

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO DE JOINVILLE
CURSO DE ENGENHARIA DE TRANSPORTES E LOGÍSTICA

RUBENS CARLOS DE OLIVEIRA NETO

ESTUDO DE VIABILIDADE DO MODELO METROVIÁRIO SEM PARADAS

Joinville

2019

RUBENS CARLOS DE OLIVEIRA NETO

ESTUDO DE VIABILIDADE DE MODELO METROVIÁRIO SEM PARADAS

Trabalho apresentado como requisito para obtenção do título de bacharel no Curso de Graduação em Engenharia de Transportes e Logística do Centro Tecnológico de Joinville da Universidade Federal de Santa Catarina.

Orientadora: Dra. Silvia Lopes de Sena Tagliarenha

Joinville

2019

RUBENS CARLOS DE OLIVEIRA NETO

ESTUDO DE VIABILIDADE DE MODELO METROVIÁRIO SEM PARADAS

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do título de bacharel em Engenharia de Transportes e Logística, na Universidade Federal de Santa Catarina, Centro Tecnológico de Joinville.

Banca Examinadora:

Dra. Silvia Lopes de Sena Tagliapietra
Orientadora

Dr. Diogo Nardelli Siebert
1º Membro

Dra. Elisete Santos de Silva Zagheni
2º Membro

Se é inevitável termos fraquezas, e devemos enfim reconhecê-las como leis acima de nós, então desejo que cada um tenha força artística suficiente para tornar suas fraquezas o pano de fundo em que ressaltam suas virtudes, e, através de suas fraquezas, fazermos desejosos de suas virtudes.

-Nietzsche, Aurora, 1881.

AGRADECIMENTOS

Agradeço a toda minha família pelo apoio incondicional que sempre me deram.

RESUMO

O transporte metroviário é caracterizado por alto custo de implementação e relativo baixo custo de operação, além disso, proporciona benefícios sociais como redução de emissão de poluentes, porém, no Brasil, este modal é subutilizado em comparação com os modos de transporte rodoviários, como ônibus e automóveis. Por isso este estudo tem como objetivo analisar a viabilidade de uma nova forma de transporte metroviário, nomeado pelo autor de “Sem-paradas” neste estudo, onde o passageiro pode chegar ao seu destino, sem precisar parar em nenhuma estação intermediária. O estudo aborda este problema com ferramentas de Pesquisa Operacional. Realizam-se cálculos para definir os tempos de viagem teóricos e a capacidade de transporte de passageiros deste modelo sem paradas. Para otimizar o tempo de viagem, aplicou-se um método exato e um algoritmo genético, e obteve-se melhores resultados com o método exato, que conseguiu redução nos tempos de viagem de 10-30%. A capacidade máxima de transporte foi determinada a partir do *headway* mínimo encontrado com base nos tempos de viagem. Enquanto que o *headway* necessário para atender a demanda da linha metroviária estudada neste trabalho foi encontrado por meio de uma simulação.

Palavras-chave: Meta-heurística. Transporte metroviário. Otimização de atendimento.

ABSTRACT

Rail transport is known for high infrastructure costs and relatively low operation costs. It also generates social benefits like reduction of gas emission. In Brazil however this passenger transportation method has underwhelming ridership in comparison to road transportation, such as buses and automobiles. Therefore this study's aim is to test the feasibility of a new rail transport model, named *Non-stop* on this study. In this model a passenger wouldn't have to stop at any station in order to arrive at his destination. Operational research tools will be used as means to achieve optimal service time and capacity. The theoretical travel times and passenger carrying capacity of this model are measured. To optimize the travel time, an exact method and a genetic algorithm were applied, better results were obtained by the exact method, which achieved reduction in travel times of 10-30%. The maximum passenger carrying capacity was determined based on the minimum headway, which was found based on the travel times. While the headway needed to meet the demand of the subway line studied in this work was found through of a simulation.

Keywords: Metaheuristic. Rail transport. Service optimization.

LISTA DE FIGURAS

Figura 1 - Benefícios sociais do metrô de São Paulo em 2018	13
Figura 2 - Ilustração do modelo Sem-paradas	14
Figura 3 - Ilustração do modelo Sem-paradas	14
Figura 4 - Ilustração do modelo Sem-paradas	15
Figura 5 - Ilustração do modelo Sem-paradas	15
Figura 6- Movimento de veículos entre duas paradas consecutivas.....	18
Figura 7 – Conforto dos Passageiros	21
Figura 8 - Modelo AMPL.....	25
Figura 9 - Dados AMPL	26
Figura 10 – Comandos AMPL	26
Figura 11 – <i>Crossover</i>	28
Figura 12 - <i>Crossover</i> com dois pontos.....	29
Figura 13 - Mutação	29
Figura 14 - Fluxograma Algoritmo Genético	30
Figura 15 - Módulo <i>Create</i>	31
Figura 16 - Especificações <i>Create</i>	32
Figura 17 - Módulo <i>Hold</i>	33
Figura 18 - Módulo <i>Signal</i>	33
Figura 19 - Módulo <i>Delay</i>	33
Figura 20 - Módulo <i>Dispose</i>	34
Figura 21 - Fluxograma da Metodologia.....	35
Figura 22 - Seções de Movimento do Modelo Sem Paradas.....	41
Figura 23 – Parâmetros.....	45
Figura 24 - Algoritmo de Teste	47
Figura 25 - Gráfico Comparação de Tempo de Viagem Modelo Exato.....	49
Figura 26 - Classe População C++	50
Figura 27 - Gráfico para Comparação de Tempo de Viagem Modelo Genético.....	54
Figura 28- Fluxo dos Passageiros.....	58
Figura 29 - Fluxo dos Sinais.....	58
Figura 30 – <i>Headway</i> de 4 minutos	59
Figura 31 - <i>Headway</i> de 3 minutos.....	60
Figura 32 - <i>Headway</i> Mínimo	61

Figura 33 - Variação de Carros	67
Figura 34 - Variação de Paradas por Estação	67
Figura 35 - Variação de Paradas por Estação Parte 2.....	67
Figura 36 - Parada Total	68
Figura 37 - Algoritmo de Cálculo dos Tempos de Viagem.....	75
Figura 38 - Crossover Parte 1	76
Figura 39- Crossover Parte 2	77
Figura 40 - Mutação	78
Figura 41 - Codificação Binária	81

LISTA DE TABELAS

Tabela 1 - Posição das Estações	37
Tabela 2 – Solução Modelo Exato.....	46
Tabela 3 - Tempo de Viagem Modelo Real	48
Tabela 4 – Resultado Parcial Modelo Genético	52
Tabela 5 - Tempo de Viagem Modelo Genético	53
Tabela 6 - Fluxo de Passageiros Diários	55
Tabela 7 - Porcentagem de Passageiros <i>Shady Grove - Farragut North</i>	56
Tabela 8 - Tempo de Viagem <i>Shady Grove - Farragut North</i>	60
Tabela 9 - Tempos de Viagem com Headway Otimizado.....	63
Tabela 10 - Tempos de viagem Final	65
Tabela 11 - Resultado Exato.....	73
Tabela 12 - Resultado Modelo Exato (Continua).....	74
Tabela 13 - Resultado Modelo Genético	79
Tabela 14 - Resultado Modelo Genético (Continua).....	80

SUMÁRIO

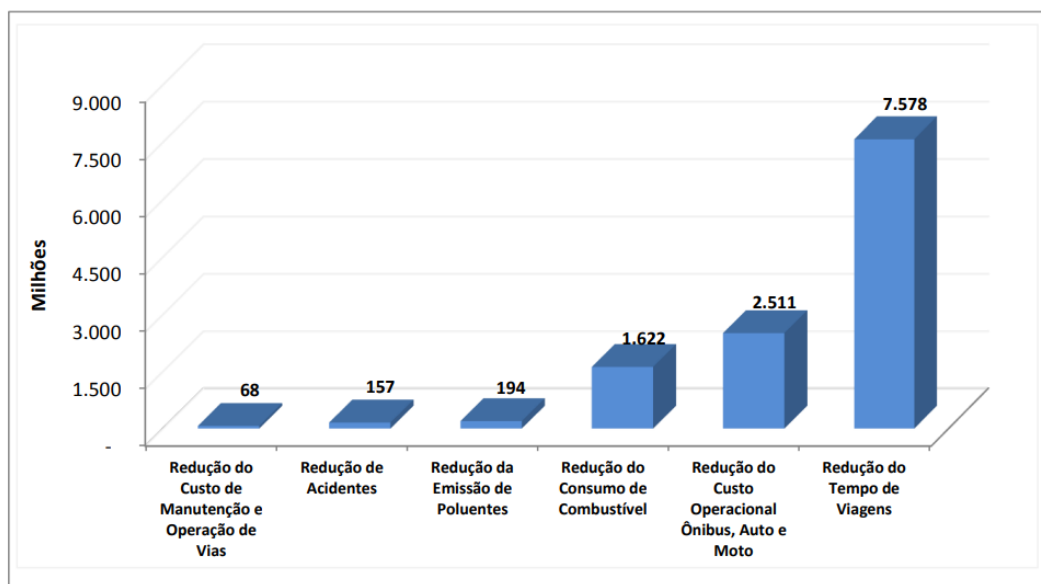
1 INTRODUÇÃO	13
1.1 OBJETIVO GERAL.....	16
1.2 OBJETIVOS ESPECÍFICOS	16
1.3 JUSTIFICATIVA	16
1.4 ESTRUTURA DO TRABALHO	16
2 FUNDAMENTAÇÃO TEÓRICA.....	18
2.1 PROPRIEDADES DO SISTEMA METROVIÁRIO	18
2.1.1 Equações de movimento.....	19
2.1.2 Especificações das locomotivas.....	21
2.2 CAPACIDADE DA LINHA	22
2.3 PROGRAMAÇÃO MATEMÁTICA	23
2.4 MÉTODOS EXATOS	25
2.5 META HEURÍSTICAS.....	27
2.5.1 Algoritmo Genético.....	27
2.6 SIMULAÇÕES.....	31
3 METODOLOGIA E OTIMIZAÇÃO DO MOVIMENTO DAS LOCOMOTIVAS ...	35
3.1 OTIMIZAÇÃO DO MOVIMENTO DAS LOCOMOTIVAS	36
3.1.1 Rota tipo 1	38
3.1.2 Rota tipo 2	39
3.1.3 Rota tipo 3	40
3.1.4 Estruturação do movimento das locomotivas	41
3.1.5 Modelagem matemática	42
3.5 MÉTODO DE SOLUÇÃO PARA O MODELO EXATO.....	44
3.5.1 Comparação com a situação real	47
3.6 ALGORITMO GENÉTICO	49
3.6.1 Comparação com a situação real	52
4 CAPACIDADE DE TRANSPORTE	55
4.1 SIMULAÇÃO DE ATENDIMENTO DE PASSAGEIROS NA LINHA	55
4.2 CAPACIDADE DE ATENDIMENTO MÁXIMA.....	61
5 PROPOSTAS DE CENÁRIOS ALTERNATIVOS	65
5.1 CENÁRIO 1 – VELOCIDADE MÁXIMA	65
5.2 CENÁRIO 2 – NÚMERO DE CARROS.....	66

5.3 CENÁRIO 3 – NÚMERO DE PARADAS POR ESTAÇÃO.....	67
5.4 CENÁRIO 4 – PARADA TOTAL.....	68
6 CONSIDERAÇÕES FINAIS.....	69
6.1 TRABALHOS FUTUROS.....	70
REFERÊNCIAS	71
APÊNDICE A – RESULTADO DO MODELO EXATO	73
APÊNDICE B – ALGORITMO DE CÁLCULO DOS TEMPOS DE VIAGEM.....	75
APÊNDICE C – ALGORITMO DE CROSSOVER	76
APÊNDICE D – ALGORITMO MUTAÇÃO	78
APÊNDICE E – RESULTADO MODELO GENÉTICO.....	79
APÊNDICE F – CODIFICAÇÃO BINÁRIA	81

1 INTRODUÇÃO

O modal metroviário, junto com suas variações, VLT e Trem suburbano, apresentam a maior capacidade de transporte de passageiros segundo Coca e Torres (2004). Além disso, segundo a Companhia do Metropolitano de São Paulo (2019), o sistema metroviário de São Paulo, gerou R\$ 12,1 bilhões de reais em benefícios sociais no ano de 2018, entre esses benefícios se encontram, redução do tempo de viagem e redução da emissão de poluentes como mostra a Figura 1.

Figura 1 - Benefícios sociais do metrô de São Paulo em 2018



Fonte: Companhia do Metropolitano de São Paulo (2019)

Porém vale ressaltar que o custo de instalação de uma linha metroviária segundo Vasconcellos (2012) é maior que o custo de instalação de vias asfaltadas, sendo que, vias segregadas de ônibus custam de 6 a 10 milhões de reais por quilômetro, enquanto que um metrô subterrâneo custa de 150 a 300 milhões de reais por quilômetro.

De acordo com Vasconcellos (2012) o consumo energético teórico por passageiro em lotação plena no modo de transporte metroviário é levemente superior ao de um ônibus

comum, sendo, 4,3 gramas equivalentes de petróleo (GEP) para o primeiro e, 4,1 GEP para o segundo. Enquanto que, para um automóvel, esse valor é igual a 19,3 GEP.

Sendo assim, a vantagem mais proeminente de se investir em metrô está em sua alta capacidade de transporte de pessoas, como apresentado em Vasconcellos (2012), um ônibus convencional consegue transportar em média, 80 pessoas, já um vagão consegue transportar 260 pessoas, contudo, o metrô de São Paulo, por exemplo, funciona com composições de 6 carros, portanto, comportando 1560 pessoas.

A fim de incentivar o uso deste modal, foi idealizada uma forma nova de executar os movimentos entre estações, onde o passageiro tem a possibilidade de chegar em sua estação destino sem a necessidade de parar em outras estações durante o trajeto.

As Figuras 2, 3, 4 e 5 ilustram como funciona este novo modelo, denominado pelo autor de Sem-paradas, que pode ser resumido da seguinte forma: Cada estação começa com uma locomotiva parada disponível para a entrada de passageiros, enquanto que, a primeira estação começa com duas locomotivas paradas, como mostra a Figura 2.

Figura 2 - Ilustração do modelo Sem-paradas



Fonte: Autor (2019)

As locomotivas da primeira estação entram em movimento, rumo à próxima estação. No meio do caminho as duas locomotivas se separam, a locomotiva traseira (Trem 1) diminui a sua velocidade a fim de parar na próxima estação, enquanto a locomotiva dianteira (Trem 2) continua em movimento uniformemente variado e ao mesmo tempo, a locomotiva que se encontrava parada na segunda estação (Trem 3) começa a se mover, como mostra a Figura 3.

Figura 3 - Ilustração do modelo Sem-paradas



Fonte: Autor (2019)

O objetivo é que a locomotiva provinda da primeira estação venha com velocidade suficiente para se emparelhar com a locomotiva proveniente da segunda estação, de forma que as duas consigam se conectar. Assim que conectadas, como na Figura 4, os passageiros podem se mover entre as locomotivas por um tempo a ser calculado.

Figura 4 - Ilustração do modelo Sem-paradas



Fonte: Autor (2019)

Após esse período as locomotivas se desconectam possibilitando que a locomotiva traseira (Trem 2) pare na terceira estação, enquanto que a locomotiva dianteira (Trem 3) siga em direção à quarta estação, e ao mesmo tempo, a locomotiva que se encontrava na terceira estação (Trem 4) entra em movimento para se emparelhar com a locomotiva provinda da segunda estação (Trem 3), como mostra a Figura 5. E dessa forma o processo se repete.

Figura 5 - Ilustração do modelo Sem-paradas



Fonte: Autor (2019)

Para calcular os tempos de viagem será utilizado um algoritmo genético, uma meta-heurística de Pesquisa Operacional idealizada por John Holland em seu trabalho *Adaptation in Natural and Artificial Systems*, capaz de consistentemente encontrar resultados muito próximos do resultado ótimo dado qualquer tipo de função ou restrição.

Usando os limites de velocidade e aceleração reais das locomotivas como parâmetro, consegue-se definir o tempo mínimo de viagem e, conseqüentemente, a capacidade da linha.

1.1 OBJETIVO GERAL

Testar a viabilidade e a capacidade ótima do modelo Sem-paradas, e compará-lo com o modelo atual empregado nos sistemas metroviários.

1.2 OBJETIVOS ESPECÍFICOS

Para alcançar o objetivo geral, também é necessário atingir os seguintes objetivos específicos:

- a) Formular um modelo algébrico que reflita o movimento das locomotivas entre estações;
- b) Desenvolver um modelo matemático que minimize o tempo de viagem global das locomotivas;
- c) Calcular a capacidade da linha metroviária criada com base no *headway*, intervalo entre viagens, obtido;
- d) Simular a linha metroviária para encontrar o nível de lotação das locomotivas.

1.3 JUSTIFICATIVA

Manter um sistema metroviário rentável em um país como o Brasil, voltado ao sistema rodoviário, não é uma tarefa fácil, prova disso está nos dados apresentados por Vasconcellos (2012), onde o metrô de São Paulo, teve déficit de 31,2 milhões de reais em 2007, enquanto que o sistema ferroviário de Fortaleza teve déficit de 8,5 milhões de reais no mesmo ano. Em compensação, com boa administração e eficiência, pode-se alcançar alta rentabilidade, por exemplo, o metrô do Rio de Janeiro teve lucro de 182,9 milhões de reais em 2007.

Por isso, entendendo as características do sistema metroviário, o objetivo desse trabalho é estudar a viabilidade da alteração do funcionamento do sistema de transporte metroviário, com a reconfiguração da forma de se realizar cargas e descargas de passageiros.

1.4 ESTRUTURA DO TRABALHO

Para alcançar o objetivo proposto, este trabalho foi organizado da seguinte maneira. O Capítulo 2 traz a fundamentação teórica, são apresentados os conceitos necessários para

melhor entendimento do trabalho. No Capítulo 3 apresenta-se a metodologia adotada e são detalhados os desenvolvimentos dos métodos de solução considerados.

