American Society of Civil Engineers, v. 135, n. 8, p. 491–505, 2009.

- KUO, Y. Design method using hybrid of line-type and circular-type routes for transit network system optimization. *Top*, Springer, v. 22, n. 2, p. 600–613, 2014.
- MANDL, C. E. Evaluation and optimization of urban public transportation networks. *European Journal of Operational Research*, Elsevier, v. 5, n. 6, p. 396–404, 1980.
- TALBI, E.-G. *Metaheuristics: from design to implementation*. [S.l.]: John Wiley & Sons, 2009. v. 74.
- YANG, Z.; YU, B.; CHENG, C. A parallel ant colony algorithm for bus network optimization. *Computer-Aided Civil and Infrastructure Engineering*, Wiley Online Library, v. 22, n. 1, p. 44–55, 2007.
- YEN, J. Y. Finding the k shortest loopless paths in a network. *management Science*, INFORMS, v. 17, n. 11, p. 712–716, 1971.