



**UNIVERSIDADE FEDERAL DO AMAZONAS**  
**FACULDADE DE TECNOLOGIA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

**Péricles Aparecido Vasconcelos Balderrama**

**PROPOSTA DE HEURÍSTICA BASEADA NO  
CONCEITO DE MERCADO PARA GERAÇÃO  
DE ROTAS**

Manaus

2018

**Péricles Aparecido Vasconcelos Balderrama**

**PROPOSTA DE HEURÍSTICA BASEADA NO  
CONCEITO DE MERCADO PARA GERAÇÃO DE  
ROTAS**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, como requisito parcial para obtenção do Título de Mestre em Engenharia Elétrica. Área de concentração: Automação e Controle.

Universidade Federal do Amazonas

Faculdade de Tecnologia

Programa de Pós-Graduação em Engenharia Elétrica

Orientador: Prof. Dr. André Luiz Duarte Cavalcante

Manaus

2018

## Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

B176p Balderrama, Péricles Aparecido  
Proposta de heurística baseada no conceito de mercado para  
geração de rotas / Péricles Aparecido Balderrama. 2018  
68 f.: il. color; 31 cm.

Orientadora: André Luiz Duarte Cavalcante  
Dissertação (Mestrado em Engenharia Elétrica) - Universidade  
Federal do Amazonas.

1. Roteização. 2. Mercado Econômico. 3. Grafos . 4. otimização.  
5. Simulações. I. Cavalcante, André Luiz Duarte II. Universidade  
Federal do Amazonas III. Título

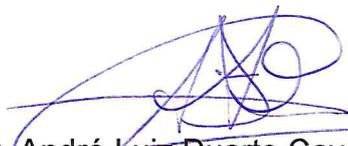
PÉRICLES APARECIDO VASCONCELOS BALDERRAMA

**PROPOSTA DE HEURÍSTICA BASEADA NO CONCEITO DE MERCADO  
PARA GERAÇÃO DE ROTAS**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Amazonas, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica na área de concentração Controle e Automação de Sistemas.

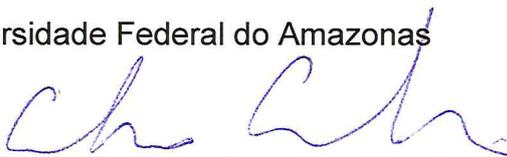
Aprovado em 09 de fevereiro de 2018.

**BANCA EXAMINADORA**



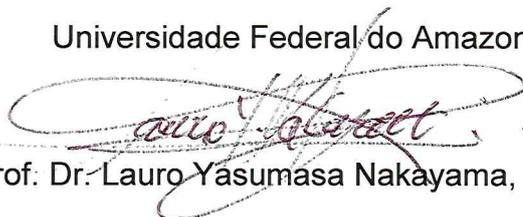
Prof. Dr. André Luiz Duarte Cavalcante, Presidente

Universidade Federal do Amazonas



Prof. Dr. Celso Barbosa Carvalho, Membro

Universidade Federal do Amazonas



Prof. Dr. Lauro Yasumasa Nakayama, Membro

INPA

*Este trabalho é dedicado  
a minha mãe, criatura amada que  
esteve ao meu lado nesta Jornada*

# Agradecimentos

Agradeço, primeiramente à Deus, à minha querida **Mãe Marta Vasconcelos** que durante esta caminhada voltou ao braços do nosso amado senhor, e ao meu **pai José Balderrama** que juntos são meus pilares de sustentação para quais tenho um amo eterno. A minha esposa e companheira **Francimara Rodrigues** que sempre esteve ao meu lado nesta jornada sendo por muitas vezes minha bússola.

Agradeço emensamente aos meus amigos e colegas do mestrado Luiz Clodovil, Yadini López, Luis López, José do patrocineo e Tayler jo'el que sempre estiveram dispostos a ajudar nos momentos de dificuldades. Ao amigo, companheiro e mestre **André Cavalcante** pela paciência infinita e aos demais professores os quais contribuíram para a produção deste trabalhos.

*“A sabedoria é resplandecente, não murcha,  
mostra-se facilmente para aqueles que a amam.  
Ela se deixa encontrar por aqueles que a buscam.  
(Bíblia Sagrada, Sabedoria 6, 12)*

# Resumo

Este trabalho propõe uma heurística que utiliza o conceito de mercado e no processo de formação de preço como diretrizes para a geração de rotas em um cenário de uma planta fabril. Assume-se como premissa que o mercado econômico é eficiente em alocar os recursos escassos e que o preço aglutina em um único número a complexidade do processo produtivo, simplificando o sistema de avaliação dos agentes econômicos. Desta forma, adota-se o preço como principal variável na seleção dos trechos que compõem as rotas. A definição de preço considerada na proposta diferencia-se do custo por se uma variável atualizada considerando unicamente o mercado, ou seja, as interações entre os agentes econômicos. O mercado é modelado e implementado para simular a movimentação de insumos em um espaço fabril, espaço este constituído por células de produção interligadas por uma grade de transporte na qual os robôs moveis transportam insumos entre o depósito central e as células de produção. No contexto proposto os robôs são agentes econômicos consumidores e os trechos da grade são os produtos negociados, existe um vendedor que atende a todos o conjunto de robôs. Na implementação da heurística proposta o algoritmo Dijkstra é utilizado para detectar aos trechos que forma a rota com o preço mínimo no mercado em determinado instante.

**Palavras-chave:** Heurística, Mercado econômico, Formação de preços, Simulação, Otimização e teoria de grafos.

# Abstract

This dissertation proposes a heuristic that uses the concept of market and the process of price formation as guidelines for the generation of routes in a factory plant scenario. The assumption is that the economic market is efficient in allocating scarce resources and that the price agglutinates in a single number the complexity of the productive process, simplifying the system of evaluation of economic agents. In this way, the price is adopted as the main variable in the selection of the sections that compose the routes. The price definition considered in the proposal differs from the cost per an updated variable considering only the market, that is, the interactions between economic agents. The market is modeled and implemented to simulate the movement of inputs in a manufacturing space, this space consisting of production cells interconnected by a transport grid in which the mobile robots carry inputs between the central warehouse and the production cells. In the proposed context robots are consumer economic agents and the crate of the grid are the traded products, there is a vendor that caters to all the robots. In the implementation of the proposed heuristic the Dijkstra algorithm is used to detect to the stretches that form the route with the minimum price in the market at a certain instant.

**Keywords:** Economic market, heuristics, price formation, simulation, optimization, graph theory.

# Lista de ilustrações

Figura 1 – Problema das Pontes de <i>Königsberg</i> . . . . .	4
Figura 2 – Grafos . . . . .	5
Figura 3 – Grafo Direcionado . . . . .	6
Figura 4 – Grafo Ponderado . . . . .	6
Figura 5 – Matriz de Adjacências . . . . .	7
Figura 6 – Lista adjacencia . . . . .	7
Figura 7 – Grafo-Caminho Mínimo e Máximo . . . . .	12
Figura 8 – Interações do Dijkstra . . . . .	19
Figura 9 – Curva de Oferta . . . . .	25
Figura 10 – Curva de Demanda . . . . .	26
Figura 11 – Ponto de Equilíbrio . . . . .	27
Figura 12 – Curva de Equilíbrio Avaliação . . . . .	28
Figura 13 – Modelo da teia de aranha de ciclo convergente. . . . .	29
Figura 14 – Modelo da teia de aranha de ciclo divergente. . . . .	29
Figura 15 – Modelo da teia de aranha de ciclo contínuo. . . . .	29
Figura 16 – Modelo Planta Fabril . . . . .	31
Figura 17 – Visão Geral - Simulações . . . . .	34
Figura 18 – Mercado Relações Básicas . . . . .	36
Figura 19 – Cenário Proposto . . . . .	37
Figura 20 – Diagrama de classe UML- Planta Fabril . . . . .	39
Figura 21 – Classe Vendedor - Modelada em UML . . . . .	40
Figura 22 – Classes PCP e Requisição em UML . . . . .	41
Figura 23 – Máquina de Estado - Classe Robô . . . . .	44
Figura 24 – Diagrama de classe completo - Simulação . . . . .	45
Figura 25 – Planta Fabril Rotas diversificadas . . . . .	48
Figura 26 – Planta Fabril Rotas Fixas . . . . .	51
Figura 27 – Relação entre $\alpha$ e deslocamento dos robôs . . . . .	55
Figura 28 – Peso demanda x deslocamento robôs - Função Custo . . . . .	56
Figura 29 – Desvio Médio - Função Preço . . . . .	57
Figura 30 – Desvio Médio - Função Custo . . . . .	58
Figura 31 – Desvio Padrão - Função Preço . . . . .	59
Figura 32 – Distribuição das rotas fixas - Função Preço . . . . .	60
Figura 33 – Distribuição das rotas fixas - Função Custo . . . . .	61

# Lista de tabelas

Tabela 1 – Aplicações TSP Parâmetros . . . . .	21
Tabela 2 – Aplicações VRP Parâmetros . . . . .	22