No Capítulo 4 apresenta-se a simulação dos resultados alcançados no aplicativo Arena. No Capítulo 5 cenários alternativos são propostos. Por último no Capítulo 6 são apresentadas as considerações finais e sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo introduzir ao leitor a fundação teórica, que serviu como base para o desenvolvimento deste estudo, a saber: pesquisa operacional, meta-heurísticas, simulação, e características básicas de um sistema ferroviário.

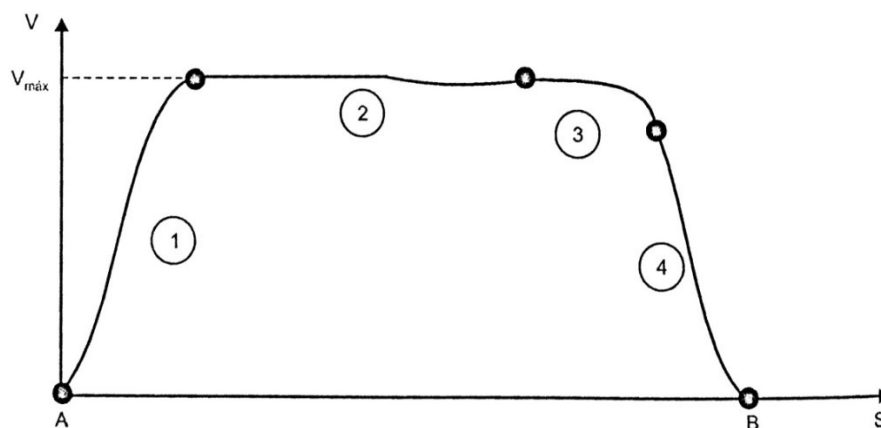
De acordo com Hillier e Lieberman (2015) a pesquisa operacional (PO) está em desenvolvimento desde meados da segunda guerra mundial, com o objetivo de alocar recursos escassos da forma mais eficiente possível.

Em Hillier e Lieberman (2015), problemas de PO são divididos em diversas categorias, como, programação linear, programação não linear, programação dinâmica e meta-heurísticas, entre outras. Entretanto, o conceito de modelagem matemática se aplica em todas as categorias citadas, e também nas categorias que foram utilizadas neste estudo. Portanto, a próxima seção, introduz este conceito.

2.1 PROPRIEDADES DO SISTEMA METROVIÁRIO

De acordo com Coca e Torres (2004), o movimento entre duas paradas de veículos de transporte público urbano em vias isoladas é constituído de quatro etapas distintas, como mostra a Figura 6.

Figura 6- Movimento de veículos entre duas paradas consecutivas



Fonte: Coca e Torres (2004)

Segundo Coca e Torres (2004):

A etapa 1 corresponde à fase de aceleração. O veículo (que pode ser uma única unidade ou um conjunto de unidades engatadas em comboio) parte com velocidade zero e é acelerado até atingir a velocidade máxima programada, normalmente fixada por razões de segurança. A aceleração é limitada tendo em conta os seguintes fatores: falta de aderência (atrito) das rodas motoras com a superfície de rolamento, restrição de potência dos motores e conforto dos passageiros. A aceleração imposta aos veículos não é, em geral, constante, embora a variação não seja grande (COCA; TORRES, 2004, p.157).

Portanto movimentos onde a aceleração é considerada constante recebem o nome de movimentos uniformemente variados (COCA; TORRES, 2004).

Já a etapa 2, é denominada fase de regime ou cruzeiro. Onde o veículo deve permanecer, tanto quanto possível, com velocidade máxima (COCA; TORRES, 2004).

A etapa 3 é denominada fase de *coasting* ou, movimento por inércia, onde normalmente ocorre uma pequena desaceleração (COCA; TORRES, 2004).

Por último, a etapa 4 é denominada fase de frenagem, na qual os freios são acionados até a parada no local desejado. “A desaceleração imposta aos veículos é limitada, tendo em vista o coeficiente de atrito roda-superfície de rolamento e o conforto dos passageiros.” (COCA; TORRES, 2004, p 159).

Portanto, assumindo que a variação da aceleração não é grande, como afirmado em Coca e Torres (2004), pode-se assumir que esta aceleração seja constante durante uma etapa. Neste caso, o movimento da locomotiva pode ser simplificado a um movimento unidimensional uniformemente variado. Portanto, em todos os momentos, a locomotiva se moverá em apenas uma direção com aceleração constante.

2.1.1 Equações de movimento

Considerando a informações da seção anterior, e levando em conta princípios básicos da dinâmica abordados em Hibbeler (2005), descritos nas equações (2.1) e (2.2). Pode-se encontrar as equações que regem o movimento das locomotivas. Portanto, considerando:

- x = Espaço;
- v = Velocidade;
- a = Aceleração;
- t = Tempo.

Pode-se afirmar que:

$$\frac{dx}{dt} = v \quad (2.1)$$

Ou seja, a velocidade é igual à derivada do espaço no tempo, ou em termos mais simples, a taxa de variação da posição no tempo (HIBBELER, 2005).

Similarmente, a aceleração é igual à derivada da velocidade no tempo (HIBBELER, 2005), ou seja:

$$\frac{dv}{dt} = a \quad (2.2)$$

Com isso, é possível modelar o deslocamento de uma locomotiva em uma seção.

Primeiramente, como citado anteriormente, pode-se declarar a aceleração como uma constante, ou seja, a aceleração não varia dentro de uma seção:

$$a = cte \quad (2.3)$$

Se a equação anterior for integrada em relação ao tempo, obtêm-se (HIBELLER, 2005):

$$v = a * t + c \quad (2.4)$$

Este c encontrado representa a velocidade inicial da locomotiva, logo:

$$\Delta v = a * t \quad (2.5)$$

Onde Δv representa a variação de velocidade, ou seja:

$$\Delta v = \textit{Velocidade Final} - \textit{Velocidade Inicial} \quad (2.6)$$

Agora, se novamente, a equação anterior for integrada em relação ao tempo, obtêm-se (HIBBELER, 2005):

$$x = \frac{a * t^2}{2} + v * t + k \quad (2.7)$$

Analogamente, o k representa a posição inicial da locomotiva, logo:

$$\Delta x = \frac{a * t^2}{2} + v * t \quad (2.8)$$

Onde:

$$\Delta x = \text{Posição Final} - \text{Posição Inicial} \quad (2.9)$$

V nas equações (2.7) e (2.8) representa a velocidade inicial.

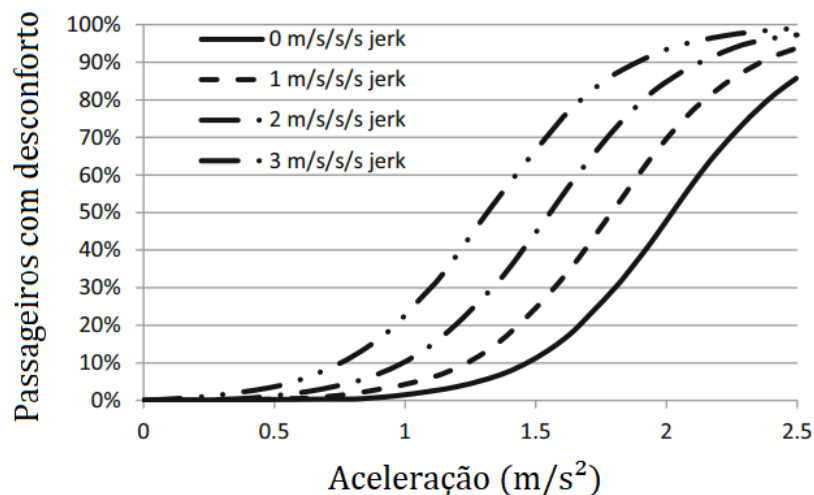
2.1.2 Especificações das locomotivas

A fim de encontrar parâmetros sobre o funcionamento de locomotivas, uma pesquisa bibliográfica foi realizada, e foi encontrado que, segundo Profillidis (2006):

- A aceleração varia entre 0,8 e 1 m/s²;
- A desaceleração é de mais ou menos 0,6 m/s²;
- Por motivos de fisiologia humana, a aceleração máxima não deve passar de 1,2 m/s²;
- Um parâmetro crítico para o conforto de passageiros é conhecido como arrancada, que representa a variação da aceleração por unidade de tempo, este valor não deve exceder 1,5 m/s³;
- A velocidade de um trem de passageiros varia entre 70 e 140 km/h.

Quanto ao conforto dos passageiros, segundo o estudo de (POWELL; PALACÍN, 2015), a aceleração da locomotiva tem relação direta quanto ao nível de conforto do usuário, essa relação é mostrada na Figura 7.

Figura 7 – Conforto dos Passageiros



Fonte: Adaptado de Powell e Palacín (2015)

Contanto, foi determinado durante o estudo, que ao se trabalhar com acelerações de até no máximo 1 m/s², é garantida a segurança e o conforto do passageiro.

Além disto, segundo Coca e Torres (2004), o tempo necessário de embarque e desembarque no modo metrôviário varia entre 20 e 40 segundos.

Enquanto que, segundo Hoel (2012), o tempo necessário de embarque e desembarque, pode ser calculado pela equação (2.10).

$$T_d = \frac{P_d * T_{pf}}{N_{cd}} + T_{oc} \quad (2.10)$$

Para:

- T_d = Tempo necessário de embarque e desembarque;
- P_d = Passageiros embarcando/desembarcando pela porta mais movimentada;
- T_{pf} = Tempo de fluxo (segundos/passageiros);
- T_{oc} = Tempo necessário para abrir e fechar as portas (normalmente 5s);
- N_{cd} = Número de portas.

Segundo (COCA; TORRES, 2004) os valores de T_{pf} variam entre 0,5 e 5 segundos.

E também, segundo a Companhia do Metropolitano de São Paulo (2019):

No sistema metroviário, 33% do total de embarques ocorrem nos períodos das 7h00 às 9h00 e das 17h00 às 19h00, caracterizados como horários de pico.

2.2 CAPACIDADE DA LINHA

Segundo (COCA; TORRES, 2004), a frequência de atendimento necessária para atender à demanda de uma linha de transporte, pode ser calculada a partir da segunda equação:

$$Q = \frac{P}{C} \quad (2.11)$$

Em que :

- P = Fluxo de passageiros na seção crítica (pass/h);
- C = Capacidade do veículo de transporte (pass/veíc);
- Q = Frequência de atendimento necessária (viag/h).

E, C pode ser definido da seguinte forma:

$$C = n * c \quad (2.12)$$

Com:

- n = Número de carros no comboio (carros);

- c = Capacidade de cada carro (pass/carro).

Além disso, pode-se definir a frequência de atendimento da seguinte forma (COCA; TORRES, 2004):

$$Q = \frac{60}{H} \quad (2.13)$$

Em que H = Intervalo entre viagens, ou *headway* (min/veíc).

Portanto, se faz necessário saber a capacidade de cada carro e o fluxo de passageiros em cada estação. A capacidade de cada carro foi disponibilizada por Committee (2019). Dois tipos de carros são produzidos pela companhia abordada nesse estudo:

- Carros do tipo A têm capacidade para 184 passageiros;
- Carros do tipo B têm capacidade para 192 passageiros.

2.3 PROGRAMAÇÃO MATEMÁTICA

Ao se resolver um problema, ou uma operação como o nome sugere, utilizando-se este método, é necessário usar uma abordagem científica, ou seja, independente da natureza do problema deve-se formular o problema (geralmente matematicamente) de forma que os recursos se tornem variáveis em uma equação. Essa equação tem como objetivo encontrar o estado hipotético onde os recursos são usados da maneira mais eficiente possível (ponto de máximo, ou mínimo da equação). Logo, fica a critério do pesquisador, definir a utilidade de cada uma das variáveis, e quais variáveis devem ser inclusas ou não, para que a formulação ‘capture’ a essência do problema real (HILLIER; LIEBERMAN, 2015).

Dado essa arbitrariedade, um mesmo problema pode ser resolvido de diversas formas, com resultados diferentes. Por isso, o pesquisador deve formular, resolver e checar resultados quantas vezes achar necessário, para garantir que esteja de fato criando hipóteses que se aproximam o máximo possível da realidade, sendo assim, de fato, uma pesquisa operacional (HILLIER; LIEBERMAN, 2015).

Segundo Fourer, Gay e Kernighan (2002) a programação matemática de um modelo exato pode ser resumida pelos seguintes passos:

- Formular um modelo matemático inicial que represente de forma geral o problema real;
- Coletar dados que definam as especificações do problema;

- iii. Definir uma função objetivo e restrições a partir do modelo inicial e dados coletados;
- iv. Resolver o problema utilizando um programa solver, que aplica um algoritmo capaz de encontrar os valores ótimos das variáveis;
- v. Analisar os resultados;
- vi. Aperfeiçoar os dados e o modelo o quanto for necessário.

As equações abaixo, adaptadas de Hillier e Lieberman (2015) representam um modelo de programação matemática simples.

$$\text{Maximizar Lucro} = 25x_1 + 30x_2 \quad (2.14)$$

Sujeito a:

$$\frac{1}{200}x_1 + \frac{1}{140}x_2 \leq 40 \quad (2.15)$$

A equação (2.14) descreve a função objetivo, onde o Lucro que varia linearmente em função das **variáveis de decisão** x_1 e x_2 deve ser maximizado. A equação (2.15), é chamada de **restrição**, e têm a função de limitar as variáveis de decisão. Essa restrição também contém uma constante, o número 40, essa constante é chamada de **parâmetro**.

O modelo (2.14)-(2.15) também pode ser representado na forma matricial (2.16)-(2.22) a seguir.

$$\text{Maximizar Lucro} = CX \quad (2.16)$$

Sujeito a:

$$AX \leq B \quad (2.17)$$

$$X \geq 0 \quad (2.18)$$

Em que:

$$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (2.19)$$

$$C = [25 \quad 30] \quad (2.20)$$

$$A = \begin{bmatrix} \frac{1}{200} & \frac{1}{140} \end{bmatrix} \quad (2.21)$$

$$B = [40] \quad (2.22)$$

O método de solução a ser utilizado em um problema de programação matemática depende de suas características, que em geral, levam em consideração conceitos de linearidade ou não das funções utilizadas na definição da função objetivo e nas restrições, bem como da quantidade e tipo de variáveis consideradas.

A seguir apresentam-se alguns métodos clássicos utilizados em PO.

2.4 MÉTODOS EXATOS

Existem vários modos de resolver um problema de programação matemática. Nesta seção é apresentada uma delas, a qual é considerada robusta por sua facilidade de implementação e adaptação em problemas complexos, a saber, a linguagem AMPL (*A Mathematical Programming Language*) (FOURER, GAY, KERNIGHAN, 2002).

Assim, considerando um problema de programação matemática que está apresentado na forma matricial (2.20)-(2.22) o mesmo pode ser escrito na linguagem AMPL, como apresentado na Figura 8.

Os dados de parâmetros do problema (A, B e C) podem ser introduzidos de forma separada do modelo, fazendo com que não seja necessário alterá-lo quando os parâmetros forem alterados.

Com isto, foi escrito o modelo em AMPL. É prática comum escrever modelos AMPL em três arquivos separados, o primeiro arquivo contendo o **modelo**, de forma genérica, o segundo arquivo contendo os **dados** específicos do problema, e o terceiro arquivo contendo os **comandos** a serem executados (FOURER; GAY; KERNIGHAN, 2002).

A Figura 8 representa o arquivo de modelo.

Figura 8 - Modelo AMPL

```

set P;
param B;
param A {j in P};
param C {j in P};
var X {j in P};
maximize Total_Lucro: sum {j in P} C[j] * X[j];
subject to Time: sum {j in P} (A[j]) * X[j] <= B;
subject to Constraint: {j in P}: X[j] >= 0;

```

Fonte: Fourer, Gay e Kernighan (2002)

Os parâmetros C e A são colunas da matriz P , pois estão declarados como $\{j \text{ in } P\}$. As variáveis X também estão representadas em uma coluna da matriz P . O parâmetro B representa uma constante (FOURER; GAY; KERNIGHAN, 2002).

Ademais, a função objetivo é descrita em *maximize*, onde é estabelecido que o objetivo é maximizar o lucro total, encontrado pelo somatório (*sum*) de $C[j] * X[j]$, onde $C[j]$ e $X[j]$ são os valores encontrados na posição j de suas respectivas colunas (FOURER; GAY; KERNIGHAN, 2002).

Já as restrições são descritas por *subject to*, a segunda restrição limita as variáveis $X[j]$, a serem maiores ou iguais a zero. Enquanto a primeira restrição funciona de maneira análoga (FOURER; GAY; KERNIGHAN, 2002).

Também é necessário definir os parâmetros que foram declarados, esta definição deve ser feita no arquivo de dados, mostrado na Figura 9.

Figura 9 - Dados AMPL

```
set P := 1 2;
param: A C :=
1 200 25
2 140 30;
param B := 40;
```

Fonte: Adaptado de Fourer, Gay e Kernighan (2002)

Na Figura 9 o número de linhas da matriz P é definido com os índices 1 e 2. Além disso, são definidos os valores dos parâmetros declarados no arquivo de modelo. (FOURER; GAY; KERNIGHAN, 2002).

Na Figura 10 está representado o arquivo de comando, o comando solve faz com que o solver resolva o modelo apresentado no arquivo de modelo, enquanto que o comando display faz com que o solver exiba os valores finais das variáveis $X[j]$ (FOURER; GAY; KERNIGHAN, 2002).

Figura 10 – Comandos AMPL

```
solve;
display X;
```

Fonte: Adaptado de Fourer, Gay e Kernighan (2002)

Como foi afirmado no passo *iv*, para resolver o problema modelado, é necessário utilizar um solver. Neste estudo o *solver Lancelot* desenvolvido em Toint, Conn e Gould (1992), disponível na plataforma online *Neos Solvers* foi escolhido. Esta plataforma online possui diversos métodos de otimização já implementados, e computadores de alta performance, capazes de processar dados rapidamente.

O solver Lancelot resolve problemas de restrições não-lineares (TOINT; CONN; GOULD, 1992). A explicação de como funciona este método não faz parte do escopo deste estudo. Portanto o capítulo 4 se concentra no desenvolvimento do modelo matemático.

2.5 META HEURÍSTICAS

Métodos Heurísticos são procedimentos utilizados para encontrar soluções viáveis para problemas de PO, estes métodos não possuem garantia de otimalidade (HILLIER; LIEBERMAN, 2013).

Ainda, de acordo com Hillier e Lieberman (2015):

O procedimento também deve ser suficientemente capaz para lidar com problemas muito grandes. O procedimento normalmente é um algoritmo iterativo completo em que cada iteração envolve a condução da procura de uma nova solução que poderia ser melhor que a melhor solução encontrada previamente. Quando o algoritmo termina após um tempo razoável, a solução por ele fornecida é a melhor que foi encontrada durante qualquer iteração (HILLIER; LIEBERMAN, 2015, p. 581).

A qualidade dos resultados depende do quão robusto é o algoritmo criado. Logo, um algoritmo robusto é aquele com resultados que se aproximam ao máximo da realidade (HILLIER; LIEBERMAN, 2015).

As meta heurísticas geralmente se baseiam em conceitos de fácil entendimento, como abordado em Hillier e Lieberman (2015), que muitas vezes imitam algum fenômeno natural. As meta heurísticas mais comuns são, *Tabu Search*, *Simulated Annealing* e *Genetic Algorithms*. Neste trabalho o algoritmo genético é utilizado.

2.5.1 Algoritmo Genético

Os Algoritmos Genéticos (GAs) são algoritmos de pesquisa baseados nos mecanismos de seleção natural e adaptação. Estes algoritmos operam com um conjunto de possíveis soluções, denominado “população”, para determinação do ótimo. Os elementos da população

são combinados através de uma troca de informações de modo a encontrar melhores soluções que as da população original (HOLLAND, 1975).

A troca de informações para determinação da nova população depende do valor da função objetivo, de tal forma que melhores soluções possuem maior possibilidade de serem combinadas. Desse processo obtêm-se, em geral, soluções melhores que as originais e toda a população desloca-se em direção ao ótimo (HOLLAND, 1975).

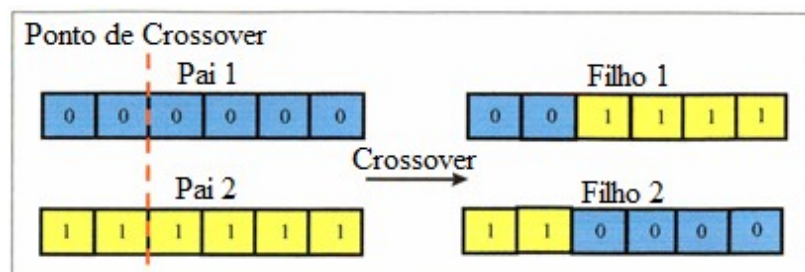
O primeiro Algoritmo Genético foi descrito por Holland (1975) e, foi inspirado em um mecanismo simplificado da adaptação natural dos seres vivos ao ambiente. A adaptação pode ser considerada um processo de modificação progressiva de uma população promovendo um melhor desempenho no ambiente. O ambiente, neste caso, é a função objetivo a ser otimizada.

Segundo Mitchell (1998) existem quatro operadores essenciais em um algoritmo genético. O primeiro é o *fitness*, que pode ser traduzido como adaptabilidade, é normalmente representado pelo resultado da função objetivo, e serve para calcular a qualidade do resultado obtido.

O segundo operador é a seleção, que envolve escolher quais indivíduos são aptos para passarem pelo processo de reprodução. Nesta etapa o fitness de cada indivíduo é calculado, e indivíduos com maior adaptabilidade são escolhidos para a reprodução com maior frequência (MITCHELL, 1998).

O terceiro operador é o *crossover*, que pode ser traduzido como recombinação, e corresponde ao processo de reprodução. Nesta fase, uma nova geração de indivíduos deve ser criada. Como mostra a Figura 11, um novo indivíduo é criado quando se recombina genes de outros dois indivíduos (MITCHELL, 1998).

Figura 11 – *Crossover*

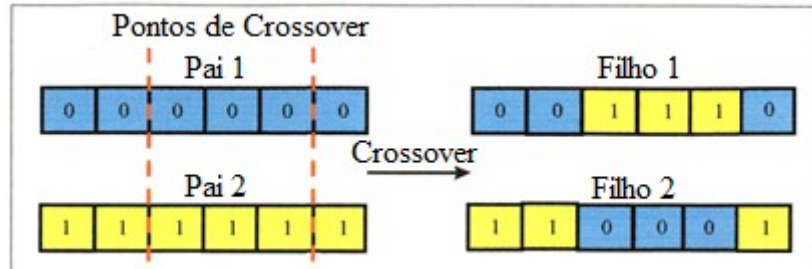


Fonte: Adaptado de Claumann (1999)

Essas recombinações podem acontecer de diversas formas. No exemplo da Figura 11, a recombinação ocorreu em um ponto entre o segundo e o terceiro gene, mas Mitchell (1998)

sugere que em cada *crossover* este ponto seja escolhido aleatoriamente. E que mais de um ponto seja escolhido, como mostra a Figura 12.

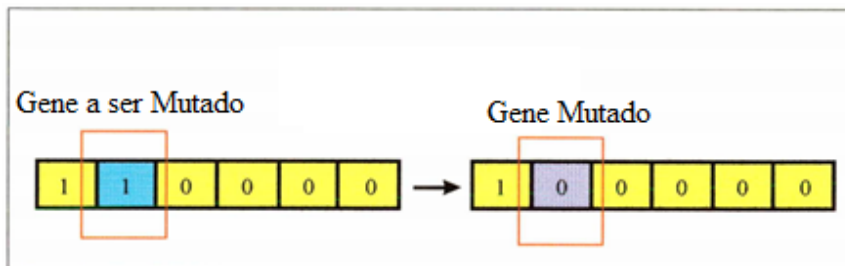
Figura 12 - *Crossover* com dois pontos



Fonte: Adaptado de Claumann (1999)

O quarto operador é a mutação. Mutações podem ocorrer aleatoriamente em qualquer parte do código genético, porém normalmente tem baixa probabilidade de ocorrência (e.g., 0.001), e podem introduzir características que não existiam em nenhum dos membros da população (MITCHELL, 1998). Por exemplo, um indivíduo com código genético 110000 pode passar a ter o código genético 100000 após uma mutação, como mostra a Figura 13.

Figura 13 - Mutação



Fonte: Adaptado de Claumann (1999)

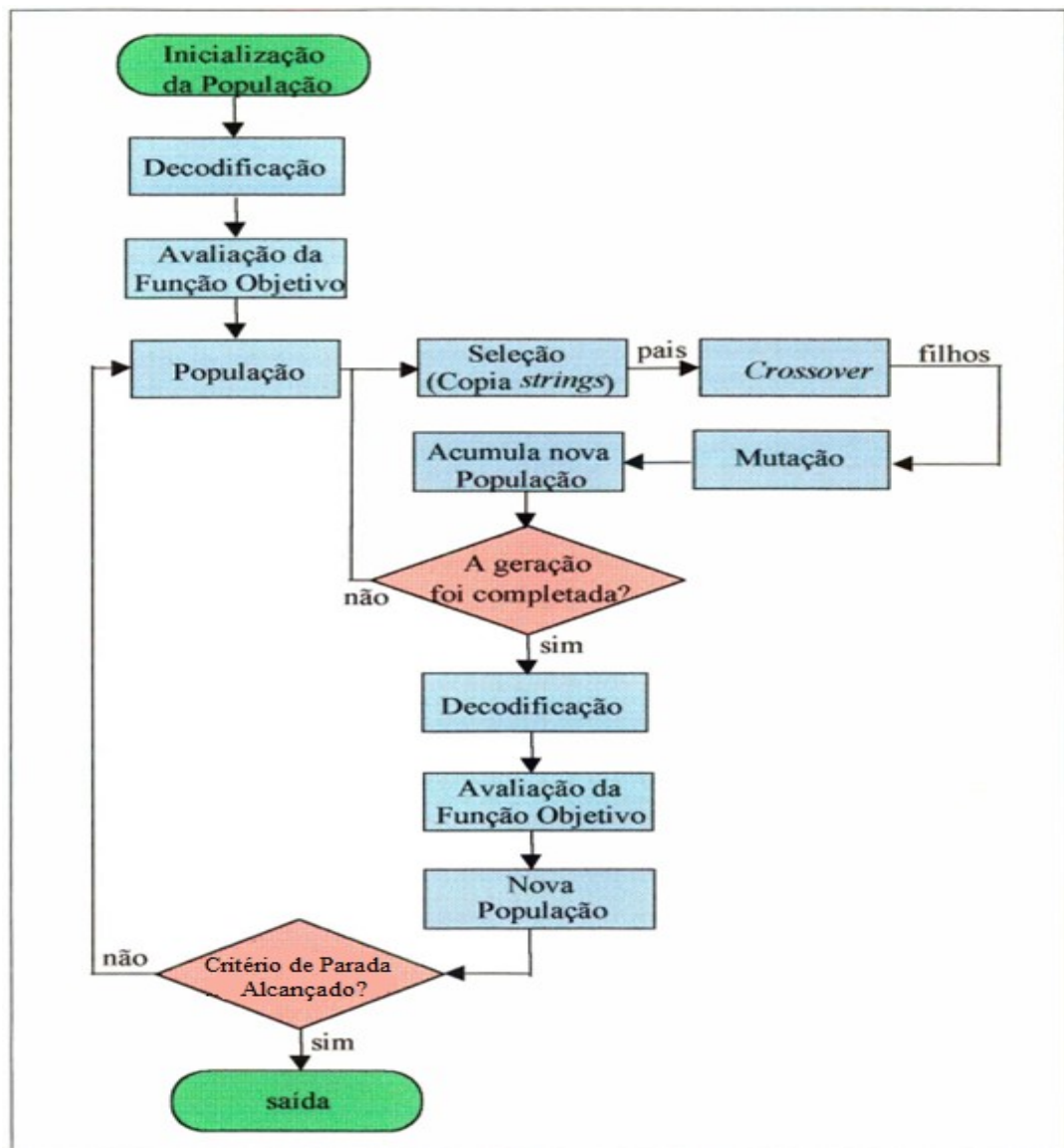
Além disso, também define-se os seguintes conceitos (MITCHELL, 1998):

- População: o tamanho da população deve ser escolhido pelo pesquisador, e define quantos indivíduos existem em cada geração.
- O critério de parada: também deve ser escolhido pelo pesquisador, este define quando o algoritmo deve parar, e pode ser feito de diversas maneiras, uma delas é definir um número máximo de gerações que devem ser concebidas até que o algoritmo pare.

- Codificação e Decodificação: Para aplicar os operadores mostrados anteriormente, é necessário codificar os valores das variáveis de decisão. Normalmente a codificação envolve a troca de base da variável, entre binária e decimal.

Um fluxograma de um algoritmo genético padrão, com os operadores e conceitos explicados anteriormente, pode ser observado na Figura 14.

Figura 14 - Fluxograma Algoritmo Genético



Fonte: Adaptado de Claumann (1999)

Utilizando esses conceitos, o algoritmo deve melhorar a adaptabilidade média da população a cada iteração (MITCHELL, 1998).

2.6 SIMULAÇÕES

A simulação consiste em criar uma representação de um sistema complexo a fim de fornecer previsões das medidas de desempenho de interesse (ALTIOK; MELAMED, 2007).

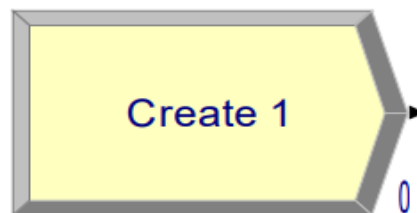
O software de simulação Arena foi desenvolvido pela empresa *Rockwell Automation*, com o objetivo de desenvolver modelos que descrevam aplicações reais utilizando um conjunto de módulos capaz de representar as atividades de processamento, separação, transporte, recebimento, entre outros (CARDOSO et al., 2016).

Os elementos básicos de um modelo desenvolvido no Arena são as entidades que se movem ao longo do sistema, as estações de trabalho, ou módulos, que representam onde será prestado um determinado serviço ou atividade, e o fluxo que representa o percurso que as entidades irão percorrer ao longo do sistema (PRADO, 1999).

Os seguintes módulos apresentados por Pereira (2009) serão utilizados neste estudo:

- *Create*;
- *Hold*;
- *Signal*;
- *Delay*;
- *Dispose*.

Figura 15 - Módulo *Create*



Fonte: Pereira (2009)

O módulo da Figura 15 destina-se como ponto de partida para as entidades em um modelo de simulação. As entidades partem deste módulo para iniciar o processamento através do sistema. O tipo de entidade é especificado neste módulo (Pereira, 2009). Ao clicar duas vezes neste módulo, aparecerá a caixa de entrada mostrada na Figura 16.

Figura 16 - Especificações *Create*

The image shows a 'Create' dialog box with the following fields and values:

- Name:** Create 1
- Entity Type:** Entity 1
- Time Between Arrivals:**
 - Type:** Random (Expo)
 - Value:** 1
 - Units:** Hours
- Entities per Arrival:** 1
- Max Arrivals:** Infinite
- First Creation:** 0.0

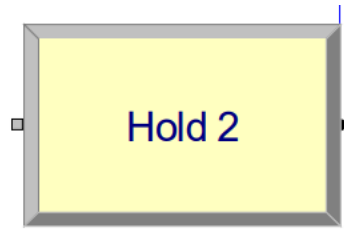
Buttons: OK, Cancel, Help

Fonte: Pereira (2009)

A Figura 16 pode ser resumida da seguinte forma segundo Pereira (2009):

- *Name*: Nome da entidade de entrada.
- *Entity Type*: Tipo da entidade.
- *Time Between Arrivals*: Tempo entre chegadas.
- *Type*: Tipo de fluxo a ser gerado. Os tipos incluem: *Random* (utiliza uma distribuição exponencial, o usuário especifica média), *Normal* (uma distribuição normal é aplicada) entre outros.
- *Value*: Determina a média da distribuição exponencial ou a média e o desvio padrão da distribuição normal para o tempo entre chegadas.
- *Units*: Unidade de tempo utilizada.
- *Entities per Arrival*: Número de entidades que entrarão no sistema em um tempo determinado no momento de chegada.
- *Max Arrival*: Número máximo de entidades que este módulo irá gerar. Quando este valor for atingido, a criação de novas entidades por este módulo cessa.
- *First Creation*: Hora de início para a primeira entidade a chegar ao sistema.

O módulo *Hold* da Figura 17 controla uma entidade em uma fila de modo a esperar por um sinal, que pode ser mandado pelo módulo *Signal*, apresentado na Figura 18.

Figura 17 - Módulo *Hold*

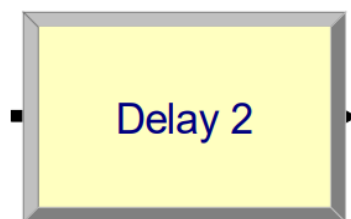
Fonte: Pereira (2009)

Já o módulo *Signal* apresentado na Figura 18 envia um sinal para cada módulo *Hold* do modelo, que então checa o valor do sinal e caso este valor seja o mesmo valor definido no módulo *Hold*, este libera as entidades.

Figura 18 - Módulo *Signal*

Fonte: Pereira (2009)

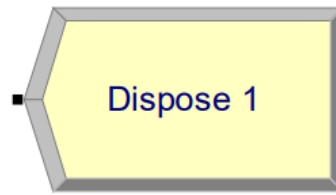
O módulo *Delay* da Figura 19 especifica um período de tempo de atraso para a entidade. Quando uma entidade chega a um módulo *Delay*, a expressão de tempo de atraso é avaliada e a entidade permanece no módulo pelo período de tempo pré-determinado.

Figura 19 - Módulo *Delay*

Fonte: Pereira (2009)

Por último, o módulo *Dispose* da Figura 20 é utilizado como ponto final para as entidades em um modelo de simulação.

Figura 20 - Módulo *Dispose*



Fonte: Pereira (2009)

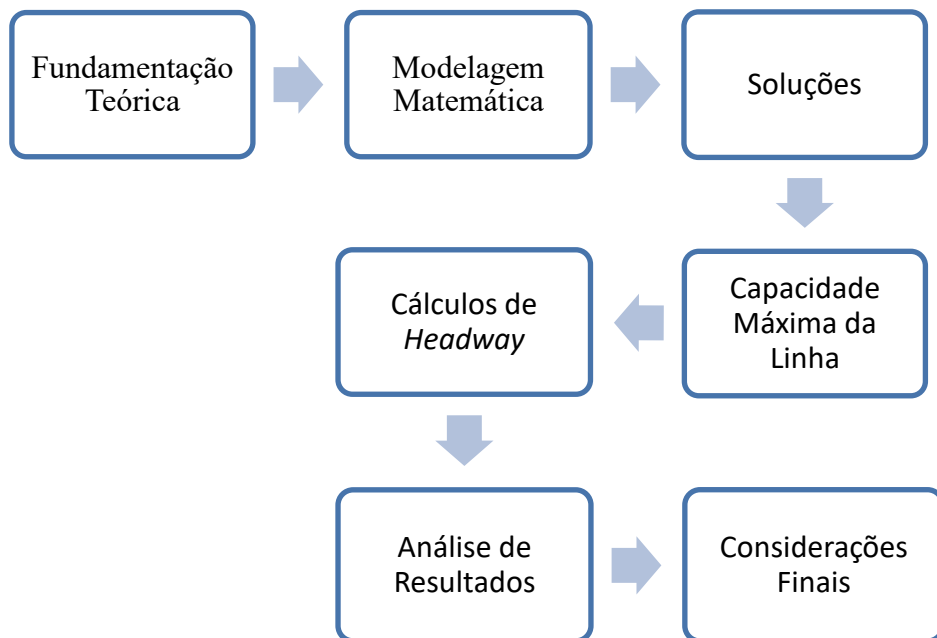
Por último, com o tempo de viagem das locomotivas já otimizado, uma simulação da linha de trem proposta será realizada com o auxílio da ferramenta Arena, o objetivo desta simulação é encontrar a capacidade de transporte do modo Sem-paradas.