# Lista de abreviaturas e siglas

<b>VRP</b>	<i>Vehicle Routing Problem</i>
<b>TSP</b>	<i>Traveling Salesman problem</i>
<b>CVRP</b>	<i>Capacitated Vehicle Routing Problem</i>
<b>VRPB</b>	<i>Vehicle Routing Problem Backhauls</i>
<b>TSPSC</b>	<i>Symmetric Clustered Traveling Salesman Problem</i>
<b>TSP-PC</b>	<i>Prize-Collecting Traveling Salesman Problem</i>
<b>mTSP</b>	<i>Multiple Traveling Salesman Problem</i>
<b>TSP-WT</b>	<i>Traveling Salesman Problem With Window Time</i>
<b>SP</b>	<i>Shortest Path Problem</i>
<b>FSM</b>	<i>Finite State Machine</i>

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>1</b>
<b>1.1</b>	<b>Definição do problema e hipótese de trabalho</b>	<b>1</b>
1.1.1	Definição do problema	1
1.1.2	Hipótese de trabalho	2
<b>1.2</b>	<b>Objetivo Geral</b>	<b>2</b>
<b>1.3</b>	<b>Objetivos Específicos</b>	<b>2</b>
<b>1.4</b>	<b>Metodologia</b>	<b>2</b>
<b>1.5</b>	<b>Contribuição</b>	<b>3</b>
<b>1.6</b>	<b>Organização do Trabalho</b>	<b>3</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>4</b>
<b>2.1</b>	<b>Teoria de Grafo</b>	<b>4</b>
2.1.1	Tipo de Grafos	5
2.1.2	Representações de Grafos	6
<b>2.2</b>	<b>Teoria da Complexidade Computacional</b>	<b>7</b>
2.2.1	Comportamento Assintótico de Funções	8
<b>2.3</b>	<b>Problema do Caminho Mínimo</b>	<b>10</b>
2.3.1	Aplicações	13
2.3.2	Caminho Mínimo com Peso Limitado	14
2.3.3	Caminho Mínimo com Recursos Limitados	14
2.3.4	K-Caminho Mínimos	15
2.3.5	O Problema do Caminho Mínimo com Incertezas	15
2.3.6	O Problema do Caminho Mínimo Robusto	16
2.3.7	Algoritmos para Solução do Problema do Caminho Mínimo	17
2.3.8	Algoritmo de Dijkstra	17
<b>2.4</b>	<b>Problema do Caxeiro Viajante (TSP)</b>	<b>20</b>
2.4.1	Aplicação do Traveling Salesman Problem	21
<b>2.5</b>	<b>Problema de Roteirização Veicular (VRP)</b>	<b>21</b>
2.5.1	Aplicações Problema Roteirização de Veículos	22
<b>2.6</b>	<b>Máquina de Estados</b>	<b>23</b>
<b>2.7</b>	<b>Microeconomia</b>	<b>24</b>
2.7.1	Mercado	24
2.7.2	Ponto de Equilíbrio	26
2.7.3	Teoria da teia de aranha	26
<b>3</b>	<b>PROPOSTA</b>	<b>30</b>

<b>3.1</b>	<b>Cenário</b> . . . . .	<b>31</b>
<b>4</b>	<b>IMPLEMENTAÇÃO</b> . . . . .	<b>35</b>
<b>4.1</b>	<b>Mercado</b> . . . . .	<b>35</b>
4.1.1	Cenário . . . . .	35
4.1.2	Grid_Aresta . . . . .	36
4.1.3	Grid_Vertices . . . . .	38
4.1.4	Planta_fabril . . . . .	38
4.1.5	Vendedor . . . . .	39
4.1.6	Requisicao . . . . .	40
4.1.7	PCP . . . . .	40
4.1.8	Dijkstra . . . . .	41
4.1.9	Robo . . . . .	42
4.1.10	Classe Main . . . . .	44
<b>5</b>	<b>EXPERIMENTOS E RESULTADOS</b> . . . . .	<b>46</b>
<b>5.1</b>	<b>Função Preço X Função Custo</b> . . . . .	<b>46</b>
5.1.1	Função Preço . . . . .	46
5.1.2	Função Custo . . . . .	47
<b>5.2</b>	<b>Roteiro Experimental</b> . . . . .	<b>47</b>
5.2.1	Cenário 1 . . . . .	48
5.2.1.1	Métricas avaliativas - Cenário 01 . . . . .	49
5.2.1.1.1	Somatória das distâncias . . . . .	49
5.2.1.1.2	Desvio Médio . . . . .	50
5.2.1.1.3	Desvio padrão populacional . . . . .	50
5.2.2	Cenário 2 . . . . .	51
5.2.2.1	Métrica avaliativa - Cenário 02 . . . . .	52
5.2.2.1.1	Densidade de utilização das rotas . . . . .	52
5.2.3	Roteiro da Simulação . . . . .	53
<b>5.3</b>	<b>Ambiente computacional</b> . . . . .	<b>54</b>
<b>5.4</b>	<b>Resultados</b> . . . . .	<b>54</b>
5.4.1	Coefficiente de valorização $\alpha$ verso deslocamento dos robôs - Função Preço	55
5.4.2	Peso da demanda x deslocamento dos robôs - Função Custo . . . . .	56
5.4.3	Coefficientes x Desvio médio das distâncias das rotas - Função Preço . . . . .	57
5.4.4	Peso demanda x Desvio médio das rotas - Função Custo . . . . .	58
5.4.5	Coefficientes x Desvio padrão das distâncias das rotas-Função Preço . . . . .	59
5.4.6	Avaliação dos densidade de rotas mediante a variação dos coeficientes de atualização de preços - Função Preço . . . . .	60
5.4.7	Avaliação dos densidade de rotas mediante a variação peso da demanda - Função Custos . . . . .	61

<b>6</b>	<b>ANÁLISE E CONCLUSÃO . . . . .</b>	<b>62</b>
<b>6.1</b>	<b>Propostas para trabalhos Futuros . . . . .</b>	<b>63</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>64</b>

# 1 Introdução

Pesquisas de algoritmos com novas abordagens na resolução de problemas de roteirização são relevantes por tratar-se de problema classificado como **NP**. Tais problemas em geral possuem ordem de complexidade exponencial. Fato que motiva o desenvolvimento de abordagem utilizando heurísticas e metaheurísticas para a solução de problemas desta natureza. Outra perspectiva relevante é o impacto financeiro das atividades pertinentes ao transporte e distribuição de insumos que representam significativa parcela na composição dos custos das empresas, visto que os investimentos relacionados ao transporte geralmente são elevados, absorvendo porcentagem significativa do investimento das atividades logísticas. Justifica-se dessa forma, a relevância de novas heurísticas que minimizem custos.

Diferente das abordagens tradicionais de roteirização que geralmente são baseados na avaliação dos custos, propõe-se neste trabalho uma abordagem baseada na análise do preço praticado em um mercado virtual de rotas.

No modelo proposto, as rotas são geradas considerando o preço associado a cada trecho que a constitui, sendo considerada a melhor rota a que apresenta o **preço mínimo**. Nesta perspectiva o preço é uma variável dinâmica que será atualizada mediante ao uso das rotas pelos agentes que compõem o sistema de movimentação de insumos.

Pretende-se com a utilização do conceito de preço conseguir o comportamento similar aos dos agentes econômicos em um mercado que aloca de forma eficiente e dinâmica os recursos escassos. Para isso, dispõe-se da teoria de formação de preços e a lei da oferta e demanda.

## 1.1 Definição do problema e hipótese de trabalho

### 1.1.1 Definição do problema

Geração de rotas com o controle, balanceamento e a minimização do total de deslocamento de forma dinâmica. Este problema ainda é um desafio para a comunidade científica, por tratar-se de um problema aplicável em vários contextos reais, e apesar dos avanços dos algoritmos exatos e o melhor desempenho computacional com os *hardwares* atuais, não se conhece um algoritmo com tempo polinomial capaz de localizar soluções ótimas. Este problema encontra-se em pautas de artigos científicos. Propõe-se uma abordagem utilizando conceito de mercado econômico que objetiva o balanceamento e a minimização da distância percorrida. Busca-se obter resultados eficientes ponderando estes dois objetivos.

### 1.1.2 Hipótese de trabalho

Assume-se por premissa que as relações de mercado entre agentes econômicos intercambiadas pelo preço tendem a alocar eficientemente e dinamicamente os recursos escassos na economia(SARAIVA, 2011). Admite-se como hipótese que seja possível desenvolver uma heurística para geração de rotas, utilizando os conceitos e comportamento de mercado econômico que aloque de forma dinâmica e eficiente as rotas.

## 1.2 Objetivo Geral

Desenvolver e implementar uma heurística para geração de rotas utilizando conceitos de mercado e formação de preços que otimize o uso das rotas.

## 1.3 Objetivos Específicos

- . Desenvolver heurística utilizando conceito de formação de preço para geração de rotas;
- Implementar simulação de mercado de rotas;
- Modelar um cenário para simulação das relações entre os agentes econômicos;
- Parametrizar as características do mercado simulado;
- Avaliar as rotas geradas pela simulação;
- Analisar o comportamento das rotas geradas mediante as variações dos parâmetros do mercado.