No próximo capítulo a metodologia aplicada neste trabalho é apresentada.

3 METODOLOGIA E OTIMIZAÇÃO DO MOVIMENTO DAS LOCOMOTIVAS

Para alcançar os objetivos propostos neste trabalho, buscou-se utilizar os procedimentos que podem ser verificados na Figura 21.

Figura 21 - Fluxograma da Metodologia



Fonte: Autor (2019)

- **Fundamentação Teórica:**
Foi realizada uma pesquisa bibliográfica, sobre os seguintes temas: pesquisa operacional, características do transporte público urbano e metroviário, entre outros.
- **Modelagem Matemática do Movimento das Locomotivas:**
Com base na definição de movimento entre duas paradas (COCA; TORRES, 2004), e princípios básicos de movimento de partículas, definiu-se como o movimento das locomotivas seria interpretado matematicamente.

- Solução do Modelo Matemático:
Para garantir a qualidade dos resultados obtidos, dois métodos de solução foram aplicados, o método exato e a meta heurística algoritmo genético.
 - a) Método Exato:
O modelo matemático desenvolvido foi reescrito em linguagem matemática de programação, e com o objetivo de minimizar os tempos de viagens da linha metroviária, o modelo executado pelo *solver Lancelot*, presente na plataforma online *Neos Solvers*.
 - b) Algoritmo Genético:
Desenvolveu-se um algoritmo genético com o objetivo de minimizar os tempos de viagens da linha metroviária. Os resultados deste algoritmo foram comparados com os resultados do Método Exato, e o método com maior qualidade nos resultados foi utilizado nos cálculos posteriores.
- Cálculos de Capacidade Máxima da Linha:
Com base nos dados obtidos pela otimização dos tempos de viagem da linha metroviária, e na definição de capacidade da linha de Coca e Torres (2004), calculou-se qual seria a capacidade máxima da linha em modo Sem-paradas, se aplicada a um trecho da linha metroviária *Red Line* de Washington D.C.
- Cálculos de *Headway*:
Com o auxílio do aplicativo de simulação *Arena*, encontrou-se qual seria o *headway* mínimo necessário para atender a *Red Line* no modo Sem-paradas.
- Análise de resultados:
Os dados obtidos foram analisados, e observações sobre o modo Sem-paradas foram realizadas.
- Considerações Finais:
É feita a conclusão geral do trabalho e indicações para trabalhos futuros.

3.1 OTIMIZAÇÃO DO MOVIMENTO DAS LOCOMOTIVAS

Nesta seção são detalhadas as etapas da aplicação dos métodos considerados para encontrar a forma mais eficiente de se operar uma linha em modo Sem-paradas.

Consideraram-se dados de um trecho da linha de trem *Red line* de Washington, D.C, apresentados na Tabela 1.

Tabela 1 - Posição das Estações

Estação	Posição (m)	Tempo de viagem (minutos)
Farragut North	0	0
Dupont Circle	643	2
Woodley Park-Zoo	2253	4
Cleveland Park	3701	6
Van Ness-UDC	4667	8
Tenleytown-AU	6115	10
Friendship Heights	7563	12
Bethesda	9977	15
Medical Center	11748	18
Grosvenor Metro	14966	21
White Flint	17059	24
Twinbrook	18829	27
Rockville	22047	31
Shady Grove	26232	35

Fonte: Google Maps (2019)

O lado esquerdo apresenta o nome das estações, enquanto que a coluna central indica a distância da respectiva estação em relação à primeira estação da lista. Por último a coluna da direita representa o tempo de viagem da respectiva estação em relação à primeira estação da lista.

Esta linha foi escolhida pois as informações sobre a quantidade de passageiros que acessam as estações são disponibilizadas abertamente pela agência governamental WMATA de Washington D.C. (WMATA, 2018).

Já o trecho selecionado é distinto, pois apenas a Red line atende a estas estações, o que garante que todos os passageiros que acessam a estação estão usando a linha em estudo.

Antes da aplicação dos modelos é necessário entender as leis da física que regem o movimento dos trens. E para isso, o movimento realizado pela locomotiva entre estações foi dividido em rotas, portanto as seções 4.1, 4.2 e 4.3 têm como objetivo apresentar as características de cada tipo rota.

Nas rotas tipo 1 e 2, pressupõe-se que as posições iniciais e finais e as velocidades iniciais e finais das locomotivas já estão definidas, pois com isto, é possível calcular a aceleração e tempo necessários para garantir a factibilidade do movimento.

Já na rota de tipo 3, a posição final não é conhecida de antemão, e portanto é necessário pré-definir o intervalo de tempo, para calcular a posição final e aceleração da locomotiva.

3.1.1 Rota tipo 1

É caracterizado por uma locomotiva que parte do repouso, em uma estação, e acelera com aceleração constante por um período de tempo, semelhante à etapa 1 da Figura 6 no capítulo 2.1.

Este deslocamento apresenta algumas particularidades:

- Velocidade inicial = 0;
- Posição inicial conhecida.

Logo, se for definida uma velocidade final e a posição final para esta rota, consegue-se calcular as demais variáveis, como mostra o exemplo a seguir.

Exemplo 1: Considere uma locomotiva partindo do repouso da estação *Rockville*, indicada na Tabela 1, a velocidade final tem valor de $30 \frac{m}{s}$, e sua posição final tem valor de 4700m.

Primeiramente substituem-se os dados disponíveis na equação (2.5), e tem-se:

$$30 \frac{m}{s} - 0 = a * t \quad (3.1)$$

E portanto,

$$t = \frac{30 \frac{m}{s}}{a} \quad (3.2)$$

Agora, é possível substituir t na equação (2.12), resultando:

$$4700 - 4184,284m = \frac{a * \left(30 \frac{m}{s}\right)^2}{2} + 0 \frac{m}{s} * \frac{30 \frac{m}{s}}{a}. \quad (3.3)$$

Ou seja,

$$515,716 = \frac{450}{a} + 0. \quad (3.4)$$

Assim,

$$a = 0,8725 \frac{m}{s^2}. \quad (3.5)$$

E agora, o valor de a encontrado é substituído na equação (3.2), e obtêm-se:

$$t = \frac{30 \frac{m}{s}}{0,8725 \frac{m}{s^2}} \quad (3.6)$$

Ou seja,

$$t = 34,38 \text{ s}. \quad (3.7)$$

Com isso, observa-se que, para esse tipo de rota, é possível estabelecer as variáveis v e x como variáveis de decisão, e a partir delas, encontrar o valor das demais variáveis.

3.1.2 Rota tipo 2

É caracterizada por uma locomotiva já em movimento, que deve parar na próxima estação, semelhante à etapa 4 da Figura 6 no capítulo 2.1. Neste caso, a posição e velocidade final já são conhecidas, e se a velocidade e posição inicial forem definidas, consegue-se calcular a aceleração e tempo, como mostra o exemplo a seguir:

Exemplo 2: Considerando uma locomotiva em movimento, partindo da posição 8500m, com destino à estação *White Flint*, com velocidade inicial igual a $25m/s$, é possível calcular o intervalo de tempo e aceleração necessária para garantir que a locomotiva pare na estação.

Para isso, primeiramente substituem-se os valores informados na equação (2.5):

$$0 - 25 \frac{m}{s} = a * t. \quad (3.8)$$

Consequentemente, tem-se:

$$t = \frac{-25 \frac{m}{s}}{a} \quad (3.9)$$

E então, se o valor de t for substituído na equação (2.12):

$$9173,238m - 8500m = \frac{a * \left(\frac{-25 \frac{m}{s}}{a}\right)^2}{2} + \left(25 \frac{m}{s} * \left(\frac{-25 \frac{m}{s}}{a}\right)\right). \quad (3.102)$$

O que implica que:

$$673,238 = \frac{312,5}{a} - \frac{625}{a} \quad (3.11)$$

E portanto,

$$a = -0,46 \frac{m}{s^2} \quad (3.12)$$

Substituindo na equação (3.9):

$$t = 53,86s \quad (3.13)$$

Portanto, a Rota Tipo 2 é análoga a Rota Tipo 1, a única diferença sendo os valores de velocidade final e inicial.

3.1.3 Rota tipo 3

Este tipo de deslocamento é o mais comum neste estudo, e será usado em qualquer seção que não atenda os critérios de movimento tipo 1 ou 2. O movimento é caracterizado por uma locomotiva que tem velocidade inicial e final maior que zero, semelhante à etapa 2 da Figura 6 do capítulo 2.1. Neste caso assume-se que a velocidade e posição inicial são conhecidas, e define-se uma velocidade final e um intervalo de tempo, como mostra o exemplo:

Exemplo: Considere uma locomotiva em movimento partindo da posição 10000m a $15 \frac{m}{s}$, ela continua em movimento por 25s até que alcança a velocidade final de $20 \frac{m}{s}$, considerando aceleração constante, calcule a posição final, e aceleração da locomotiva.

Primeiramente substituem-se os valores na equação (2.5):

$$25s = \frac{20 \frac{m}{s} - 15 \frac{m}{s}}{a}. \quad (3.14)$$

$$a = \frac{5 \frac{m}{s}}{25s}. \quad (3.15)$$

$$a = 0,2 \frac{m}{s^2}. \quad (3.16)$$

Desta vez consegue-se o valor da aceleração, e se a for substituído na equação (2.9):

$$Posição\ Final - 10000 = \frac{0,2 \frac{m}{s^2} * (25s)^2}{2} + 15 \frac{m}{s} * 25s. \quad (3.17)$$

$$Posição\ Final = 10437,5m. \quad (3.18)$$

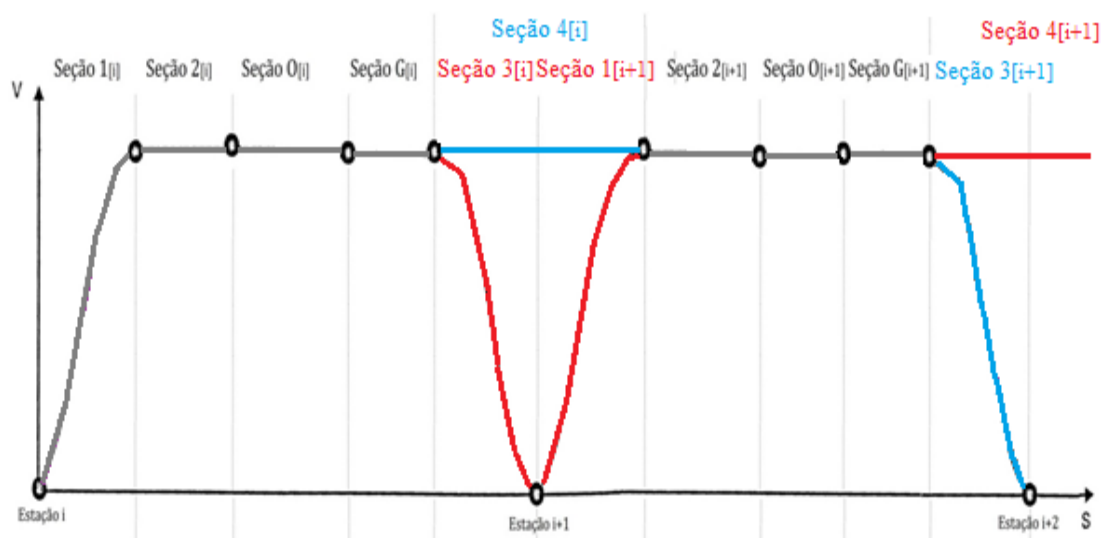
Este caso é diferente dos anteriores, pois aqui, é preciso saber o intervalo de duração da rota de antemão, enquanto o valor da posição final é desnecessário. A seção 3.1.4 aborda as implicações dessa diferença.

3.1.4 Estruturação do movimento das locomotivas

O objetivo deste capítulo é determinar como a locomotiva deve se movimentar durante todo o percurso, de forma a minimizar o tempo de viagem total do sistema.

A Figura 22 representa o movimento de duas locomotivas partindo juntas da Estação i , para $i \in N, N = \{1, \dots, N\}$. Na Figura 22, traços de cor cinza indicam um movimento realizado por ambas locomotivas. Enquanto que traços azuis indicam movimentos realizados apenas pela locomotiva dianteira, e traços vermelhos indicam movimentos realizados apenas pela locomotiva traseira.

Figura 22 - Seções de Movimento do Modelo Sem Paradas



Fonte: Adaptado de Coca e Torres (2004)

Os movimentos possíveis de qualquer locomotiva entre estações foram divididos em seis seções: 1, 2, O, G, 3 e 4, como podem ser vistos na Figura 22. E com base nos cálculos feitos anteriormente, foi definido que a Seção 1 é uma rota de tipo 1, enquanto a seção 3 é uma rota de tipo 2. Já as seções 2, O, G e 4, são rotas de tipo 3.

As trocas de locomotiva realizadas pelos passageiros ocorrem na seção 2, e, por isso, o tempo de duração dessa seção é pré-definido, para garantir que haja tempo para as trocas.

A seção 4 é percorrida apenas pela locomotiva dianteira, pois esta não pára na Estação $i + 1$. Logo, no início da seção 4, as duas locomotivas que partiram da Estação i se separam, enquanto que no final da seção 4, a locomotiva dianteira se conecta com a locomotiva que parte da Estação $i + 1$, se tornando uma locomotiva traseira.

Já as seções O e G têm a função de preencher espaços vazios do trajeto da locomotiva, portanto nenhuma das duas é indispensável, porém sem a existência dessas seções, seria impossível para uma locomotiva percorrer por estações muito distantes umas das outras, dado que as seções 1 e 3 exigem aceleração, e a seção 2 tem duração limitada.

3.1.5 Modelagem matemática

Para solucionar o modelo exato, primeiramente é fundamental descrever o modelo algébrico, no caso, o objetivo desta etapa é minimizar o tempo de operação da linha respeitando as restrições de movimento das locomotivas.

Com isto o seguinte modelo foi desenvolvido:

$$\text{Minimizar } \sum_{i=1}^n (T1_i + T2_i + T3_i + TO_i + TG_i) + \sum_{i=1}^{n-1} T4_i \quad (3.3)$$

Sujeito a:

$$T1_i = \frac{V1_i}{A1_i} \quad \forall i \in \mathbb{N} \quad (3.4)$$

$$T2_i = \frac{V2_i - V1_i}{A2_i} \quad \forall i \in \mathbb{N} \quad (3.5)$$

$$TO_i = \frac{VO_i - V2_i}{AO_i} \quad \forall i \in \mathbb{N} \quad (3.6)$$

$$TG_i = \frac{VG_i - VO_i}{AG_i} \quad \forall i \in \mathbb{N} \quad (3.7)$$

$$T3_i = \frac{V3_i - ((VG_i * (1 - bin1_i)) + (VO_i * bin1_i))}{A3_i} \quad \forall i \in \mathbb{N} \quad (3.8)$$

$$T4_i = \frac{V4_i - ((VG_i * (1 - bin1_i)) + (VO_i * bin1_i))}{A4_i} \quad \forall i = 1, \dots, n - 1 \quad (3.9)$$

$$X1_i = 0,5 * A1_i * T1_i^2 \quad \forall i \in \mathbb{N} \quad (3.10)$$

$$X2_i = (0,5 * A2_i * T2_i^2) + (V1_i * T2_i) \quad \forall i \in \mathbb{N} \quad (3.11)$$

$$X3_i = (0,5 * A3_i * T3_i^2) + (((VG_i * (1 - bin1_i)) + (VO_i * bin1_i)) * T3_i) \quad \forall i \in N \quad (3.12)$$

$$XO_i = (0,5 * AO_i * TO_i^2) + (V2_i * TO_i) \quad \forall i \in N \quad (3.13)$$

$$XG_i = (0,5 * AG_i * TG_i^2) + (VO_i * TG_i) \quad \forall i \in N \quad (3.14)$$

$$X4_i = (0,5 * A4_i * T4_i^2) + (((VG_i * (1 - bin1_i)) + (VO_i * bin1_i)) * T4_i) \quad \forall i \in N \quad (3.15)$$

$$XG_i \leq (1 - bin1_i) * 100000 \quad \forall i \in N \quad (3.16)$$

$$Distancia_i = X1_i + X2_i + XO_i + XG_i + X3_i \quad \forall i \in N \quad (3.17)$$

$$V4_i = V1_{i+1} \quad \forall i = 1, \dots, n - 1 \quad (3.18)$$

$$X4_i = X1_{i+1} + X3_i \quad \forall i = 1, \dots, n - 1 \quad (3.19)$$

$$0 \geq V1_i, V2_i, VO_i, VG_i \leq VelMax \quad \forall i \in N \quad (3.45)$$

$$0 \geq V4_i \leq VelMax \quad \forall i = 1, \dots, n - 1 \quad (3.46)$$

$$V3_i = 0 \quad \forall i \in N \quad (3.47)$$

$$-AcelMax \geq A1_i, A3_i, AO_i, AG_i \leq AcelMax \quad \forall i \in N \quad (3.48)$$

$$-AcelMaxTran \geq A2_i \leq AcelMaxTran \quad \forall i \in N \quad (3.49)$$

$$-AcelMax \geq V4_i \leq AcelMax \quad \forall i = 1, \dots, n - 1 \quad (3.50)$$

$$T2_i = 25 \quad \forall i \in N \quad (3.51)$$

$$bin1_i \in \{0,1\} \quad \forall i \in N \quad (3.52)$$

A equação (3.28) representa a medida de desempenho a ser otimizada, ou seja, é a função objetivo do modelo de otimização proposto, que neste caso, objetiva minimizar a tempo total de viagem do percurso.

O número de estações na linha é descrito pelo parâmetro N . As variáveis $T1_i$, $T2_i$, $T3_i$, TO_i , TG_i e $T4_i$, representam o intervalo de tempo de diferentes seções da linha.

O conjunto de variáveis $T1_i$, $X1_i$, $V1_i$ e $A1_i$ representam respectivamente, o tempo de viagem, a posição final, a velocidade no final da seção e a aceleração durante o trajeto para uma rota i do tipo 1, onde uma locomotiva parte do repouso de uma estação e se desloca em movimento acelerado constante. Por conta de não ter velocidade inicial, suas restrições são simplificadas, em relação às demais.

O conjunto $T2_i$, $X2_i$, $V2_i$ e $A2_i$ representa uma rota de tipo 3. Esta seção é peculiar, por que nela ocorre o trânsito de passageiros entre trens, característico do modelo Sem-paradas, por isso na restrição (3.49), $A2_i$ recebe um limite de aceleração específico, de menor magnitude, para garantir o conforto dos usuários. Além disso, para garantir que os passageiros tenham tempo de cumprir sua transição, a variável $T2_i$ é definida como parâmetro de 35s.

O conjunto $T3_i$, $X3_i$, $V3_i$ e $A3_i$ representa uma rota de tipo 2, que representa a situação onde a locomotiva deve parar na próxima estação. Portanto na restrição (3.47), a velocidade $V3_i$ foi definida como 0.

O conjunto $T4_i$, $X4_i$, $V4_i$ e $A4_i$ representa uma rota tipo 2, e retrata uma locomotiva que atravessa uma estação a fim de se conectar a próxima locomotiva, já em movimento.

Como este deslocamento ocorre entre estações, o tamanho de seu vetor é menor em uma unidade, em relação às demais seções.

A seção 4 tanto quanto a seção 3 são diferentes das demais, pois têm velocidade inicial dinâmica. Isso acontece por que a seção que antecede essas duas não é pré-definida e pode ser possível que a seção O ou G seja a antecessora.

Para contornar este problema, foi definida a variável auxiliar binária $bin1$, da seguinte forma:

$$bin1_i = \begin{cases} 0, & \text{se } XG_i > 0 \\ 1, & \text{caso contrário} \end{cases}$$

Assim com a restrição (3.41), no caso da variável XG_i ser maior que zero, ou seja, a seção G existe e, portanto antecede as seções 4 e 3, a variável binária deve assumir o valor de 0, assim, obedecendo a restrição, e garantindo que as restrições (3.33), (3.34), (3.37) e (3.40) utilizem VG como a velocidade inicial. Já no caso contrário, $bin1$ deve assumir o valor de 1, o que força as demais restrições a utilizarem VO como velocidade inicial. O valor de 100.000 considerado na restrição (3.41) é arbitrário (positivo grande), mas deve sempre ser maior que o limite máximo de XG_i .

A restrição (3.42) garante que a locomotiva saia da estação e pare na próxima estação. Vale notar que $Distância_i$ representa a distância entre estações, e não sua posição em relação a Estação 1, como mostrado na Tabela 1.

As restrições (3.43) e (3.44) garantem que as locomotivas se encontrem com mesma velocidade, no mesmo ponto, para assegurar que as locomotivas são capazes de se conectarem.

As demais restrições têm como objetivo garantir a factibilidade dos resultados obtidos.

3.2 MÉTODO DE SOLUÇÃO PARA O MODELO EXATO

Para resolver o modelo anterior definido em (3.28)-(3.52), considerou-se a linguagem AMPL (*A Mathematical Programming Language*) para representação do mesmo. Trata-se de uma linguagem bastante utilizada para resolução de problemas de programação que possuem alta complexidade computacional (FOURER, GAY, KERNIGHAN, 1993), como descrito na seção 2.2.

E com o modelo representado em AMPL, e considerando-se os dados apresentados na Tabela 1, utilizou-se o *solver Lancelot* para obter a solução do mesmo.

Os parâmetros de velocidade, aceleração e distância entre estações foram definidos no arquivo.dat, como mostra a Figura 23.

Neste caso, *AcelMax* presente nas restrições (3.50) e (3.52) é representado pelos parâmetros *lima* e *lima2*. *AcelMaxTran* presente na restrição (3.51) é representado por *lima3* e *lima4*. *VelMax* presente nas restrições (3.47) e (3.48) é representado por *limv*. A coluna *dist* equivale ao vetor *Distancia* presente na restrição (3.42) e da mesma forma os vetores *V3* e *T2* do modelo matemático, equivalem às colunas *V3* e *T2* no arquivo de dados.

Figura 23 – Parâmetros

```

param lima :=-1;
param lima2:= 1;
param lima3:=-0.2;
param lima4:= 0.2;
param limv:= 20;

set J := 1 2 3 4 5 6 7 8 9 10 11 12 13;
param: dist V3 T2 :=
1 643.736 0 35
2 1609.34 0 35
3 1448.406 0 35
4 965.604 0 35
5 1448.406 0 35
6 1448.406 0 35
7 2414.01 0 35
8 1770.274 0 35
9 3218.68 0 35
10 2092.142 0 35
11 1770.274 0 35
12 3218.68 0 35
13 4184.284 0 35 ;
set K := 1 2 3 4 5 6 7 8 9 10 11 12;

```

Fonte: Autor (2019)

Na Tabela 2 apresentam-se parte dos resultados obtidos com a execução do solver. Os dados completos são apresentados na Tabela 11 no Apêndice A. Nestas tabelas, Trajeto representa o espaço entre duas estações. A função objetivo teve resultado igual a 1536,94 segundos. Este valor não tem aplicação direta no tempo de viagem de nenhum dos percursos dos passageiros, porém serve como indicador do tempo de viagem global, pois leva em conta o tempo de viagem de todas as seções.

Pode-se observar que o solver maximizou as acelerações nas seções 1 e 3. Indicando uma correlação direta entre o tempo de viagem, e as acelerações nessas seções.

As demais variáveis foram determinadas de maneira independente, podendo assumir valores de baixa ou alta magnitude, e em alguns casos, valores positivos ou negativos, de forma a se adaptar à situação.

Tabela 2 – Solução Modelo Exato

Trajeto	Seção						Variável
	1	2	O	G	3	4	
0	1	0,2	-1	-1	-1	0,7117	Aceleração (m/s^2)
	9,62634	16,6217	15,6968	5,3678	0	13,015	Velocidade (m/s)
	9,61948	35	0,92	10,3208	5,36244	10,7464	Tempo (s)
	46,2675	459,422	14,8706	108,744	14,4308	98,831	Distancia (m)
1	1	0,2	0	-0,43536	-1	0,35623	Aceleração (m/s^2)
	13	20	20	11,0917	0	16,0886	Velocidade (m/s)
	12,9925	35	28,3754	20,4468	11,0872	14,0259	Tempo (s)
	84,4024	577,823	567,609	317,931	61,574	190,689	Distancia (m)
2	1	0,11215	0	-1	-1	0,696621	Aceleração (m/s^2)
	16,0834	20	20	4,98416	0	11,7691	Velocidade (m/s)
	16,0696	35	24,3756	15	5	9,71143	Tempo (s)
	129,116	631,613	487,587	187,542	12,5477	81,5021	Distancia (m)
3	1	0,2	0,147598	-1	-1	0,72234	Aceleração (m/s^2)
	11,7634	18,7655	20	5,48783	0	13,6174	Velocidade (m/s)
	11,7436	35	8,3784	14,5042	5,49051	11,2457	Tempo (s)
	68,9555	534,218	162,406	184,899	15,1258	107,531	Distancia (m)

Fonte: Autor (2019)

Para garantir a factibilidade dos resultados obtidos foi desenvolvido um algoritmo na linguagem C++ capaz de verificar se todas as restrições foram obedecidas. Parte do algoritmo pode ser visto na Figura 24. Restrições de aceleração máxima ou mínima foram testadas em equações simples, enquanto que nas restrições que exigem mais de uma variável, foi adicionada uma tolerância de erro, para lidar com o fato das variáveis envolverem números fracionários.

Os resultados obtidos pelo algoritmo mostram que as margens de erro para as restrições de velocidade eram menores que $0,01 \frac{m}{s}$, indicando ótima qualidade nestes resultados. Porém, para as restrições relacionadas à posição, a margem de erro chegou 0,8 metros. Indicando que, caso os resultados da Tabela 2, fossem de fato utilizados numa linha real, seria necessário, a presença de um controlador, ou de um sistema digital inteligente, para consertar a posição da locomotiva durante o trajeto.

Figura 24 - Algoritmo de Teste

```

if(A1[i] < 0 || A1[i] > amax)
{
    cout << "erro A1 - " << A1[i] << endl;
}
if(A2[i] < -amax2 || A2[i] > amax2)
{
    cout << "erro A2 - " << A2[i] << endl;
}
if(A3[i] < -amax || A3[i] > 0)
{
    cout << "erro A3 - " << A3[i] << endl;
}
if(AO[i] < -amax || AO[i] > amax)
{
    cout << "erro AO - " << AO[i] << endl;
}
if(AG[i] < -amax || AG[i] > amax)
{
    cout << "erro AG - " << AG[i] << endl;
}
if(XO[i] + XG[i] + X2[i] + X3[i] + X1[i] - (est[i+1] - est[i]) > tolerancia || XO[i] + XG[i] + X2[i] + X3[i] + X1[i] - (est[i+1] - est[i]) < -tolerancia )
{
    cout << "-erro- Distancia da parada: " << i << " = " << XO[i] + XG[i] + X2[i] + X3[i] + X1[i] - (est[i+1] - est[i]) << endl;
}
if(V1[i] - (A1[i]*T1[i]) > toleranciav)
{
    cout << "erro VA1 - " << V1[i] - (A1[i]*T1[i]) << endl;
}

```

Fonte: Autor (2019)

Também foi calculado que neste percurso de 26232,24 metros como indicado na Tabela 1, uma locomotiva que parte da primeira estação, com destino à última estação, chega na última estação com um erro acumulado de 13 metros. Ou seja, um erro acumulado percentual de 0,05%.

Na Seção 3.5.1 apresenta-se a análise comparativa de tempos de viagem, entre o modo Sem-paradas, e o modo real utilizado atualmente no modal metroviário.

3.2.1 Comparação com a situação real

Nesta seção apresenta-se um novo algoritmo que foi desenvolvido para calcular os tempos de viagem para cada estação, com base nos resultados obtidos na solução do modelo exato dado pelas equações (3.28)-(3.56),

A Tabela 3 apresenta os resultados obtidos pelo algoritmo, onde o Modo em movimento indica que o passageiro trocou de locomotivas durante o percurso, evitando assim, a necessidade de parar em qualquer uma das estações.

Tabela 3 - Tempo de Viagem Modelo Real

Destino	Tempo de percurso (minutos)		
	Modo Real	Modo em movimento	Modo sentado
Dupont Circle	2	1	1
Woodley Park-Zoo	4	2,7	3,24
Cleveland Park	6	4	4,6
Van Ness-UDC	8	5,2	6,27
Tenleytown-AU	10	6,7	7,78
Friendship Heights	12	8,1	9,8
Bethesda	15	10,3	12
Medical Center	18	11,9	14,26
Grosvenor Metro	21	14,65	17
White Flint	24	16,45	19,5
Twinbrook	27	18	21,1
Rockville	31	20,8	24,6
Shady Grove	35	24,85	28,63

Fonte: Autor (2019)

Já o Modo sentado indica que o passageiro não realiza troca de locomotivas durante o trajeto. Nota-se que mesmo sem realizar trocas de locomotiva, este passageiro ainda evita parar em metade das estações durante o percurso, pois cada locomotiva para apenas em estações alternadas.

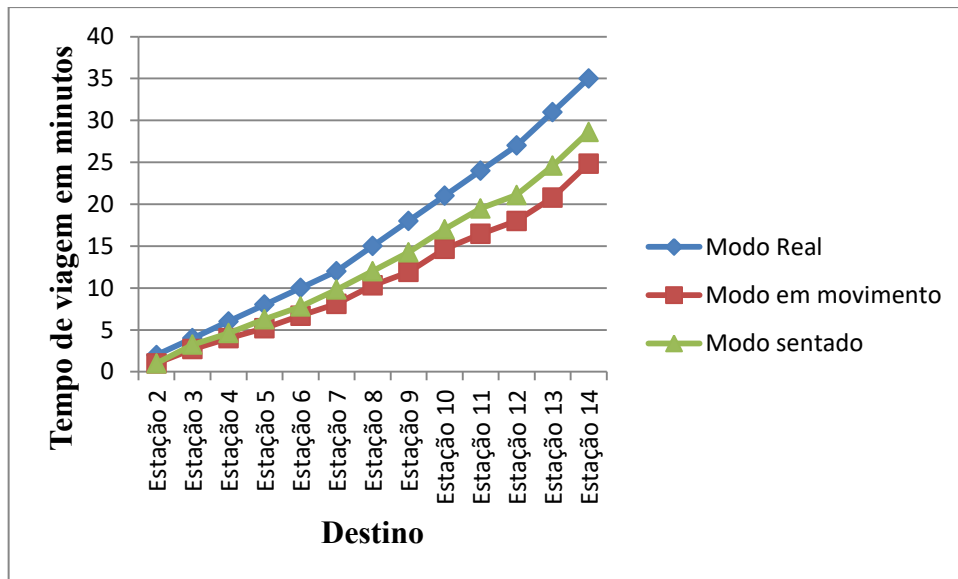
Com base nos resultados pode-se observar que o modo Sem-paradas, foi capaz de reduzir consideravelmente o tempo de viagem para longas distâncias.

No caso de um passageiro que entra na primeira estação, e deseja descer na 13ª estação, seu tempo de viagem pode ser reduzido em 29%, caso esteja disposto a trocar de locomotiva durante a viagem, ou em 18,2% caso contrário. Também é possível que o passageiro não troque de locomotiva em todas as oportunidades, mas apenas em algumas delas, neste caso seu tempo de viagem estaria entre 24,85 minutos e 28,63 minutos, dependendo de quantas trocas forem realizadas.

Na Figura 25 apresentam-se os tempos de viagem nos diferentes modos.

Vale notar que os trajetos finais têm maior comprimento que os iniciais, como mostrado na Tabela 1.

Figura 25 - Gráfico Comparação de Tempo de Viagem Modelo Exato



Fonte: Autor (2019)

3.3 Algoritmo Genético

O algoritmo foi programado na linguagem C++, uma de suas características distintas são as classes, uma ferramenta bastante útil, que foi usada extensivamente durante o estudo.

Utilizando uma classe, o programador pode criar e manipular objetos. Por exemplo, como ilustrado na Figura 26, durante a programação do algoritmo, foi definida a classe *Populacao* que representa a população de indivíduos, parte primordial de um algoritmo genético. Como se pode observar, esta classe tem seis atributos, linha **Individuos* e **PosicaoEstacoes* são vetores de objetos, o primeiro guarda objetos da classe linha fixada anteriormente, enquanto o segundo guarda números inteiros.

TamanhoPopulacao e *QtdeEstacoes* são números inteiros e guardam o tamanho dos vetores citados anteriormente, este tipo de composição é comum pois possibilita que vetores tenham tamanho dinâmico.

Por último *Gene* e *Melhor* guardam a informação genética e nível de adaptação do indivíduo mais adaptado.

Figura 26 - Classe População C++

```

class Populacao
{
private:
    linha* Individuos;
    long int TamanhoPopulacao;
    long int *PosicaoEstacoes;
    long int QtdeEstacoes;
    vector<string> Gene;
    long double Melhor;

public:
    Populacao(long int tampop, long int *estcoes, long int tamestc)
    {

    }

    ~Populacao()
    {

    }

    string crossover(string x, string y, long int siz)
    {

    }

    string mutacao(string j, long int siz)
    {

    }

    void ordenar()
    {

    }

    evolucao()
    {

    }

};

```

Fonte: Autor (2019)

Já abaixo de *public*, estão contidas as funções responsáveis por manipular esses dados. Como esperado, foi necessário definir uma função *crossover* e uma função *mutação*, indispensáveis em um algoritmo genético.

Outra função importante é a *ordenar*, necessária para ranquear os indivíduos de acordo com sua adaptabilidade. As funções que têm o mesmo nome da classe, *Populacao* e *~Populacao*, são características de qualquer classe em C++, e são responsáveis por criar ou destruir um objeto genérico da classe que está sendo definida.

Por último a função *evolução* é encarregada de reunir todas as outras operações e aplicar a meta heurística.

Com isto fica definido que os indivíduos da população são da classe *linha*. A classe *linha* por sua vez é formada por seis tipos de seções onde cada tipo de seção possui os atributos de aceleração, velocidade, tempo e distância, como apresentado na seção 3.4 deste estudo.

Cada tipo de seção está definido em vetores, onde cada posição do vetor representa um trajeto, e o tamanho dos vetores depende do número de estações a serem atendidas pela linha.

Na equação (3.53) tem-se um exemplo de um vetor de aceleração que faz parte da seção 1, do trajeto 0. Onde a primeira letra indica o atributo, o número posterior indica o tipo de seção, e o número entre colchetes indica o trajeto, no lado direito encontra-se o valor deste atributo.

$$A1[0] = 0,5322 \frac{m}{s^2} \quad (3.53)$$

As variáveis *tempo* e *velocidade* foram definidas como variáveis de decisão, o que significa que, os valores destas variáveis definem o código genético de cada indivíduo. Enquanto que, os valores das demais variáveis são encontrados a partir das restrições de movimento do sistema.