## 1.4 Metodologia

- Estudar as implementações de algoritmos de roteirização existentes;
- Estudar as teorias de formação de preços e microeconomia;
- Elencar os requisitos necessário para elaboração da simulação;
- Definir o escopo e as funcionalidades do simulador do mercado virtual;
- Analisar os dados obtidos, apresentá-los e discuti-los.

## 1.5 Contribuição

Ao considerar o mercado como modelo de sistema para alocação de recursos escassos, a proposta traz uma nova perspectiva de pesquisa, sendo assim a principal contribuição deste trabalho. Trata-se de uma abordagem inovadora por utilizar conceitos de mercado e processo de formação dos preços para a elaboração de uma heurística que otimize a utilização das rotas em um espaço fabril.

Investiga-se a possibilidade de simplificar o cálculo da função de custo do algoritmo de caminho mínimo adotando a função de preço mínimo para realizar a criação das rotas de forma eficiente, tendo como principal parâmetro o preço.

O trabalho tem como fruto a criação de um mercado virtual de rotas com a simulação das interações entre os agentes e uma avaliação do comportamento dos robôs mediante as modificações dos parâmetros de mercado.

## 1.6 Organização do Trabalho

Este trabalho está organizado conforme a lista a seguir:

Capítulo 1 é esta introdução que contém a definição do problema, a hipótese, o objetivos e a metodologia;

No capítulo 2 são apresentadas as fundamentações teóricas, que contém os conceitos básicos da teoria de grafos, problema do caminho mínimo, algoritmo de Dijkstra, problema de caixeiro viajante, problema de roteirização veicular, outros que serão usados neste trabalho, assim como definições de microeconomia, mercado, heurística, máquina de estado e de simulações;

No capítulo 3 é apresentada a proposta deste trabalho que é o desenvolvimento uma heurística utilizando o conceito de mercado e do processo de formação de preço;

No capítulo 4 contém a implementação da proposta. Nele serão apresentados todos os procedimentos necessários ao desenvolver um simulador do mercado virtual de rotas;

No capítulo 5 apresentará os experimentos realizados e os resultados obtidos por meio das simulações de um mercado virtual de rotas aplicado a um cenário de uma planta fabril;

No capítulo 6 será realizada a análise dos resultados dos experimentos realizados;

O capítulo 07 apresentará as conclusões deste trabalho, discutirá e comparará os resultados obtidos e apresentará a proposta de trabalhos futuros.

## 2 Fundamentação Teórica

Neste texto supõe-se que o leitor tem conhecimento prévio em diversos temas como teoria dos grafos, problema do caminho mínimo, problema do caixeiro viajante, problema de roteirização, conceitos de microeconomia, heurística e conceito de simulações. Este capítulo objetiva apresentar, de forma sucinta, os conceitos fundamentais ao entendimento desta proposta. Sugere-se ao leitor não familiarizado com algum destes conteúdos que leia o capítulo e em caso de dúvida busque as publicações referenciadas na bibliografia desta obra.

### 2.1 Teoria de Grafo

A Teoria dos Grafos surgiu com os trabalhos de L. Euler, G. Kirchho e A.Cayley. O primeiro e mais famoso problema é o chamado o problema das pontes de *Königsberg*, que foi enunciado por Euler em 1736. Na cidade de Königsberg (antiga Prússia), existiam sete pontes que cruzavam o rio Pregel estabelecendo ligações entre diferentes partes da cidade e a ilha Kneiphof conforme a figura 1. O problema consistia em determinar se era possível fazer um passeio pela cidade começando e terminando no mesmo lugar, cruzando cada ponte exatamente uma única vez (FEOFILOFF; KOHAYAKAWA; WAKABAYASHI, 2011).

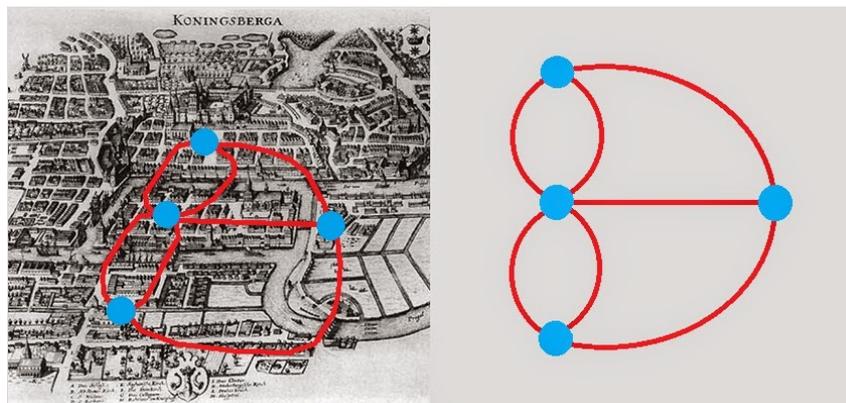


Figura 1 – Problema das Pontes de *Königsberg*

Fonte (GOLDBARG, 2012) Adaptada.

Euler demonstrou que somente é possível realizar o trajeto passando uma única vez em cada ponte e retornando ao ponto de partida, se houvesse, partindo de cada vértice, um número par de arestas. Assim, todos os vértices deveriam ser de grau par. O problema tornou-se ponto de partida para o desenvolvimento da Teoria dos Grafos.

Grafo é uma representação visual de um determinado conjunto de dados e da ligações existentes entre os elementos desse conjunto.

Em muitos dos problemas que nos surgem, uma das forma mais simples de descreve-lo é representá-lo em forma de grafo, uma vez que a representação visual apresentam vantagens na construção de um modelo matemático com vista à resolução do problema. Por esta razão muitas áreas do conhecimento tais como, matemática, telecomunicações, engenharia civil e gestão recorrem à teoria de grafos para resolver problemas complexos.

Pode-se definir um grafo  $G$  por  $G = (V, A)$ , sendo que  $V$  representa o conjunto de vértices e  $A$  o conjunto de arestas  $(i, j)$ , onde  $(i, j) \in V$ . Os vertices são denominados vizinhos quando estão conectados por uma aresta. Adotaremos a seguinte simbologia  $i \sim j$  para vertices vizinhos. Na figura 2 apresenta-se dois exemplos de grafos: o grafo  $A$  constituído pelo conjunto  $V = \{a, b, c, d\}$  e  $A = \{e1, e2, e3, e4, e5, e6\}$  e o grafo  $B$  constituído por  $V = \{a, b, c, d\}$  e  $A = \{e1, e2, e3, e4, e5, e6\}$  e apresenta um vértice não conectado com nenhum outro vértice do grafo. Quando o grafo possui mais vertices não-adjacentes do que adjacentes, ele é denominado de grafo esparsos. As arestas associadas a um mesmo vértice constituem um laço ou *loop* (FEOFILOFF; KOHAYAKAWA; WAKABAYASHI, 2011)

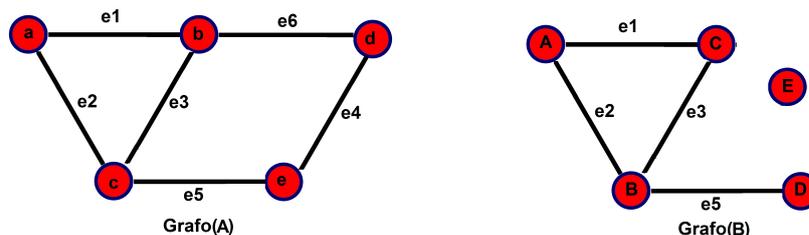


Figura 2 – Grafos

Fonte: (GOLDBARG, 2012)

### 2.1.1 Tipo de Grafos

- **Grafo direcionado** – Um grafo é dito direcionado, ou dígrafo, quando é necessário ser estabelecido um sentido (orientação) para as arestas. O sentido da aresta é indicado através de uma seta, como ilustrado na Figura 3 Nesta situação a aresta passa a ser denominada de arco.
- **Grafo completo** – Se todos os vértices de  $G$  são mutuamente adjacentes o grafo é denominado completo.
- **Grafo ponderado** – trata-se de um grafo ao qual estão associados as aresta um conjunto de pesos  $A$ . Define o conjunto  $A = \{w(i, j)|i, j \in V\}$  onde  $w(i, j)$  e o peso entre a aresta  $i j$ . Um grafo com atributos  $A$ , é definido por  $G = (V, E, A)$ , onde os valores são chamados de atributos. Conforme apresentado na figura 4

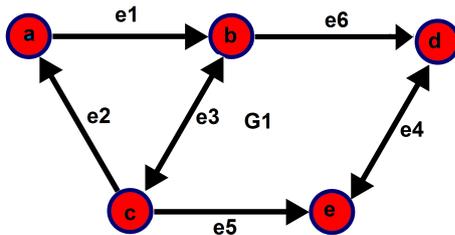


Figura 3 – Grafo Direcionado

**Fonte:** Desenvolvido pelo Autor

Figura 4 – Grafo Ponderado

**Fonte** (GOLDBARG, 2012)

## 2.1.2 Representações de Grafos

Um grafo pode ser representado por diversas formas, a seguir apresentamos as mais relevantes para esta proposta.