Portanto, a lógica aplicada para solucionar o problema, pode ser resumida da seguinte forma:

1. Definir como variáveis de decisão as velocidades finais, e o tempo de duração de cada seção. E então, gerar uma população inicial de 100 indivíduos onde cada seção desses indivíduos tem sua velocidade final e duração definida aleatoriamente;
2. Com isto, usar as fórmulas definidas na seção 3.1, 3.2 e 3.3 deste estudo, para encontrar as variáveis de cada seção;
3. Ranquear os indivíduos de acordo com seu nível de adaptabilidade, calculado a partir da equação (3.28);
4. Codificar as variáveis de decisão, transformando os valores que estão a princípio na base decimal para valores na base binária. Este processo pode ser observado na Figura 41 do Apêndice F. A função $x = \text{bitset}<6>(y.\text{to_string}())$ transforma um valor y da base decimal, para um valor na base binária de seis casas e atribui o valor obtido na *string* x . Enquanto que a função $J.\text{push_back}(x[j])$ passa o valor da *string* x para a *string* J , que contém, sequência genética inteira do indivíduo. O valor contido na *string* x é apagado na função $x.\text{clear}()$, e o processo se repete para todas as variáveis de decisão.
5. Selecionar os 50 indivíduos mais adaptados;
6. Criar uma nova população a partir do crossover dos indivíduos selecionados. Onde 80 indivíduos são criados a partir do crossover entre os 10 indivíduos mais adaptados, e os indivíduos com ranque de 10 a 50. 10 Indivíduos são gerados a partir do crossover entre os indivíduos com ranque 10 a 20 e os indivíduos com ranque 20 a 30. E por

último, os 10 indivíduos mais adaptados também são adicionados a esta nova população, sem sofrerem crossover.

7. O crossover é efetuado de 11 maneiras diferentes para garantir maior diversificação genética, no momento do crossover uma dessas maneiras é escolhida aleatoriamente. As 11 formas de crossover escolhidas podem ser visualizadas nas figuras 38 e 39 no Apêndice C.
8. A nova população passa pela fase de mutação, onde cada indivíduo tem probabilidade de 5% sofrer mutação. Caso escolhido, cada um de seus genes terá probabilidade de 5% de ser mutado. No caso, um gene de valor “1”, passaria a ter valor “0” após a mutação. Essa função pode ser visualizada na Figura 42 do Apêndice D.
9. Os passos 3 a 8 são repetidos por 1.000 iterações.

Como mostrado na Figura 41 do Apêndice F, no processo de codificação, foi adotada a base binária com seis casas, o que permite 64 resultados diferentes. Por causa disso, para simplificar a aplicação do algoritmo genético, foi adotado um valor de velocidade máxima de 32 m/s, valor que é múltiplo de 64.

3.3.1 Comparação com a situação real

O resultado parcial obtido com aplicação do algoritmo genético pode ser visualizado na Tabela 4 em que, *Trajeto* se refere ao espaço entre estações. Os dados completos são apresentados na Tabela 13 do Apêndice E.

Tabela 4 – Resultado Parcial Modelo Genético

Trajeto	Seção						Variável
	1	2	O	G	3	4	
0	0.532	-0.442	0.055	0	-0.092	0.428	Aceleração (m/s^2)
	16.5	1	2.5	0	0	14.5	Velocidade (m/s)
	31	35	27	0	27	28	Tempo(s)
	255.75	306.25	47.25	0	33.75	238	Distância (m)
1	0.514	0.257	-0.422	0	-0.091	0.444	Aceleração (m/s^2)
	14.5	23.5	4.5	0	0	16.5	Velocidade (m/s)
	28.1724	35	45	0	492.222	27	Tempo(s)
	204.25	665	630	0	110.75	283.5	Distância(m)
2	0.787	-0.214	0.607	0	-0.997	-0.322	Aceleração (m/s^2)
	16.5	9	26	0	0	16	Velocidade (m/s)
	20.939	35	28	0	26.076	31	Tempo(s)
	172.75	446.25	490	0	339	651	Distância(m)

O algoritmo genético não maximizou as acelerações das seções 1 e 2, e não fez uso da seção G. Contudo o algoritmo genético obteve uma menor taxa de erro em relação às restrições.

Para as restrições relacionadas à posição, a margem de erro chegou 0,45 metros, 44% menor do que o modelo exato. Enquanto que o erro acumulado até a estação final foi de 5 metros, 62% menor que o erro do modelo exato.

A Tabela 5 apresenta o tempo de percurso a partir da primeira estação.

Primeiramente, pode-se observar que para os destinos iniciais, o modelo genético obteve resultados até piores que o modelo real, mas ao compararem-se os resultados finais da [Tabela 3 e Tabela 5](#), observa-se que ambos os modelos obtiveram resultados próximos.

Porém, vale ressaltar que o limite de velocidade permitido com o algoritmo genético é 60% maior que o limite obtido com o método exato, o que indica que o algoritmo genético não foi tão eficiente quanto o método exato.

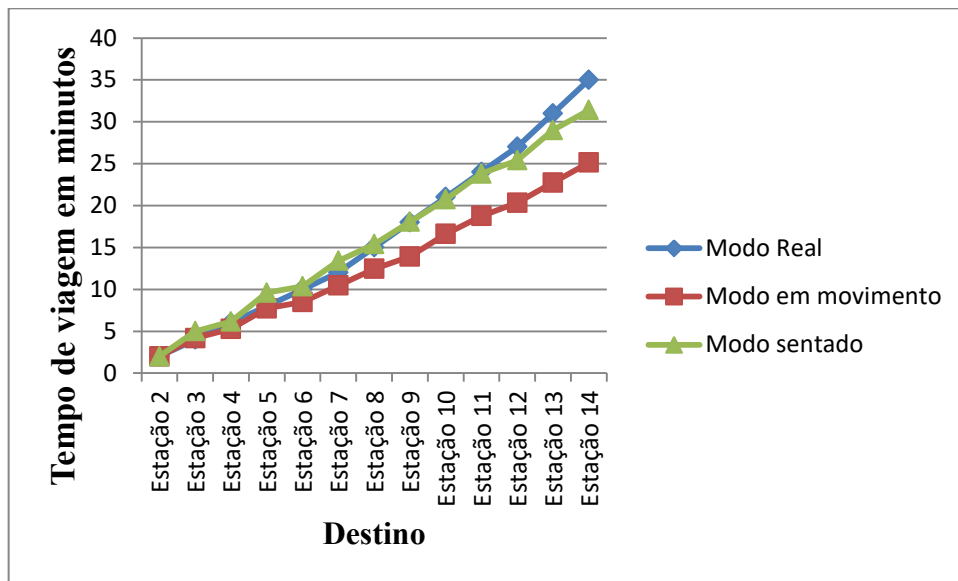
Tabela 5 - Tempo de Viagem Modelo Genético

Destino	Tempo de percurso (minutos)		
	Modo Real	Modo em Movimento	Modo Sentado
Estação 2	2	2	2
Estação 3	4	4,17	5
Estação 4	6	5,28	6,15
Estação 5	8	7,73	9,59
Estação 6	10	8,49	10,35
Estação 7	12	10,45	13,4
Estação 8	15	12,44	15,38
Estação 9	18	13,91	18,06
Estação 10	21	16,6	20,74
Estação 11	24	18,74	23,81
Estação 12	27	20,339	25,4
Estação 13	31	22,73	29
Estação 14	35	25,13	31,4

Fonte: Autor (2019)

Na Figura 27 é possível identificar um comportamento incomum na progressão dos tempos de viagem dos primeiros destinos, que se ajusta a partir da Estação 6 e passa a se comportar de maneira similar ao modelo exato. Isto pode ter acontecido, pois, como se observa na Tabela 1, as estações iniciais estão mais próximas umas das outras, o que indicaria que o modelo genético tem dificuldade em lidar com este tipo de situação.

Figura 27 - Gráfico para Comparação de Tempo de Viagem Modelo Genético



Fonte: Autor (2019)

Portanto, apesar de ser um pouco mais preciso que o modelo exato, o algoritmo genético apresentou resultados de qualidade razoavelmente inferior. Com isso, para os próximos capítulos que têm como objetivo, avaliar a capacidade de transporte de passageiros do modo Sem-paradas, foi escolhido o método exato como base para os cálculos.

4 CAPACIDADE DE TRANSPORTE

Neste capítulo foi calcula-se a capacidade máxima de transporte do modelo Sem-paradas para o trecho da linha *Red line*. E também se analisa a frequência de atendimento necessária para atender a demanda desta linha, com o auxílio da ferramenta Arena. Na próxima seção é apresentada a simulação que determina a frequência de atendimento necessária.

4.1 SIMULAÇÃO DE ATENDIMENTO DE PASSAGEIROS NA LINHA

Primeiramente, é necessário determinar o fluxo de passageiros na linha, para definir a frequência de atendimento necessária. Os seguintes dados foram disponibilizados pela WMATA (2018) na Tabela 6:

Tabela 6 - Fluxo de Passageiros Diários

Estação	Número de Passageiros
Shady Grove	11139
Rockville	3756
Twinbrook	3807
White Flint	3506
Grosvenor	4995
Medical Center	5181
Bethesda	8999
Friendship Heights	7900
Tenleytown-AU	5282
Van Ness-UDC	5476
Cleveland Park	3657
Woodley Park-Zoo	5913
Dupont Circle	16542
Farragut North	22184

Fonte: WMATA (2018)

A coluna à direita contém a média do número de passageiros que entra na estação diariamente em dias de semana.

Todas as estações da Figura 8 atendem apenas a *Red Line*, porém não há dados sobre o destino de cada passageiro que acessa a estação. Além disso, é importante ressaltar que as

estações escolhidas representam apenas um trecho da linha inteira, que começa em *Shady Grove* e continua após *Farragut North*, porém, como as estações posteriores atendem a mais de uma linha, elas foram retiradas do escopo deste estudo.

Com isso as premissas foram admitidas:

- Por ser uma estação de topo, ou seja, de final ou começo de linha, pode-se afirmar que 100% dos passageiros que entram na estação *Shady Grove* viajam no sentido *Shady Grove – Farragut North*.
- Já os passageiros que acessam a estação *Rockville* têm como opção apenas um destino no sentido *Farragut North – Shady Grove*, e mais de 12 destinos no sentido *Shady Grove – Farragut North*. Com isso foi considerado que 96% dos indivíduos que entram nesta estação, utilizam a *Red Line* no sentido *Shady Grove – Farragut North*, enquanto que 4% a utilizam no sentido contrário.

A mesma lógica foi aplicada para as demais estações, o que resultou nos dados apresentados na segunda coluna da Tabela 7. Utilizando-se estes dados, foi feita a terceira coluna da Tabela 7.

Tabela 7 - Porcentagem de Passageiros *Shady Grove - Farragut North*

Estação	Porcentagem de Passageiros	Fluxo de Passageiros Real	Fluxo de Passageiros por Hora no Horário de Pico
Shady Grove	100%	11136	919
Rockville	96%	3605	297
Twinbrook	92%	3502	289
White Flint	88%	3085	255
Grosvenor	84%	4195	346
Medical Center	80%	4144	342
Bethesda	76%	6839	564
Friendship Heights	72%	5688	469
Tenleytown-AU	68%	3591	296
Van Ness-UDC	64%	3504	289
Cleveland Park	60%	2194	181
Woodley Park-Zoo	56%	3311	273
Dupont Circle	52%	8601	710
Farragut North	48%	10648	878

Fonte: Autor (2019)

Como citado na Seção 2.1.2, na linha de metrô de São Paulo, 33% das viagens diárias estão concentradas nos horários de pico matutino e vespertino, que duram 2 horas cada. Assumindo que a *Red Line* tenha comportamento parecido, pode-se afirmar que 16,5% dos

11139 passageiros que acessam a Estação *Shady Grove* diariamente, estão concentrados em um horário de pico.

Assim, a quarta coluna da Tabela 7 foi desenvolvida com o fluxo de passageiros/hora no horário de pico.

Os valores da quarta coluna da Tabela 7 foram aplicados nos módulos *Create* do Arena para representar a chegada de passageiros nas estações.

A chegada da locomotiva na estação foi representada pelos módulos *Hold* e *Signal*, e para representar o movimento entre locomotivas dos passageiros algumas considerações foram feitas.

Primeiramente admitiu-se que em cada estação 10% dos passageiros de uma locomotiva desembarcam, e que 10% dos passageiros estariam dispostos a viajar de Modo em Movimento. Com isso, a lotação da locomotiva não deve mudar significativamente durante as trocas de passageiros.

Além disso, admitiu-se que a lotação da locomotiva não deve ultrapassar 80%, para garantir que exista espaço suficiente para que os passageiros possam se movimentar.

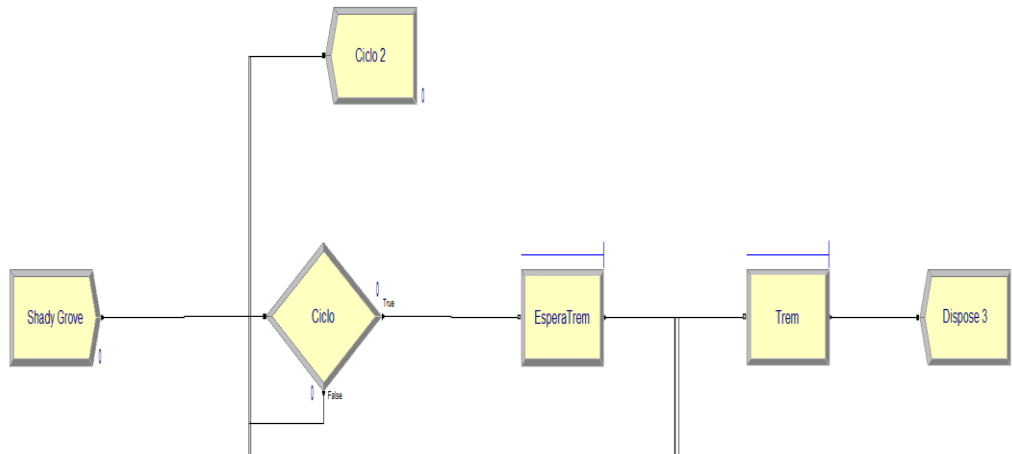
Estas considerações garantem que mesmo em uma situação crítica, onde a locomotiva esteja em sua capacidade máxima, apenas 37 passageiros tentariam atravessar a porta na seção 2 de movimento da locomotiva. Portanto, substituindo estes valores na Equação (2.10):

$$35 = \frac{37 * T_{pf}}{1} + 5 \quad (4.1)$$

O que resulta em T_{pf} , ou tempo de fluxo de passageiros, de 0,81 passageiros/segundo, valor que se encontra entre os valores máximo e mínimo definidos por (COCA; TORRES, 2004).

Com estas considerações, e os tempos de viagem obtidos na Seção 3.4, foi feita a simulação no Arena, parte do modelo desenvolvido no Arena pode ser visualizado na Figura 28 e na Figura 29.

Figura 28- Fluxo dos Passageiros

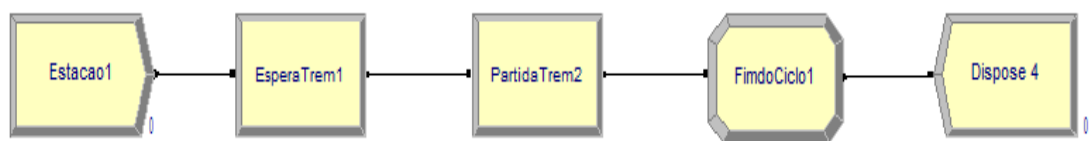


Fonte: Autor (2019)

Na Figura 28 observa-se que a representação do movimento dos passageiros no sistema. O módulo *Shady Grove* cria entidades com base na Tabela 7, com uma frequência baseada em distribuição normal. O módulo *ciclo* checa se a locomotiva já partiu ou não, caso tenha partido, a entidade é excluída do sistema no módulo *Ciclo 2*. Caso não tenha partido, a entidade parte para o módulo *EsperaTrem*, onde aguarda pela chegada da locomotiva.

Quando o módulo *EsperaTrem* recebe o sinal de que a locomotiva chegou, os passageiros são liberados para o módulo *trem*, onde ficam até receberem o sinal de desembarque.

Figura 29 - Fluxo dos Sinais



Fonte: Autor (2019)

Na Figura 29, os sinais que controlam o sistema da Figura 28 são apresentados. O módulo *Estacao1* cria uma entidade que espera no módulo *EsperaTrem1* por um período equivalente a um headway. Depois de esperar, a entidade ativa o sinal de que a locomotiva deve partir da estação no módulo *PartidaTrem2*. E então, no módulo *FimdoCiclo1* a entidade informa ao sistema que o primeiro ciclo desta locomotiva já acabou, por último, a entidade é eliminada do sistema no módulo *Dispose 4*.

Com base em dados disponibilizados pela plataforma online *Google Maps*, encontrou-se que o *headway* empregado na *Red Line* em horários de pico é de 4 minutos. Com isso 2 testes foram realizados, no primeiro o *headway* era igual a 4 minutos, no segundo, igual a 3 minutos.

Os resultados do teste realizado com *headway* de 4 minutos podem ser observados na Figura 30. Nesta figura observa-se que em média, a lotação da locomotiva era de 98,46 passageiros, e que a locomotiva teve lotação máxima de 171 passageiros.

A lotação máxima de 171 passageiros é excessiva, pois caso haja outra locomotiva com 171 passageiros, no momento em que 10% destes passageiros precisarem mudar de locomotiva para desembarcar, eles não serão capazes de realizar a troca de trem, pois para isso, seria necessário que a outra locomotiva fosse capaz de suportar 188 passageiros. Além disso, admitiu-se anteriormente que a lotação máxima para viabilizar a movimentação de usuários seria 80%.

Figura 30 – *Headway* de 4 minutos

Number Waiting	Average	Half Width	Minimum Value	Maximum Value
Patio 1.Queue	3.4775	(Insufficient)	0.00	59.0000
Patio 2.Queue	1.2080	(Insufficient)	0.00	20.0000
Patio 3.Queue	1.4355	(Insufficient)	0.00	25.0000
Patio 4.Queue	2.2400	(Insufficient)	0.00	40.0000
Patio 5.Queue	1.3213	(Insufficient)	0.00	21.0000
Patio 6.Queue	0.8186	(Insufficient)	0.00	13.0000
Patio 7.Queue	2.9432	(Insufficient)	0.00	52.0000
Trem.Queue	98.4640	(Insufficient)	0.00	171.00

Fonte: Autor (2019)

Um segundo teste foi realizado com *headway* de três minutos, e os resultados obtidos podem ser visto na Figura 31.

Figura 31 - *Headway* de 3 minutos

Number Waiting	Average	Half Width	Minimum Value	Maximum Value
Patio 1.Queue	2.3083	(Insufficient)	0.00	45.0000
Patio 2.Queue	0.9063	(Insufficient)	0.00	17.0000
Patio 3.Queue	0.9774	(Insufficient)	0.00	19.0000
Patio 4.Queue	1.3402	(Insufficient)	0.00	28.0000
Patio 5.Queue	0.7840	(Insufficient)	0.00	14.0000
Patio 6.Queue	0.4937	(Insufficient)	0.00	9.0000
Patio 7.Queue	1.9535	(Insufficient)	0.00	37.0000
Trem.Queue	69.8947	(Insufficient)	0.00	126.00

Fonte: Autor (2019)

Neste teste a lotação máxima foi de 126 passageiros, o que representa 68,74% da capacidade máxima da locomotiva. Com isso pode-se concluir que para atender este trecho da linha com comboios de apenas dois carros, será necessário utilizar um *headway* menor que o *headway* aplicado atualmente na *Red Line*.

Com esta informação, resta saber qual seria o *headway* mínimo aplicável para esta linha, no modo Sem-paradas. Por isso, a próxima seção tem como foco, calcular este *headway*.

Tabela 8 - Tempo de Viagem *Shady Grove - Farragut North*

Destino	Tempo de percurso (minutos)		
	Modo Real	Modo em movimento	Modo sentado
Estação 2	4	4,29	4,29
Estação 3	8	6,97	7,72
Estação 4	11	8,55	9,3
Estação 5	14	10,47	11,87
Estação 6	17	13,16	14,56
Estação 7	20	14,75	16,87
Estação 8	23	16,86	18,98
Estação 9	25	18,15	20,98
Estação 10	27	19,65	22,48
Estação 11	29	20,77	24,14
Estação 12	31	22,29	25,66
Estação 13	33	23,9	27,87
Estação 14	35	24,79	28,75

Fonte: Autor (2019)

Contudo, vale notar que na seção anterior o tempo de viagem foi minimizado para o sentido *Farragut North – Shady Grove*, por isso, nesta seção uma nova minimização de tempo

de viagem foi feita, para o sentido *Shady Grove – Farragut North*, com o objetivo de testar os efeitos desta mudança, e garantir que a linha opere em eficiência máxima. A Tabela 8 apresenta os resultados obtidos por essa mudança.

O tempo de viagem total obtido foi quase idêntico ao obtido no teste anterior, porém um dado ficou mais claro nesse teste. No trajeto *Shady Grove – Rockville*, que é o trajeto mais extenso da linha, o tempo de viagem real é menor que o tempo de viagem pelo modo Sem-Paradas, o que sugere que as locomotivas reais, operam com velocidades superiores a $20 \frac{m}{s}$ quando possível.

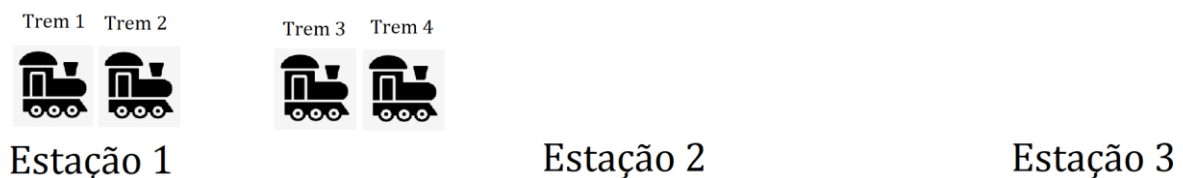
A capacidade de cada locomotiva já foi apresentada na Seção 2.6, portanto, foi considerado que cada carro de um comboio de capacidade para 184 passageiros.

4.2 CAPACIDADE DE ATENDIMENTO MÁXIMA

Pode-se calcular o *headway* mínimo com base nos resultados obtidos na Seção 3.4 da seguinte forma:

Considere a Figura 32, nesta figura o comboio formado pelos trens 1 e 2 espera na Estação 1, enquanto que o comboio dianteiro de move em direção a Estação 2. Como mostrado anteriormente, o Trem 2 deve se conectar com o Trem 3 no final da seção [i+1], portanto o percurso realizado pelo Trem 2 passará pelas seções: 1[i], 2[i], O[i], G[i] e 4[i]. Enquanto que o percurso do Trem 3 passará pelas seções: 1[i], 2[i], O[i], G[i], 3[i] e 1[i+1]. Além disto, o Trem 3 deve parar na Estação 2 por um certo período de tempo, para que os passageiros realizem o embarque e desembarque.

Figura 32 - *Headway* Mínimo



Fonte: Autor (2019)

Portanto, considerando um tempo de embarque/desembarque na Estação 2 de 30s, pode-se afirmar que a diferença entre os tempos de viagem dos dois trens que devem se encontrar é dado pela Equação (4.2).

$$\begin{aligned}
 & \text{Diferença entre tempos de percurso} \\
 & = (T1[i] + T2[i] + TO[i] + TG[i] + T3[i] + 30 + T1[i + 1]) \\
 & - (T1[i] + T2[i] + TO[i] + TG[i] + T4[i])
 \end{aligned} \tag{4.2}$$

Que pode ser simplificado para:

$$\text{Diferença entre tempos de percurso} = T3[i] + T1[i + 1] + 30 - T4[i] \tag{4.3}$$

Esta diferença representa o tempo mínimo que o Trem 2 deve esperar na Estação 1 antes de poder partir.

Contudo, nota-se que cada trajeto da linha apresenta tempos de percurso diferentes, enquanto que a frequência de locomotivas deve ser padronizada para a linha inteira. Por isso defini-se que, o *headway* mínimo deve ser calculado a partir do trajeto crítico, ou seja, o trajeto que tem o maior tempo mínimo de espera da linha.

Utilizando-se a equação (4.3) em cada um dos trajetos da *Red Line* sentido *Shady Grove – Farragut North*, encontra-se que o *headway* mínimo é igual a 50s, encontrado no Trajeto *Shady Grove – Rockville*.

Com isso pode-se afirmar que a linha tem capacidade máxima de:

$$\text{Capacidade máxima da linha} = (184 * 2) * \frac{3600}{50} \tag{4.4}$$

Portanto,

$$\text{Capacidade máxima da linha} = 26496 \text{ passageiros por hora} \tag{4.5}$$

Entretanto, foi observado que a otimização realizada nas seções anteriores tinha como objetivo a redução do tempo de viagens, e não a otimização da capacidade da linha, por isso o modelo apresentado na Seção 3.4 foi adaptado para calcular capacidade máxima da linha.

Como mostrado anteriormente, a capacidade máxima da linha varia em função do *headway* mínimo e da capacidade de cada carro, considerando que a capacidade dos carros é fixa, se o *headway* for minimizado encontra-se a capacidade máxima.

Portanto, a função objetivo passou a ser:

$$\text{Minimize Headway} \tag{4.6}$$

E uma restrição nova foi adicionada:

$$\text{Headway} \geq T3_i + T1_{i+1} + 30 - T4_i \quad \forall i = 1, \dots, n-1 \quad (4.7)$$

O modelo adaptado foi solucionado pelo *solver Lancelot* e foi obtido que o *headway* mínimo neste caso seria de 47,39 segundos. Uma redução de tempo de apenas 5%, que resulta em uma melhora na capacidade de apenas 5% também.

$$\text{Capacidade máxima da linha} = 27949 \text{ passageiros por hora} \quad (4.8)$$

Contudo, essa adaptação também afetou os tempos de viagem, como mostra a Tabela 9. Em comparação com o resultado da Seção 3.2, o tempo de viagem aumentou em 14,72% e em 12,73% para o Modo em movimento e o Modo sentado respectivamente.

Com isso, conclui-se que a adaptação para diminuir os *headways* não vale a pena, e que o *headway* mínimo aplicável neste caso deve ser de 50s, que é suficiente para atender a seção da linha estudada.

Tabela 9 - Tempos de Viagem com Headway Otimizado

Destino	Tempo de percurso(minutos)		
	Modo Real	Modo em movimento	Modo sentado
Estação 2	4	4,69	4,69
Estação 3	8	7,57	8,28
Estação 4	11	9,35	10
Estação 5	14	11,38	12,8
Estação 6	17	14,3	15,7
Estação 7	20	16	18,2
Estação 8	23	18,37	20,49
Estação 9	25	20	22,74
Estação 10	27	21,75	24,48
Estação 11	29	23,3	26,58
Estação 12	31	25,17	28,45
Estação 13	33	26,95	30,93
Estação 14	35	28,44	32,41

Fonte: Autor (2019)

O próximo capítulo deste estudo apresenta cenários alternativos de aplicação do método Sem-paradas, no primeiro, todas as informações desenvolvidas neste estudo são compiladas, e uma otimização de tempo de viagens desta linha é feita. Nas demais, são

discutidos problemas que o modo Sem-paradas acarretaria na prática, e possíveis soluções para esses problemas.

5 PROPOSTAS DE CENÁRIOS ALTERNATIVOS

Levando-se em consideração todas as informações desenvolvidas neste trabalho, nas próximas seções serão abordados cenários alternativos de aplicação do modo Sem-paradas.

5.1 CENÁRIO 1 – VELOCIDADE MÁXIMA

Na Seção 5.1, foi observado que a velocidade máxima das locomotivas que atendem a *Red Line*, passa dos 20 m/s estabelecidos anteriormente. Além disso, também se notou que o tempo de embarque/desembarque de 35 segundos pode não ser o suficiente para atender a linha, pois permite que em média apenas 37 pessoas consigam se mover entre carros durante esse período.

Portanto no último teste, a velocidade máxima permitida para a locomotiva foi estabelecida como 25 m/s, enquanto que o tempo de duração da seção 2 de movimento das locomotivas foi estabelecido como 45 segundos.

Os resultados dos tempos de viagem podem ser observados na Tabela 10.

Tabela 10 - Tempos de viagem Final

Destino	Tempo de percurso (minutos)		
	Modo Real	Modo em movimento	Modo sentado
Estação 2	4	3,45	3,45
Estação 3	8	5,71	6,5
Estação 4	11	7,1	7,9
Estação 5	14	8,9	10,26
Estação 6	17	11,28	12,64
Estação 7	20	12,67	14,76
Estação 8	23	14,69	16,78
Estação 9	25	16	18,76
Estação 10	27	17,48	20,23
Estação 11	29	18,6	21,87
Estação 12	31	20	23,35
Estação 13	33	21,59	25,42
Estação 14	35	22,61	26,44

Fonte: Autor (2019)

Primeiramente, percebe-se que neste caso o tempo de viagem para a Estação 2 pelo modo Sem-paradas passou a ser menor do que o tempo real. Levando-se em consideração que este tempo era maior que o tempo real no teste com limite de velocidade igual a 20 m/s, pode-se concluir que a velocidade máxima real se encontra entre 20 e 25 m/s.

Além disso, o tempo de viagem total diminuiu em relação ao teste da Seção 3.4.1, com isso pode-se perceber que o aumento do limite de velocidade máximo influencia mais no tempo de viagem do que o aumento na duração da seção de Movimento 2.

Houve também um aumento no *headway* mínimo, que passou a ter valor de 52.61 segundos.

Com isso conclui-se que esta linha teria as seguintes características:

- Redução de tempo viagem para passageiros no Modo sentado igual a 24,46%.
- Redução de tempo viagem para passageiros no Modo em movimento igual a 35,4%.
- Número de passageiros que conseguem realizar a troca de locomotivas, considerando um fluxo de passageiros de 0,81 segundos/passageiro, igual a 49.
- Capacidade máxima de atendimento da linha igual a 25181 passageiros/hora.

Durante todo o estudo considerou-se que um comboio deve ser formado por dois carros. Entretanto, a composição do comboio pode ser feita de várias formas, que serão exploradas nas próximas seções.

Assim, nas próximas seções serão abordadas problemáticas que envolvem o modo Sem-paradas.

5.2 CENÁRIO 2 – NÚMERO DE CARROS

O número de carros por comboio não precisa ser fixo, como mostra a Figura 33. Contanto que o número de carros que parem em uma estação, seja igual ao número de carros parados na estação, o sistema consegue funcionar normalmente em *loop*.

Figura 33 - Variação de Carros



Fonte: Autor (2019)

5.3 CENÁRIO 3 – NÚMERO DE PARADAS POR ESTAÇÃO

Com base na subseção anterior, é possível definir o número de carros que param por estação de acordo com a demanda da estação, como mostra a Figura 34.

Figura 34 - Variação de Paradas por Estação



Fonte: Autor (2019)

Considere que a Estação 2 tem pouca demanda de passageiros, neste caso, pode-se determinar que a Estação 2 deva começar um ciclo com apenas uma locomotiva parada na estação, e que apenas o Trem 1 deve parar na Estação 2.

De forma análoga, caso a Estação 2 tenha alta demanda pode-se determinar que 3 carros devem aguardar nesta estação, como mostra a Figura 35.

Figura 35 - Variação de Paradas por Estação Parte 2



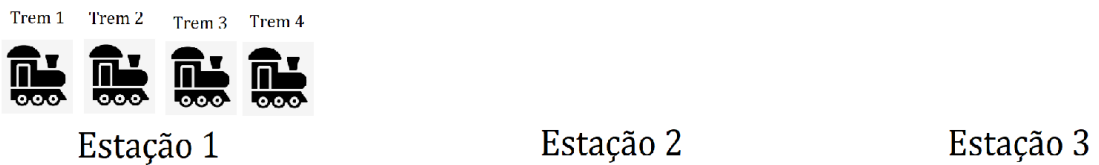
Fonte: Autor (2019)

Neste caso o Trem 4 não para na Estação 2, enquanto que os trens 1, 2 e 3 param.

5.4 CENÁRIO 4 – PARADA TOTAL

Também é possível que uma estação tenha uma demanda tão alta, que só pode ser atendida caso o comboio pare totalmente na estação, neste caso, deve ser determinado que esta estação de alta demanda não tenha nenhuma locomotiva parada nela no início de cada ciclo, como mostra a Figura 36, considerando que a Estação 2 tenha altíssima demanda.

Figura 36 - Parada Total



Fonte: Autor (2019)

Com isso, o comboio inteiro pode parar na Estação 2, e recomeçar o modo Sem-paradas a partir desta estação.

Portanto, a disposição das locomotivas pode ser feita de forma mista pela linha, de forma a atender as demandas eficientemente. É claro que essas mudanças afetariam a capacidade e tempo de viagem do modal, por isso sua aplicação deve ser estudada e bem planejada.

6 CONSIDERAÇÕES FINAIS

O sistema metroviário apresenta-se como uma alternativa de locomoção de passageiros em uma cidade com alta capacidade de transporte, porém com altos custos construção. A fim de tentar tornar este modal mais vantajoso, este trabalho introduziu um novo conceito de logística para o movimento das locomotivas, e testou sua viabilidade.

O conceito apresentou vantagens como, a redução dos tempos de viagem e, flexibilização da oferta de trens de acordo com a demanda de cada estação. Enquanto que, foi possível atender a demanda da linha metroviária estudada com a redução do *headway* aplicado.

Na realização dos cálculos, houve a simplificação do movimento das locomotivas que foi dividido em seis tipos de deslocamentos possíveis. Essa consideração foi eficaz e possibilitou a formulação do modelo matemático.

A maior dificuldade no desenvolvimento do trabalho se deu na construção do algoritmo genético, uma vez que o modelo conta com quatro variáveis bem distintas (velocidade, aceleração, tempo e posição) que restringem umas as outras. Portanto, foi necessário definir que apenas duas das quatro variáveis seriam variáveis de decisão, enquanto que os resultados das outras variáveis seriam obtidos por consequência das restrições.

No final do Capítulo 5 se observou um comportamento interessante, em que a função objetivo foi adaptada drasticamente para se minimizar o *headway*, contanto o resultado final obtido era apenas ligeiramente melhor que os resultados obtidos anteriormente. Isso mostra que a função objetivo que minimiza todos os tempos de viagem, também conseguiu garantir uma boa qualidade na capacidade de transporte de passageiros.

No início do Capítulo 5, a falta de dados específicos sobre o comportamento dos passageiros que acessam a *Red Line*, fez com que considerações tivessem que ser tomadas.

Por conta dessa falta de dados específicos, os cenários 2, 3 e 4 propostos no Capítulo 6 foram apenas apresentados e não testados, visto que sua aplicabilidade depende exatamente da demanda e comportamento dos passageiros por trecho.

Em geral, de acordo com os cálculos, dados e considerações feitas neste estudo o modo Sem-paradas pôde reduzir tempos de viagem em mais de 20% sem necessariamente

comprometer a capacidade da linha. Sendo seu maior limitante de capacidade máxima de transporte, o fluxo de passageiros pela porta que conecta as locomotivas.

6.1 TRABALHOS FUTUROS

Algumas variáveis não fizeram parte do escopo deste trabalho, uma delas sendo o traçado da linha, que influencia diretamente na velocidade máxima permitida para um trajeto.

Além disso, é evidente que este projeto envolveria tecnologias sofisticadas para ser empregado. Contudo o potencial deste modo de reduzir tempos de viagem é promissor, por isso seria interessante se trabalhos futuros coletassem dados e levassem em consideração os seguintes fatores:

- O traçado físico da linha;
- Coleta de dados para definir quantos passageiros estariam dispostos a atravessar locomotivas para economizar tempo;
- Coleta de dados para definir quantos passageiros desembarcam em cada estação;
- Estudos para definir com precisão o quão larga a porta que liga locomotivas pode ser, e qual o fluxo de passageiros que esta porta suportaria;
- Estudo para definir qual o nível de aceleração da locomotiva é apropriado, para que passageiros consigam se locomover dentro da locomotiva;
- Estudos do ponto de vista mecânico, para garantir que as locomotivas consigam se conectar com segurança.

REFERÊNCIAS

ALTIOK, T.; MELAMED, B. **Simulation Modeling and Analysis with ARENA**. New Jersey: Academic Press, 2007. 456 p.

CARDOSO, L. E. Z.; JOAQUIM J.; CELSO F. **Simulação computacional: análise comparativa dos softwares Arena® e Promodel®**. Tekhne e Logos, Botucatu, v. 7, n. 1, p.14-26, abr. 2016.