- **Listas de arestas** - Trata-se de uma representação de grafos por meio de uma simples lista, ou arranjo. Para representar uma aresta, temos simplesmente um arranjo de dois números de vértice, ou um arranjo de objetos que contém os números dos vértices sobre os quais as arestas são incidentes. Se as arestas tiverem pesos, adiciona-se um terceiro elemento ao arranjo ou mais informações sobre o objeto.
- **Matriz de adjacência** - É uma das formas mais utilizadas para representar grafos. Seja  $A = \{a_{ij}\}$  uma matriz  $n \times n$ , onde  $n$  é o número de vértices de um grafo  $G = (V, E)$  qualquer. A matriz de adjacência  $A$  é construída da seguinte forma:

$$A_{ij} = \begin{cases} 1, & \text{se } i \sim j \\ 0, & \text{caso contrário} \end{cases}$$

Onde, uma aresta é representada por uma "marca" na posição  $(i, j)$  da matriz. A Figura 5 ilustra o conceito de matriz de adjacência para um grafo simples.

- **Listas de adjacência** - É uma representação de grafos baseada na relação de adjacência dos vertices sendo uma combinação das matrizes de adjacência com as listas de arestas. Armazena-se em um arranjo dos vértices adjacentes a cada vértice  $i$ . Temos um arranjo de uma coluna vertical a qual esta vinculada a cada posição  $i$  uma listas de vertices adjacêntes. Na proposta desse trabalho utilizou-se esta estrutura de dados para representar os grafos na implementação.

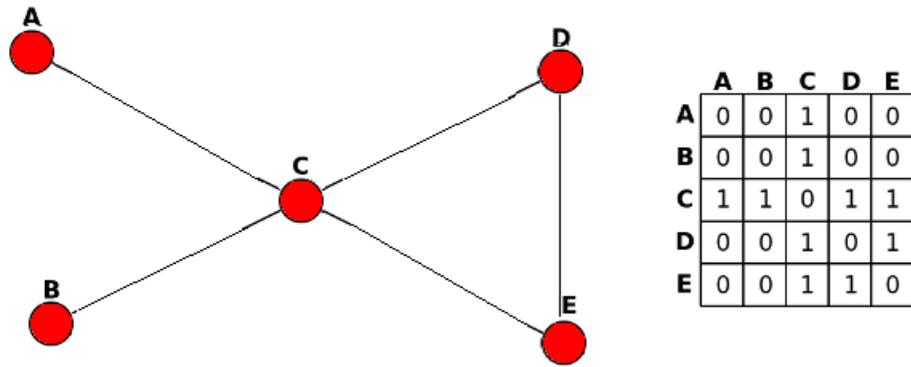


Figura 5 – Matriz de Adjacências

Fonte (GOLDBARG, 2012)

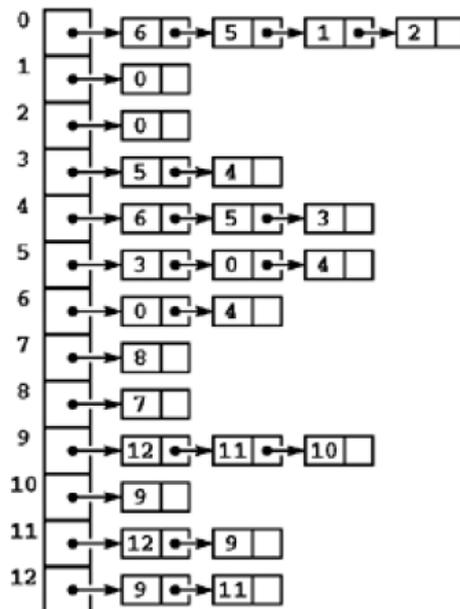


Figura 6 – Lista adjacencia

Fonte (FEOFILOFF; KOHAYAKAWA; WAKABAYASHI, 2011)

## 2.2 Teoria da Complexidade Computacional

A teoria da complexidade computacional é um ramo da ciência da computação que se concentra em classificar problemas computacionais de acordo com sua dificuldade inerente. Neste contexto, um problema computacional é entendido como uma tarefa que a princípio seja solucionável por um computador. Por exemplo, o teste de primalidade é o problema de determinar se um dado número é primo ou não.

A teoria formaliza esta intuição através da introdução de modelos matemáticos de computação para estudar estes problemas e quantificar os recursos necessários para resolvê-los, tais como tempo de processamento e memória utilizada. Um dos papéis da teoria da complexidade computacional é determinar os limites práticos dos computadores.

O tempo de processamento é frequentemente considerado como um dos fatores dominantes para a avaliação de um problema. Desta forma, complexidade de tempo é determinada por uma função definida sobre os tamanhos das instâncias do problema, que considera o tempo necessário para processar soluções para entradas de mesmo tamanho.

### 2.2.1 Comportamento Assintótico de Funções

Para medir o custo de execução do algoritmo é comum definir uma função de complexidade  $f$ , onde  $f(n)$  é a medida do tempo necessário para solucionar um problema de tamanho  $n$ . Se  $f(n)$  é uma medida da quantidade do tempo necessário para executar um algoritmo em um problema de tamanho  $n$ , então  $f$  é chamada função de complexidade de tempo do algoritmo. Uma das formas de se avaliar o tempo de execução de um algoritmo é a análise assintótica, que busca encontrar uma tendência no tempo de execução, quando o volume de dados de entrada do problema tende ao infinito. O comportamento assintótico de  $f(n)$  representa o limite do comportamento do custo quando  $n$  cresce. A análise geralmente avalia algumas operações elementares e, em muitos casos, somente uma operação elementar. A medida de custo ou medida de complexidade relata o crescimento assintótico da operação considerada. As funções de complexidade de tempo são definidas sobre os inteiros não negativos.

**Definição notação  $O$ :** Uma função  $g(n)$  é  $O(f(n))$  se existem duas constantes positivas  $c$  e  $m$  que  $g(n) \leq cf(n)$ , para todo  $n \geq m$ .

**Definição notação  $\Omega$ :** Uma função  $g(n)$  é  $\Omega(f(n))$  se existem duas constantes positivas  $c$  e  $m$  que  $g(n) \geq cf(n)$ , para todo  $n \geq m$ .

**Definição notação  $\Theta$ :** Uma função  $g(n)$  é  $\Theta(f(n))$  se existirem constantes positivas  $c_1$ ,  $c_2$  e  $m$  tais que  $0 \leq c_1f(n) \leq g(n) \leq c_2f(n)$ , para todo  $n \geq m$ .

As principais classes de problemas possuem as funções de complexidade descritas abaixo:

1.  $f(n) = O(1)$ . Algoritmos de complexidade  $O(1)$  são ditos de **complexidade constante**. O uso do algoritmo independe do tamanho de  $n$ . Neste caso, as instruções do algoritmo são executadas um número fixo de vezes.
2.  $f(n) = O(\log n)$ . Um algoritmo de complexidade  $O(\log n)$  é dito ter **complexidade logarítmica**. Este tempo de execução ocorre tipicamente em algoritmos que resolvem um problema transformando-o em problemas menores. Nestes casos, o tempo de execução pode ser considerado como menor do que uma constante grande. Quando  $n$  é mil e a base do logaritmo é 2,  $\log_2 n \approx 10$ , quando  $n$  é um milhão,  $\log_2 n \approx 20$ . Para dobrar o valor de  $\log n$  temos que considerar o quadrado de  $n$ . A base do logaritmo muda pouco estes valores: quando  $n$  é um milhão, o  $\log_2 n$  é 20 e o  $\log_{10} n$  é 6.

3.  $f(n) = O(n)$ . Um algoritmo de complexidade  $O(n)$  é dito ter **complexidade linear**. Em geral, um pequeno trabalho é realizado sobre cada elemento de entrada. Esta é a melhor situação possível para um algoritmo que tem de processar  $n$  elementos de entrada ou produzir  $n$  elementos de saída. Cada vez que  $n$  dobra de tamanho, o tempo de execução dobra.
4.  $f(n) = O(n \log n)$ . Este tempo de execução ocorre tipicamente em algoritmos que resolvem separadamente e depois ajuntando as soluções (algoritmos com essa característica obedecem ao paradigma de dividir para conquistar). Quando  $n$  é 1 milhão e a base do logaritmo é 2,  $n \log_2 n$  é cerca de 20 milhões. Quando  $n$  é 2 milhões,  $n \log_2 n$  é cerca de 42 milhões, pouco mais do que o dobro.
5.  $f(n) = O(n^2)$ . Um algoritmo de complexidade  $O(n^2)$  é dito ter **complexidade quadrática**. Algoritmos desta ordem de complexidade ocorrem quando os itens de dados são processados aos pares, muitas vezes em um laço dentro de outro. Quando  $n$  é mil, o número de operações é da ordem de 1 milhão. Sempre que  $n$  dobra, o tempo de execução é multiplicado por 4. Algoritmos deste tipo são úteis para resolver problemas de tamanho relativamente pequenos.
6.  $f(n) = O(n^3)$ . Um algoritmo de complexidade  $O(n^3)$  é dito ter **complexidade cúbica**. Algoritmos desta ordem de complexidade são úteis apenas para resolver pequenos problemas. Quando  $n$  é 100, o número de operações é da ordem de 1 milhão. Sempre que  $n$  dobra, o tempo de execução fica multiplicado por 8.
7.  $f(n) = O(2^n)$ . Um algoritmo de complexidade  $O(2^n)$  é dito ter **complexidade exponencial**. Algoritmos desta ordem de complexidade geralmente não são úteis sob o ponto de vista prático. Eles ocorrem na solução de problemas quando se usa a força bruta para resolvê-los. Quando  $n$  é 20, o tempo de execução é cerca de 1 milhão. Quando  $n$  dobra, o tempo de execução fica elevado ao quadrado.
8.  $f(n) = O(n!)$ . Um algoritmo de complexidade  $O(n!)$  é também dito ter **complexidade exponencial**, apesar de que a **complexidade fatorial**  $O(n!)$  tem comportamento muito pior do que a complexidade  $O(2^n)$ . Algoritmos desta ordem de complexidade geralmente ocorrem na solução de problemas quando se usa força bruta para resolvê-los. Quando  $n$  é 20,  $20! = 2432902008176640000$ , um número com 19 dígitos.