CLAUMANN, C. A. **MODELAGEM DINÂMICA E CONTROLE DE PROCESSOS NÃO LINEARES**: Uma aplicação de algoritmos genéticos para treinamento de redes neurais recorrentes. 1999. 221 f. Dissertação (Mestrado) - Curso de Pós-graduação em Engenharia Química, Universidade Federal de Santa Catarina, Florianópolis, 1999.

Companhia do Metropolitano de São Paulo. **RELATÓRIO INTEGRADO**. São Paulo, 2019.

COCA, A. C.; TORRES, I. G. E. **Transporte Público Urbano**. São Carlos: Rima, 2004.

COMITTEE, Finance And Administration. **7000 Series Railcar Program Contract Award**. Disponível em: <https://www.wmata.com/about/board/meetings/upload/032510_37000PresentationwithResolutions032510.pdf>. Acesso em: 10 jun. 2019.

FOURER, R.; GAY, D. M.; KERNIGHAN, B. W. **AMPL: A Mathematical Programming Language**. New Jersey: Duxbury Press, 2002.

GUALDA, N. D. F. **Terminais de transportes: Contribuição ao planejamento e ao dimensionamento operacional**. São Paulo, 288 p., 1995. Tese (Livre Docência), Departamento de Engenharia de Transportes, Escola Politécnica da Universidade de São Paulo (EPUSP).

HIBBELER, R. C. **Dinâmica: Mecânica para Engenharia**, 10 Edição. Rio de Janeiro: LTC Editora, 2005.

HILLIER, F. S.; LIEBERMAN, G. J. **Introdução à pesquisa operacional**. 10. ed. Nova Iorque: Mcgraw-hill Education, 2015.

HOEL, L. A. **Engenharia de Infraestrutura de Transportes: Uma Integração Multimodal**. São Paulo: Cengage Learning Nacional, 2012.

HOLLAND, J. H. **Adaptation in Natural and Artificial Systems**. Cambridge: Mit Press, 1975.

MITCHELL, M. **An Introduction to Genetic Algorithms**. Cambridge: Mit Press, 1998.

PEREIRA, C. R. **Apostila Arena 11.0**. 2009. 67 f. Dissertação (Mestrado) - Curso de Engenharia de Produção, Ufscar, São Carlos, 2009.

PRADO, D. S. **Usando o ARENA em Simulação**. Belo Horizonte: Editora de Desenvolvimento Gerencial, 1999.

PROFILLIDIS, V. A. **Railway Management and Engineering**. 3. ed. Farnham: Ashgate Publishing, 2006.

POWELL, J. P.; PALACÍN, R. **Passenger Stability Within Moving Railway Vehicles: Limits on Maximum Longitudinal Acceleration**. *Urban Rail Transit*, [s.l.], v. 1, n. 2, p.95-103, jun. 2015. Springer Nature. <http://dx.doi.org/10.1007/s40864-015-0012-y>.

TOINT, Philippe L.; CONN, Andrew R.; GOULD, G.i.m.. **LANCELOT: a Fortran Package for Large-Scale Nonlinear Optimization (Release A)**. Berlin: Springer Berlin Heidelberg, 1992.

VASCONCELLOS, E. A. **Mobilidade urbana e cidadania**. Rio de Janeiro: Senac, 2012.

WMATA. **2018 Historical Metrorail Ridership**. Disponível em: https://www.wmata.com/about/records/public_docs/upload/2018_historical_rail-ridership_May-weekday-avg.pdf. Acesso em: 11 jun. 2019.

Apêndice A – Resultado do Modelo Exato

Tabela 11 - Resultado Exato

Trajeto	Seção 1	Seção 2	Seção O	Seção G	Seção 3	Seção 4	Variável
0	1	0,2	-1	-1	-1	0,711725	Acel ($\frac{m}{s^2}$)
	9,62634	16,6217	15,6968	5,3678	0	13,015	Vel ($\frac{m}{s}$)
	9,61948	35	0,92	10,3208	5,36244	10,7464	Tempo (s)
	46,2675	459,422	14,8706	108,744	14,4308	98,8321	Dist (m)
1	1	0,2	0	-0,435361	-1	0,35623	Acel ($\frac{m}{s^2}$)
	13	20	20	11,0917	0	16,0886	Vel ($\frac{m}{s}$)
	12,9925	35	28,3754	20,4468	11,0872	14,0259	Tempo (s)
	84,4024	577,823	567,609	317,931	61,574	190,689	Dist (m)
2	1	0,11215	0	-1	-1	0,696621	Acel ($\frac{m}{s^2}$)
	16,0834	20	20	4,98416	0	11,7691	Vel ($\frac{m}{s}$)
	16,0696	35	24,3756	15	5	9,71143	Tempo (s)
	129,116	631,613	487,587	187,542	12,5477	81,5021	Dist (m)
3	1	0,2	0,147598	-1	-1	0,72234	Acel ($\frac{m}{s^2}$)
	11,7634	18,7655	20	5,48783	0	13,6174	Vel ($\frac{m}{s}$)
	11,7436	35	8,3784	14,5042	5,49051	11,2457	Tempo (s)
	68,9555	534,218	162,406	184,899	15,1258	107,531	Dist (m)
4	1	0,182826	0	-0,82272	-1	0,674361	Acel ($\frac{m}{s^2}$)
	13,6112	20	20	6,52272	0	14,7714	Vel ($\frac{m}{s}$)
	13,5946	35	26,4475	16,3739	6,53156	12,2125	Tempo (s)
	92,4065	588,375	529,037	217,191	21,3957	130,179	Dist (m)
5	1	0,14983	0	-0,951638	-1	0,719967	Acel ($\frac{m}{s^2}$)
	14,7657	20	20	8,07267	0	20	Vel ($\frac{m}{s}$)
	14,7502	35	26,1147	12,5288	8,08724	16,5405	Tempo(s)
	108,784	608,572	522,38	175,887	32,7825	232,421	Dist (m)
6	1	0	0	-0,023546	-1	-0,16187	Acel ($\frac{m}{s^2}$)
	19,9954	20	20	19,325	0	16,4116	Vel ($\frac{m}{s}$)
	19,982	35	38,4841	28,3583	19,3151	17,9681	Tempo (s)
	199,639	700,074	769,868	557,699	186,73	321,107	Dist (m)

Continua

Tabela 12 - Resultado Modelo Exato (Continua)

Trajeto	Seção 1	Seção 2	Seção O	Seção G	Seção 3	Seção 4	Variavel
7	1	0,102894	0	-0,359017	-1	0,359609	Acel ($\frac{m}{s^2}$)
	16,4065	20	20	13,7254	0	20	Vel ($\frac{m}{s}$)
	16,3938	35	30,4831	17,4629	13,7267	17,4308	Tempo(s)
	134,377	637,251	609,779	294,517	94,3489	294,074	Dist (m)
8	1	0	0	0	-1	-0,0827997	Acel ($\frac{m}{s^2}$)
	19,9975	20	20	20	0	18,4039	Vel ($\frac{m}{s}$)
	19,9863	35	62,4578	43,4482	19,99	19,2156	Tempo (s)
	199,726	7,11	1249,64	869,2	200	369,025	Dist (m)
9	1	0,0459735	0	-0,173014	-1	0,0265152	Acel ($\frac{m}{s^2}$)
	18,3987	20	20	15,7951	0	16,2183	Vel ($\frac{m}{s}$)
	18,3862	35	34,5841	24,2645	15,7893	15,9908	Tempo (s)
	169,026	672,114	691,835	434,358	124,808	256,033	Dist (m)
10	1	0,108419	0	-0,338022	-1	0,338931	Acel ($\frac{m}{s^2}$)
	16,2133	20	20	14,0532	0	20	Vel ($\frac{m}{s}$)
	16,2	35	30,3417	17,5777	14,0536	17,5304	Tempo (s)
	131,225	633,872	606,951	299,334	98,892	298,62	Dist (m)
11	1	0	0	0	-1	0	Acel ($\frac{m}{s^2}$)
	19,9976	20	20	20	0	20	Vel ($\frac{m}{s}$)
	19,9864	35	62,512	43,3938	19,99	19,9914	Tempo (s)
	199,728	700,111	1250,73	868,111	200	399,879	Dist (m)
12	1	0	0	0	-0,233937	---	Acel ($\frac{m}{s^2}$)
	20	20	20	20	0	---	Vel ($\frac{m}{s}$)
	19,994	35	72,2731	49,1421	85,3163	---	Tempo (s)
	199,88	700,215	1446,11	983,15	854,927	---	Dist (m)

Fonte: Autor (2019)

Apêndice B – Algoritmo de Cálculo dos Tempos de Viagem

Figura 37 - Algoritmo de Cálculo dos Tempos de Viagem

```

float tot = 0;
float dil = 0;
ofstream rata(data);
if(rata.is_open())
{
    for(int i = 0; i<numest-1; i++)
    {
        if(i==0)
        {
            tot = T1[i]+T2[i]+TO[i]+TG[i];
            dil = tot + T3[i];
            rata << "Tempo para a estacao: " << i+1 << " = " << dil << " segundos." << endl
<< endl;
            dil = dil+25;
        }
        else
        {
            if(i%2 ==1)
            {
                tot = tot +T2[i]+TO[i]+TG[i] + T4[i-1];
                rata << "Modo em movimento: Tempo para a estacao: " << i+1 << " = " << (tot
+ T3[i])/60.0<< " minutos."<< endl;
                rata << "Modo sentado:    Tempo para a estacao: " << i+1 << " = " << (dil
+T3[i] + T1[i] + T2[i] + TO[i]+TG[i])/60.0<< " minutos."<< endl << endl;
            }
            else
            {
                tot = tot +T2[i]+TO[i]+TG[i]+T4[i-1];
                dil = dil + T1[i-1] + T2[i-1] + TO[i-1] + TG[i-1] + T4[i-1] + T2[i] + TO[i] +
TG[i] + T3[i];
                rata << "Modo em movimento: Tempo para a estacao: " << i+1 << " = " << (tot
+ T3[i])/60.0<< " minutos."<< endl;
                rata << "Modo sentado:    Tempo para a estacao: " << i+1 << " = " <<
dil/60.0<< " minutos."<< endl << endl;
                dil = dil +25;
            }
        }
    }
}
}

```

Fonte: Autor (2019)

Apêndice C – Algoritmo de Crossover

Figura 38 - Crossover Parte 1

```
string cross(string x,string y,long int siz)
{
    long int p = random(11,1);
    string r;
    r =x;
    long int k =0;
    if(p ==1)
    {
        for(int i =0; i<siz; i=i+2)
        {
            r[i] = y[i];
        }
    }
    if(p ==2)
    {
        for(int i =0; i<siz; i=i+3)
        {
            r[i] = y[i];
        }
    }
    if(p ==3)
    {
        for(int i =0; i<siz; i=i+4)
        {
            r[i] = y[i];
        }
    }
    if(p ==4)
    {
        for(int i =0; i<siz; i=i+5)
        {
            r[i] = y[i];
        }
    }
    if(p ==5)
    {
        for(int i =0; i<siz; i=i+6)
        {
            r[i] = y[i];
        }
    }
}
```

Fonte: Autor (2019)

Figura 39- Crossover Parte 2

```
if(p==6)
{
  for(int i=0; i<siz; i++)
  {
    if(k!=2)
    {
      r[i] = y[i];
      k++;
    }
    else
    {
      k=0;
      i = i+2;
    }
  }
}
if(p==7)
{
  for(int i=siz/2; i<siz; i++)
  {
    r[i] = y[i];
  }
}
if(p==8)
{
  for(int i=siz/3; i<siz; i++)
  {
    r[i] = y[i];
  }
}
if(p==9)
{
  for(int i=siz/4; i<siz; i++)
  {
    r[i] = y[i];
  }
}
if(p==10)
{
  for(int i=siz - (siz/4); i<siz; i++)
  {
    r[i] = y[i];
  }
}
if(p==11)
{
  for(int i=siz - (siz/3); i<siz; i++)
  {
    r[i] = y[i];
  }
}
}
```

Apêndice D – Algoritmo Mutaç o

Figura 40 - Mutaç o

```
string mut(string j,long int siz)
{
    long int o;
    for(int i=0; i<siz; i++)
    {
        o = random(100,0);
        if(o< 5)
        {
            if(j[i] == '1')
            {
                j[i] = '0';
            }
            else
            {
                j[i] = '1';
            }
        }
    }
    return j;
}
```

Fonte: Autor (2019)

Apêndice E – Resultado Modelo Genético

Tabela 13 - Resultado Modelo Genético

Trajeto	Seção 1	Seção 2	Seção O	Seção G	Seção 3	Seção 4	Variável
0	0.532	-0.442	0.055	0	-0.092	0.428	Aceleração ($\frac{m}{s^2}$)
	16.5	1	2.5	0	0	14.5	Velocidade ($\frac{m}{s}$)
	31	35	27	0	27	28	Tempo (s)
	255.75	306.25	47.25	0	33.75	238	Distância (m)
1	0.514	0.257	-0.422	0	-0.091	0.444	Aceleração ($\frac{m}{s^2}$)
	14.5	23.5	4.5	0	0	16.5	Velocidade ($\frac{m}{s}$)
	28.1724	35	45	0	492.222	27	Tempo (s)
	204.25	665	630	0	110.75	283.5	Distância (m)
2	0.787	-0.214	0.607	0	-0.997	-0.322	Aceleração ($\frac{m}{s^2}$)
	16.5	9	26	0	0	16	Velocidade ($\frac{m}{s}$)
	20.939	35	28	0	26.076	31	Tempo (s)
	172.75	446.25	490	0	339	651	Distância (m)
3	0.410	-0.257	-0.153	0	-0.037	0.310	Aceleração ($\frac{m}{s^2}$)
	16	7	3	0	0	14.5	Velocidade ($\frac{m}{s}$)
	39	35	26	0	81	37	Tempo (s)
	312	402.5	130	0	121.5	323.75	Distância (m)
4	0.519	-0.2	1	0	-0.841	-0.154	Aceleração ($\frac{m}{s^2}$)
	14.5	7.5	28.5	0	0	22.5	Velocidade ($\frac{m}{s}$)
	27.896	35	21	0	33.877	39	Tempo (s)
	202.25	385	378	0	482.75	994.5	Distância (m)
5	0.494	-0.2	-0.483	0	-0.021	0.562	Aceleração ($\frac{m}{s^2}$)
	22.5	15.5	1	0	0	19	Velocidade ($\frac{m}{s}$)
	45.488	35	30	0	47.5	32	Tempo (s)
	511.75	665	247.5	0	23.75	320	Distância (m)
6	0.609	-0.157	0.328	0	-0.423	-0.173	Aceleração ($\frac{m}{s^2}$)
	19	13.5	26	0	0	17	Velocidade ($\frac{m}{s}$)
	31.184	35	38	0	61.4231	52	Tempo (s)
	296.25	568.75	750.5	0	798.5	1118	Distância (m)
7	0.452	0.057	-0.16	0	-0.846	0.209	Aceleração ($\frac{m}{s^2}$)
	17	19	11	0	0	24	Velocidade ($\frac{m}{s}$)
	37.588	35	50	0	13	62	Tempo (s)
	319.5	630	750	0	71.5	1085	Distância (m)

Continua

Tabela 14 - Resultado Modelo Genético (Continua)

Trajetos	Seção 1	Seção 2	Seção O	Seção G	Seção 3	Seção 4	Variável
8	0.284	-0.214	0.402	0	-0.749	-0.5	Aceleração ($\frac{m}{s^2}$)
	24	16.5	31	0	0	12.5	Velocidade ($\frac{m}{s}$)
	84.458	35	36	0	41.338	37	Tempo (s)
	1013.5	708.75	855	0	640.75	804.75	Distância (m)
9	0.476	-0.028	0.27	0	-0.523	-0.187	Aceleração ($\frac{m}{s^2}$)
	12.5	11.5	25	0	0	17.5	Velocidade ($\frac{m}{s}$)
	26.24	35	50	0	47.72	40	Tempo (s)
	164	420	912.5	0	596.5	850	Distância (m)
10	0.604	-0.0714	0.529	0	-0.467	-0.17	Aceleração ($\frac{m}{s^2}$)
	17.5	15	24	0	0	15.5	Velocidade ($\frac{m}{s}$)
	28.971	35	17	0	51.354	50	Tempo (s)
	253.5	568.75	331.5	0	616.25	987.5	Distância (m)
11	0.323	0.2	0.225	0	-0.450	-0.107	Aceleração ($\frac{m}{s^2}$)
	15.5	22.5	31.5	0	0	25.5	Velocidade ($\frac{m}{s}$)
	47.903	35	40	0	69.952	56	Tempo (s)
	371.25	665	1080	0	1101.75	1596	Distância (m)
12	0.657	0.085	0.024	0	-0.498	---	Aceleração ($\frac{m}{s^2}$)
	25.5	28.5	30	0	0	---	Velocidade ($\frac{m}{s}$)
	38.764	35	63	0	60.2	---	Tempo (s)
	494.25	945	1842.75	0	903	---	Distância (m)

Fonte: Autor (2019)

Apêndice F – Codificação Binária

Figura 41 - Codificação Binária

```
for(int i = 0; i<(tamest-1)*2; i++)
{
    x = bitset<6>(vc[i]*2).to_string();
    for(int j = 0; j< 6; j++)
    {
        J.push_back(x[j]);
    }
    x.clear();

    x = bitset<6>(tc[i]).to_string();
    for(int j = 0; j< 6; j++)
    {
        J.push_back(x[j]);
    }
    x.clear();

    if(i%2 == 1)
    {
        x = bitset<6>(vo[(i-1)/2]*2).to_string();
        for(int j = 0; j< 6; j++)
        {
            J.push_back(x[j]);
        }
        x.clear();

        x = bitset<6>(to[(i-1)/2]).to_string();
        for(int j = 0; j< 6; j++)
        {
            J.push_back(x[j]);
        }
        x.clear();
    }
}
```

Fonte: Autor (2019)