Numerosos problemas são considerados *intratáveis* no sentido da não existência de algoritmos eficientes capazes de resolvê-los otimamente. Dentre estes, podemos exemplificar os de natureza combinatória, aos quais a prova da otimalidade, muitas vezes, somente é possível pela enumeração de todas as possíveis soluções. Para um dado problema

combinatório, o tempo de computacional, para a enumeração das soluções viáveis, cresce abruptamente em detrimento ao tamanho da entrada dos dados que o define. Garey e Johnson (1979) classificam da seguinte forma os problemas, segundo sua complexidade:

- **Indecidível:** São problemas tão complexos que nenhum algoritmo pode resolvê-los.
- **Intratável:** Problemas decidíveis porém difíceis, para os quais possivelmente não existe algoritmo que os resolvam em tempo polinomial mas, pode existir em tempo exponencial;
- **Tratável:** Problemas para os quais existe algoritmo que os resolvam em tempo polinomial.

Este trabalho apresenta uma heurística para a geração de rotas com balanceamento de carga que pode ser conceituado como um problema tratável. As classificações também podem considerar a resposta esperada:

- **Problemas de decisão:** Consiste na verificação da veracidade ou não de determinada questão do problema tendo como retorno duas possibilidades (SIM ou NAO).
- **Problemas de localização:** Consiste na verificação da existência e localização de uma solução segundo algum critério para o problema.
- **Problemas de otimização:** Consiste na verificação da existência e identificação da melhor solução possível, dentre as soluções factíveis para o problema.

Complementado a classificação dos problemas avaliaremos segundo o tempo necessário encontrar uma solução e para verificar se uma resposta fornecida é realmente uma solução para o problema. As classes mais estudadas nesses dois quesitos, respectivamente, são denominadas  $P$  e  $NP$ . A classe  $P$  consiste nos problemas que podem ser resolvidos em tempo polinomial por algum algoritmo determinístico. Alguns exemplos: Ordenação, caminho mínimo em grafos e fluxo máximo em redes. Observe que se os algoritmos conhecidos de um dado problema  $A$  são todos exponenciais, não necessariamente  $A \notin P$ .

## 2.3 Problema do Caminho Mínimo

O problema de caminhos mínimos (*Shortest-path problem-SP*) é considerado uma das questões fundamentais da computação e consiste, em resumo, em determinar o caminho de menor custo em um grafo, partindo de um vértice origem até um vértice destino. Sendo objeto de estudo desde a década de 50, a inclusão deste tema deve-se ao fato que a heurística proposta por este trabalho, utiliza um dos principais algoritmos desenvolvidos

para a soluções deste problema. Na literatura existe grande quantidade de publicações a respeito do tema e inclusive são conhecidas várias soluções eficientes (GOLDBARG, 2012).

Na indústria de telecomunicações e de transportes geralmente quando há necessidade de enviar dados ou mecadorias entre dois locais geograficamente separados de forma a otimizar os recursos, o problema do caminho mais curto surge como alternativa de abordagem, senão como modelo único, como solução de subproblemas de problemas mais complexos.

Uchoa (2012) propõe a seguinte modelagem para o problema: seja  $G = (V, A)$  um grafo direcionado no qual  $V$  é um conjunto de vértices e  $A$  um conjunto de arestas  $(i, j)$  e associa-se a cada aresta um peso  $c_{ij}$ .

Define-se uma **função de custo**  $c$  em  $G = (V, A)$  sendo uma função sobre  $A$ , onde,  $\forall wu \in A$ ,  $c(wu)$  é o valor do  $c$  em  $wu$  (custo da aresta  $wu$ ). Considerando os vertices  $(s, t) \in N$ , como  $s$  origem e  $t$  destino, e seja  $D$  um caminho entre  $s$  e  $t$  respectivamente pode-se definir o **custo do caminho**  $D$  da seguinte forma:

$$c(D) = \sum_{wu \in D} C(wu) \tag{2.1}$$

Entretanto, somente podemos afirmar que  $D$  é o caminho de custo mínimo se,  $c(D) \leq c(P)$  para todo  $P$  que começa em  $s$  e termina em  $t$ .

Delibera-se a **distância** entre os vértice  $s$  e  $t$  como sendo o **menor caminho** e simboliza-se por  $dist_{(s,t)}$ . Observa-se a distância  $dist_{(s,t)}$ , distacada pelo traçado em azul na figura 7 do grafo(a) somando  $dist_{(s,t)} = 17$ . O **maior caminho** entre os vértices  $s$  e  $t$  denomina-se por  $C = \{max c(st) \setminus st \in A\}$  distacado em traçado vermelho na figura 7, grafo(b), com  $C=33$ .

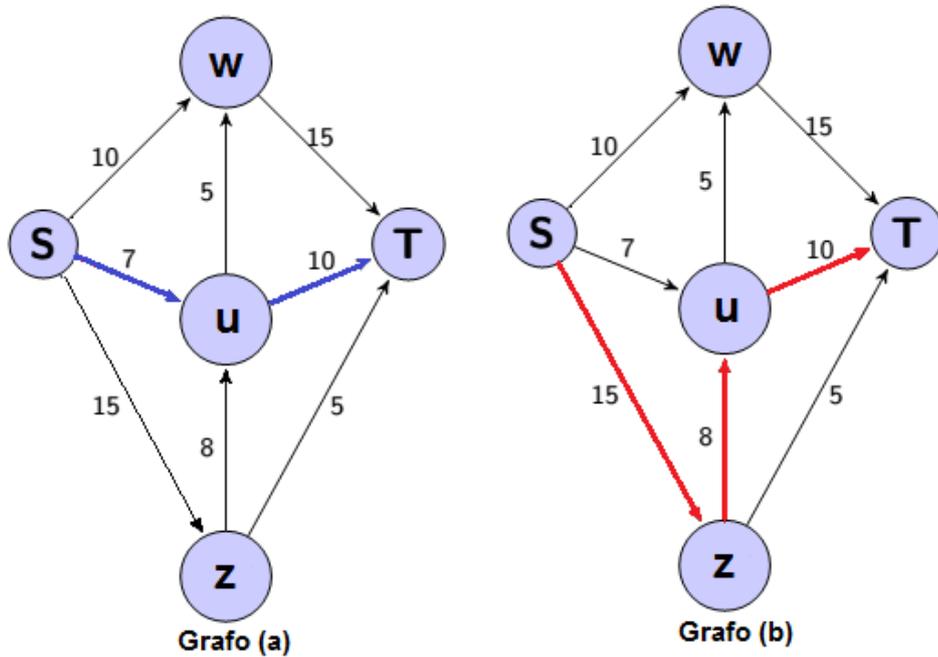


Figura 7 – Grafo-Caminho Mínimo e Máximo

Fonte:(??)

O caminho entre quaisquer dois vértices distintos de  $G$  equivale diretamente a uma rota possível entre os respectivos vértices. Dessa forma, podemos concluir que, se selecionarmos a rota no grafo que é equivalente ao caminho mínimo entre  $(s,t)$  exibido, estaremos certamente transportando a carga desejada pela rota de menor distância disponível. Pode-se formular este problema da seguinte forma:

$$\text{Minimizar } Z = \sum_{i \in N} \sum_{j \in N} C_{ij} X_{ij} \quad (2.2)$$

Sujeito a

$$\sum_{j \in N} X_{sj} = 1 \quad (2.3)$$

$$\sum_{i \in N} X_{ij} - \sum_{k \in N} X_{ik} = 0 \quad \forall j \in N - s, t \quad (2.4)$$

$$\sum_{i \in N} X_{it} = 1 \quad \forall j \in N - s, t \quad (2.5)$$

em que,  $X$  é um vetor de decisão do problema.

$$x_{ij} = \begin{cases} 1, & \text{se pertence ao caminho } D \\ 0, & \text{se não pertence ao caminho } D \end{cases}$$

Algumas observações relacionadas com este tipo de problemas:

- O comprimento de um caminho é maior do que o de qualquer dos seus subcaminhos;
- Qualquer subcaminho de um caminho mais curto, é ele próprio um caminho mais curto (princípio da optimalidade);
- Para uma rede com  $n$  nós, qualquer caminho mais curto tem no máximo  $n-1$  arcos.

O Problema do Caminho Mínimo é de natureza combinatória e tem foco na otimização, logo é aplicável em diversas áreas do conhecimento. Entretanto, na proposta clássica do problema são associados custos fixos às aresta  $(i, j) \in A$  e desta forma, a modelagem de problemas reais, que apresentam dados dinâmicos e algum grau de incertaza, é pouco eficaz fazendo-se necessário outras abordagens. Um exemplo prático desta limitação é observado quando pretende-se modelar o **Problemas de Roteamento de Veículos**-(VRP) como um problema do **SP** considerando os custos fixos. Ainda que em muitos casos consiga-se definir a menor distância entre o início o destino do trajeto de um veículo, fatores como o trânsito teriam impacto sobre a solução considerada ótima para o problema, uma vez que o menor caminho não garante o menor tempo de viagem. Soluções para determinar caminhos mínimos em espaços dinâmicos são desejados, na medida em que a robótica exige cada vez mais bons algoritmos para a navegação. (DENG et al., 2012)

### 2.3.1 Aplicações

O Problema de Caminhos Mínimos pode ser aplicado em uma imensa quantidade de problemas práticos. Esta seção descrever algumas destas aplicações.

#### 1. Roteamento de tráfego de veículos

O cenário atual do tráfego de veículos é um vasto campo de aplicação do **Problemas Caminho Mínimo**, atualmente existe disponível ao público diversos aplicativos de suporte aos motoristas para a definição de melhores rotas, por exemplo os bem conhecidos **Google Maps** e **Waze**. A necessidade de deslocar-se de um ponto a outro em uma rede de tráfego veicular é natural e envolver uma série de fatores na escolha da rota para o trajeto. Deseja-se percorrer o melhor caminho possível e a seleção avaliar um conjunto de variáveis, tais como, gasto com combustível, pedágio, segurança e congestionamentos etc. É válido ressaltar que a ponderação dos peso das variáveis e as restrições aplicadas ao processo de seleção da rota é particular a cada aplicação ou usuário do sistema.(GOMES; VALENTE; FREITAS, )

## 2. Roteamento de dados baseado em Qualidade de Serviço-QoS

A qualidade de serviço (**QoS – quality of service**) é um aspecto importante para as redes de pacotes como um todo. Sendo um aspecto fundamental avaliação do desempenho de determinadas aplicações fim-a-fim, como vídeo e áudio conferência e transferência de dados. Geralmente, o problema da escolha do caminho é formulado como um **Problema do Caminho Mínimo** com uma única função objectivo com uma métrica ou várias métricas diferentes, cujos requisitos de QoS são incorporados nestes modelos matemáticos, através da utilização de restrições adicionais.(UCHOA, 2012)

### 2.3.2 Caminho Mínimo com Peso Limitado

O problema do Caminho Mínimo com Peso Limitado é uma variação bastante conhecida e estudada do Problema do caminho minimo. Foi proposta por (JOKSCH, 1966) ao contrário do problema original, é classificado como NP-HARD, e tem aplicações, por exemplo, em gerência de linhas férreas e em sistemas de controle aéreo militar. Nesta abordagem, a cada aresta  $(i, j)$  além da distância é associada o tempo de deslocamento. Assim, busca-se encontrar a menor rota que não utrapasse o tempo máximo permitido para a viagem. Assim, problema para o qual avalia-se a viabilidade utilizando da abordagem do Caminho Mínimo com Peso Limitado, objetiva determinar se existe solução para uma dada instância do problema são de orden polinomial, já que é suficiente determinar um caminho de peso mínimo em um grafo. Existem para este problemas métodos de resolução exatos, aproximativos e heurísticos disponíveis na literatura. Os mais relevantes para esta abordagem são os exatos, os quais, baseam-se no uso do k-menores caminhos, programação dinâmica, rotulação de vértices (DIJKSTRA, 1959) e de relaxação lagrangeana em otimização linear. (GOLDBARG, 2012)

### 2.3.3 Caminho Mínimo com Recursos Limitados

Trata-se de uma generalização do Caminho Mínimo com Peso Limitado, onde são associados vários recursos a cada aresta  $(ij)$  formando um conjunto de k-pesos associado a cada aresta do grafo. Assim, a avaliação considera um conjunto limitado de recursos das arestas. Seja  $K$ , o conjunto de recursos associados às arestas de  $G$ , sendo  $G$  um grafo simétrico,  $d_{wu} = d_{uw}$  O problema Caminho Mínimo com Recursos Limitados consiste em encontrar o caminho limitado pelos recursos de menor custo possível. (HANDLER; ZANG, 1980) e (JAFFE, 1984) demonstram que o problema é NP-HARD mesmo em grafos acíclicos, com restrições sobre um único recurso, e com todos os consumos de recursos positivos. Existe a possibilidade de redução do tamanho problema pela eliminação de vértices ou arcos que não podem fazer parte de uma solução ótima. Assim, uma possibilidade é a redução baseada nos recursos, na qual são removidos os arcos que não podem pertencer a

qualquer caminho viável. Outra possibilidade é caso exista um limite superior conhecido para uma solução ótima do problema, realizar a redução baseada em custos que parte do mesmo princípio retira os arcos ou vértices que ultratassam o limite. (UCHOA, 2012)

### 2.3.4 K-Caminho Mínimos

O problema do K-Caminhos Mínimos assemelha-se ao problema do K-Ésimo Menor Elemento, sendo o **Problema k-ÉSIMO(S,k)** definido da seguinte forma: Dado um conjunto  $S$  de números inteiros e um  $k$  número inteiro positivo, objetiva-se encontrar  $k$ -ÉSIMO elementos de  $S$ . Adequando o problema para **K-Caminhos Mínimos** apresenta-se as seguintes definições: Suponha que  $G(V, A)$  seja um grafo,  $c$  uma função de custo e  $(s,t)$  vértices pertencente a  $G$ . Considere um conjunto  $P_{st}$  de todos os caminhos distintos e possíveis entre  $s$  e  $t$ , como solução do problema, espera-se como retorno uma lista com **k-K-Caminhos Mínimos** de forma crescente:

$$c(P_1) \leq c(P_2) \leq \dots \leq c(P_K) \leq \min\{c(P) : P \in P_{st} - \{P_1, \dots, P_K\}\}$$

O algoritmo proposto por Yen (1971) tem um papel central nas soluções para **Problema do k-Menores Caminhos** apresentado resultados eficientes. É aplicável a grafos com pesos negativos sem circuitos negativos, para tal aplicação, utiliza-se o algoritmo de **Ford** no lugar do algoritmo de Dijkstra (1959). O algoritmo de Yen foi posteriormente melhorado e adequado por Lawler (1972) para problemas de **otimização combinatória** sendo utilizando como subrotina para a determinar a solução ótima diante de algumas restrições. Podemos referenciar algumas propostas relevantes para a solução do **K-Caminho Mínimos**. Bock, Kantner e Haynes (1957) propõem um método Genérico no qual ele enumera os caminhos mínimos e posteriormente ordena de acordo com o custo entrega no final uma lista dos K-menores caminhos. Clarke, Krikorian e Rausen (1963) introduz o procedimento *branch-and-bound* localizando os caminhos e enumera os que desviavam do Caminho Mínimo em algum momento.

### 2.3.5 O Problema do Caminho Mínimo com Incertezas

O problema do caminho mínimo está associado a várias aplicações nas áreas de roteamento, manufatura, fluxo em redes, dentre outras. Seu principal objetivo é minimizar os esforços de busca, sendo  $u$  e  $v$  dois vértices de um grafo  $G = (V, A)$ , o caminho mínimo entre  $u$  e  $v$  é uma sequência de arcos que, passando por nós distintos, une  $u$  a  $v$  de forma a acumular o menor distância. Para que haja um caminho mínimo entre dois arcos  $u$  e  $v$  é necessário que exista uma conexão entre eles, isto é,  $v$  seja sucessor de  $u$ . (GOLDBARG, 2012)

Na maioria dos casos, existe mais de um caminho entre dois vértices específicos; com isso, o problema do caminho mínimo consiste em encontrar o caminho com menor

custo dentre todos os possíveis. Em Geral na teoria dos grafos, os peso (custo) de cada arco é um número real, existem algoritmos eficientes para resolver este problema. Porém, em muitas aplicações reais estes parâmetros (custos, capacidades, dentre outros) não precisos, vezes, por falta de informações.

Hernandes, Berton e Castanho (2009) propõem um o algoritmo que aborda incertezas nos arcos, além de incluir restrições de tempo nos nós. sendo composto de duas fases: na primeira é realizada conjunto solução de caminhos não-dominados; na segunda é feita a distribuição do fluxo ao longo dos arcos dos caminhos não-dominados.

### 2.3.6 O Problema do Caminho Mínimo Robusto

O Problema de Caminho Mínimo Robusto é estudado desde do fim da década de 60 tendo aplicações o auxílio à tomada de decisões quando o conhecimento sobre o estado atual de um problema não pode ser determinado. O Caminho mais Mínimo Robusto (**RSP, do inglês Robust Shortest Path Problem**) é uma generalização do **Problema do Caminho Mínimo**, onde o custo de cada arco  $(i, j) \in A$  é definido por um intervalo  $[L_{ij}, U_{ij}]$ , com  $L_{ij}, U_{ij} \in N$ , onde  $U_{ij} \geq L_{ij} \geq 0, \forall (i, j) \in A$ .

Seja  $P \subset A$  um caminho que parte de um vértice de origem  $s$  até um vértice de destino  $t$  em  $G$ . O arrependimento de  $P$  no cenário  $r$ , também chamado de desvio robusto de  $P$  em  $r$ , é definido como a diferença entre o custo de  $P$  no cenário  $r$  e o custo do caminho mais curto  $S^r$  entre  $s$  e  $t$  em  $r$ . Ou seja, o desvio robusto de  $P$  em  $r$  é o custo extra pago na utilização de  $P$  ao invés de  $S^r$ , caso o cenário  $r$  ocorra. O custo robusto de  $P$  é o maior desvio robusto de  $P$  sobre todos os cenários. Portanto, o Problema do Caminho Mínimo Robusto consiste em encontrar o caminho  $P^*$  de  $s$  para  $t$  com o menor custo robusto. Este problema é NP-HARD mesmo para grafos acíclicos. (COCO; NORONHA; SANTOS, 2014)

As abordagem apresentadas anteriormente propõem algumas modelagens para o Problema do Caminho Mínimo cada uma considerando algumas particularidades do mesmo problema e resultando em varias possibilidades de aplicações distintas. Tem-se como base comun a utilização da teoria de grafos na modelagem das soluções. A proposta desse trabalho esta diretamente relacionada com o objeto de estudo das abordagem apresentadas entretanto, na modelagem proposta, avalia-se a possibilidade da inclusão do preço como principal variável a ser considerada na seleção dos caminhos e avarigua a possibilidade que esta inclusão simplifique a modelagem para problemas complexos sem a perda da aplicabilidade das soluções. A aplicação do preço esta vinculada a um ambiente dinâmico assim, como no problema k-caminho minimos temos ao final uma lista de k-caminhos mínimos possíveis como no recurso limitados temos uma forma de balanciar a utilização dos caminhos sem a necessidade de restigirtrechos do grafo.

### 2.3.7 Algoritmos para Solução do Problema do Caminho Mínimo

Existem vários algoritmos eficientes para resolução problemas de caminho mínimo, os mais conhecidos os algoritmos são os de **Dijkstra**, **Ford** e de **Floyd-Warshall**. Os Algoritmo de Dijkstra e Ford aplicam-se a problemas de caminho mais mínimo quando necessita-se determinar o menor caminho entre um vertice inicia  $s$ , para um vértice final,  $t$  ou para todos os outros vertices, e baseiam-se no processo de rotulação<sup>1</sup>. O algoritmo de Floyd aplica-se a problemas de caminho mais curto entre todos os pares de vertices.

### 2.3.8 Algoritmo de Dijkstra

Conforme exposto, existem na literatura diversas propostas de resolução para o **SP**, dentre várias alternativas optou-se por utilizar o algoritmo Dijkstra principalmente por ser um algoritmo tradicional que apresenta ótimos resultados.

Proposto pelo holandês Edsger Wybe Dijkstra em 1959, (DIJKSTRA, 1959) trata-se de um algoritmo de rotulação para caminhos em grafos com arestas com peso positivos, com tempo computacional  $O([m + n] \log n)$  onde  $m$  é o número de arestas e  $n$  é o número de vértices. Algoritmo capaz de resolver problema *Shortest Path* encontrando o menor caminho entre dois vértices em um grafo ponderado. Por exemplo, determinar a menor distância para uma viagem que atravessa várias cidades com diferentes caminhos para se chegar a um destino. Dado um vértice de início  $s$ , localiza o menor caminho a partir de  $s$  para todos os outros vértices do grafo, inclusive o destino  $t$  (GOLDBARG, 2012)

O algoritmo utiliza-se da indução e ajuste, sendo uma ferramenta eficiente e de fácil implementação computacional que usa uma fórmula de recorrência descrita por:

$$d(x)^i = \min\{d(x)^{i-1}, d(y) + d(x, y)\} \quad (2.6)$$

Onde:

$d(x)^i$  =Comprimento do caminho da origems  $s$  até o vertice  $x$  na interação  $i$ ;

$i$  =índice do vértice a ser fechado na interação  $t$ ;

$d(y)$  =Comprimento do caminho da origem  $s$  até o vértice fechado  $y$ ;

$d(y, x)$  =Comprimento da aresta  $(y, x)$ .

---

<sup>1</sup> 1. Ação ou resultado de rotular(-se);

2. Fixação de rótulo, etiqueta etc. em um objeto ou produto, a fim de divulgar informações importantes sobre o mesmo;

3. Atribuição de qualidades negativas ou positivas sobre alguém ou algo, ger. com simplismo e inadequação

Uma característica importante do algoritmo se deve a não necessidade de uma busca exaustiva em todos os caminhos, característica que permite um aumento da eficiência da busca.

O funcionamento do algoritmo de dijkstra da-se por meio do processo de rotulação dos vértices como **fechados** ou **aberto**, quando o caminho mínimo da origem  $s$  até o vertice é conhecido rotula-se este vértice como fechado. Entretanto, quando ainda é indeterminado o caminho mínimo entre o vértice avaliado e a origem  $s$  rotula-se como aberto. A perspectiva da rotulação dos vertices está associada à possibilidade ou não de se localizar um caminho menor que o atual.

Larson e Odoni (1981) propõem a seguinte estruturação para o algoritmo:

**1ºPasso:** Atribui-se um valor  $d(x)$  para cada um dos vértices do grafo sendo:  $d(s) = 0$  e  $d(x) = \infty$  para todo vértice  $x \neq s$  considere  $y$  o último vértice rotulado como fechado. Inicialmente o vértice  $s$  é o único rotulado e  $y = s$ .

**2ºPasso:** Para cada vértice  $x$  aberto redefine-se  $d(x)$  conforme a expressão  $d(x)^i = \min\{d(x)^{i-1}, d(y) + d(x, y)\}$ . O vértice aberto que possuir o menor valor  $d(x)$  será fechado e se faz  $y = x$

**3ºPasso:** Se o vértice de destino  $t$  foi rotulado como fechado então o algoritmo para, um caminho de  $s$  para  $t$  foi encontrado. Se o  $t$  ainda não foi fechado retorna ao segundo passo.

A Figura 8 mostra a evolução do algoritmo quando aplicado em um grafo na busca do menor caminho entre  $s$  e  $t$ . O rótulo colocaliza-se próximo de cada vértice. Na do primeira posição do rótulo guarda-se a origem da rotulação e a segunda posição anota-se o valor do caminho acumulado até o vértice. A cada iteração  $t$  o vértice examinado que acumula a menor distância é fechado. Nota-se que ao final de cinco interações o caminho

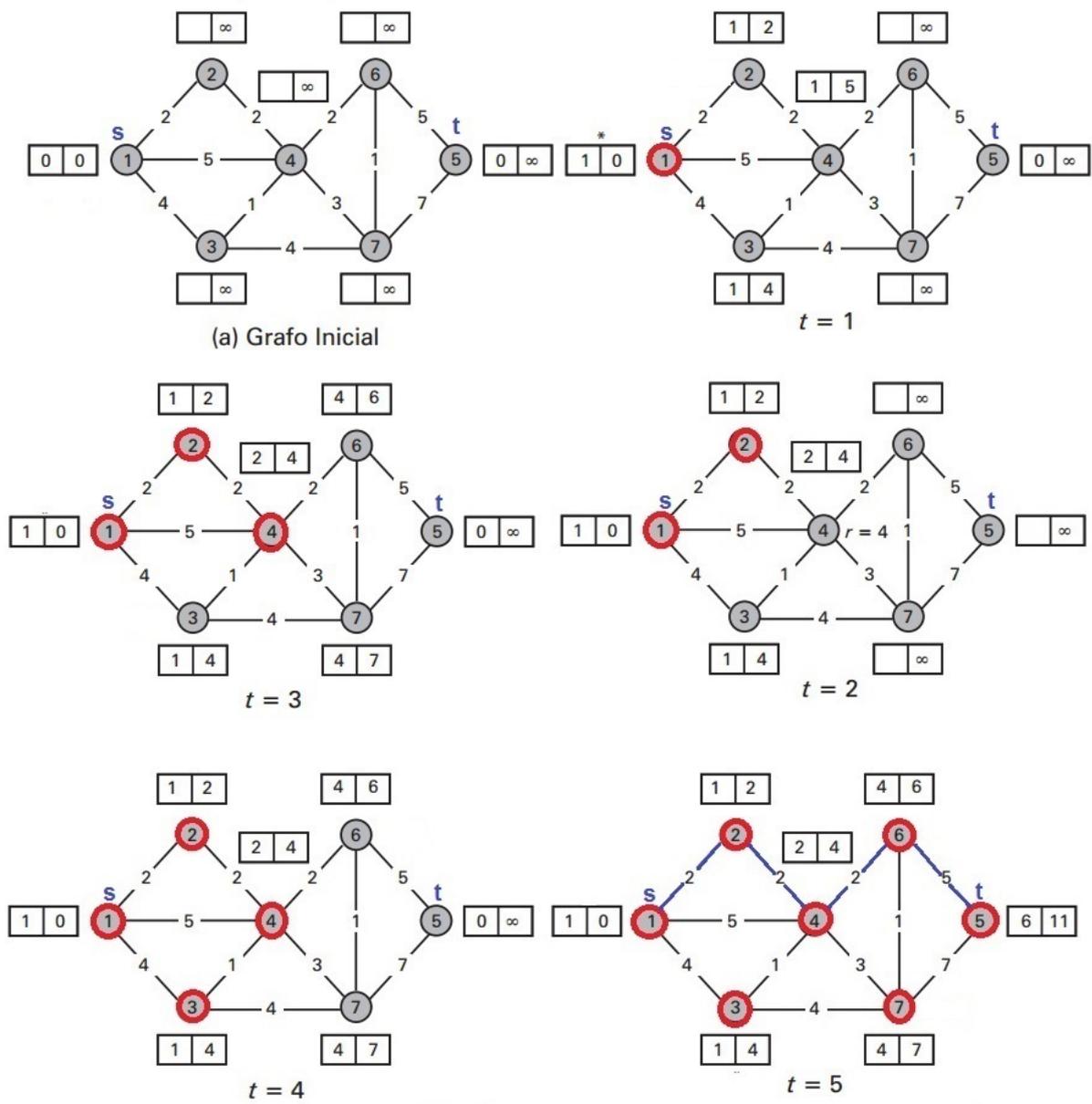


Figura 8 – Interações do Dijkstra  
 Fonte (GOLDBARG, 2012) Adaptada.

minimo entre  $s$  e  $t$  foi localizado pelo algoritmo sendo compostos pelos vértices 1,2,4,6 e 5. Quando o vértice 1 é analisado na interação 1 o vértice adjacente 4 é rotulado com a distância 5 entretando, na próxima interação na avaliação do vértice 2 a distância para o vertice 4 é reavaliada e modificada para 4. Desta forma, o caminho minimo da origem para todas as vértices são determinados.

## 2.4 Problema do Caixeiro Viajante (TSP)

A literatura especializada considera o Problema do Caixeiro Viajante (Traveling Salesman Problem — TSP) o primeiro problema de roteirização, datado do século XVIII, foi proposto pelo matemático Irlandês William Rowan Hamilton e pelo matemático britânico Thomas Penyngton Kirkman (MATAI; SINGH; MITTAL, 2010). Em síntese, o problema consiste em definir um roteiro ou sequência de cidades a serem visitadas por um caixeiro viajante de forma que o mesmo retorne à cidade de origem e visite cada cidade apenas uma vez, tendo como norteador a busca pelo trajeto mínimo. Sendo considerado um dos problemas clássicos da programação matemática, em geral os problemas de roteirização determinam sequência de pontos a serem visitados para atender uma demanda. Na modelagem de problemas reais os pontos a serem visitados podem representar cidades, postos de trabalho ou atendimento, depósitos etc. (RUSIC, 2015)

O TSP pertence à classe de problemas conhecido por NP-Hard, problemas para os quais ainda não existe algoritmo em tempo polinomial capaz de resolvê-los. Apenas instâncias com poucos pontos podem ser solucionados de forma ótima, situações com grande número pontos são intratáveis com métodos exatos, em virtude do esforço computacional necessário para resolvê-los. Muitas abordagens de algoritmos heurísticos e metaheurísticos fornecem soluções factíveis, próxima da ótima. Uma abordagem geralmente utilizada é modelagem do problema por meio da teoria de grafos, sem perda de generalização, um grafo completo e ponderado  $G = (V, A)$ , onde  $V$  é conjunto de vértices associadas as cidades que deseja-se visitar e  $A$  o conjunto de arestas para as quas são associados os custos do deslocamento. Inicialmente consideraremos cada aresta  $(i, j)$  com apenas o valor  $d_{ij}$  que representa a distância entre as cidades  $i$  e  $j$ . O objetivo é encontrar o menor caminho Harniltoniano no grafo, caminho este fechado que visita cada vértice do grafo uma única vez. Mediante a sua aplicabilidade existe na literatura especializada diversas generalizações do TSP. A elaboração de novas generalizações são realizadas pela incorporação de novas restrições e objetivos ao problema original, no intuito de encontrar soluções envolvendo diferentes tipos variáveis e aplicáveis à problemas reais, tais como roteiro de viagem, roteirização de veículos de cargas, atendimento de clientes com tempo determinado para entrega etc (CACCALANO; CUNHA, 2015). Entre as generalizações do TSP presentes na literatura, vamos citar as mais relevantes:

- Symmetric Clustered Traveling Salesman Problem TSP-Simétrico Cluster;
- Traveling Salesman Problem Backhauls (TSP-Backhauls)
- Prize-Collecting Traveling Salesman Problem (TSP-PC)
- Multiple Traveling Salesman Problem (mTSP)
- Traveling Salesman Problem With Window Time (TSP-WT)

### 2.4.1 Aplicação do Traveling Salesman Problem

Conforme exposto anteriormente, a origem do TSP é datada do século XVIII, entretanto, apresenta em suas diversas versões aplicabilidade, estando presente na formulação de soluções para inúmeros problemas práticos, dos quais destacamos:

Tabela 1 – Aplicações TSP. Fonte: (GOLDBARG, 2012)

<b>Aplicações</b>	<b>Autor</b>
Determinação de acesso a arquivos computacionais	Henry-Labordere (1969).
Programação de assistência social	Saskena (1970).
Manipulação de itens em estoque	Ratliff e Rosenthal (1981) .
Programação de transporte entre células de manufatura	Finke e Kusiak (1985).
Programação de operações de máquinas em manufatura	Kusiak e Finke (1987).
Otimização do movimento de ferramentas de corte	Chauny et al. (1987).
Otimização de palhetas em turbinas a gás	pante et al. (1987).
Análise de estruturas cristalinas	Bland e Shallcross (1987).
Fabricação de Chips VLSI	Korte (1988).
perfuração de furos em placas de circuitos impressos	Reinelt (1989).
sequenciamento de tarefas	Whitley (1991).
Trabalhos administrativos	Laporte et al. (1996).
Roteamento de entrega postal	Laporte et al. (1996).
programação e distribuição de tarefas em plantas	Salomon et al.(1997).

## 2.5 Problema de Roteirização Veicular (VRP)

Admite-se que Dantzig e Ramser (1959) sejam os primeiros autores a formular o problema da roteirização de veículos (*Vehicle Routing Problem-VRP*), para uma aplicação real de distribuição de combustível a postos de venda. Os autores se referem ao VRP com uma particularização do TSP, para o qual foi considerando as demandas nos postos de vendas e a capacidade dos veículos utilizados (CACCALANO; CUNHA, 2015). Cinco anos depois (CLARKE; WRIGHT, 1964) generalizou este problema para um problema de Otimização Linear comumente encontrado no domínio da logística e transporte. De forma simplificada o VRP pode ser compreendido como o procedimento para a elaboração de um plano de atendimento a um conjunto de clientes dispersos geograficamente com demandas conhecidas ou estimadas, partindo de um depósito central ou vários depósitos, de modo que as demandas dos clientes sejam atendidas. Por tratar-se de uma generalização do TSP, o VRP é classificado como um problema NP-Hard. O objetivo mais comum é minimizar a soma dos custos dos trajetos dos veículos. De acordo com (EKSIOGLU; VURAL; REISMAN, 2009), a literatura VRP tem crescido exponencialmente a uma taxa de 6 a cada ano. Essa popularidade torna difícil acompanhar os desenvolvimentos na área e ter uma visão clara, pois, extensões e métodos de solução são relativamente novos. Assim, o número de variáveis, diversidade de restrições e objetivos presentes nos problemas

reais impõem aos pesquisadores a necessidade da realização de um exame cuidadoso da taxonomia dos VRP durante o processo de modelagem das abordagens.(JAEGERE; DEFRAEYE; NIEUWENHUYSE, 2014) Apresentam a mais recente taxonomia acerca do tema, sendo uma versão adaptada da taxonomia proposta por (EKSIIOGLU; VURAL; REISMAN, 2009) assinalam cinco tópicos principais:

- Tipo de estudo;
- Características do Cenário;
- Problema Físico;
- Características de Informação;
- Características dos Dados,

### 2.5.1 Aplicações Problema Roteirização de Veículos

O Problema de roteirização tem inúmeras aplicações práticas, principalmente na indústria, no comércio, no setor de serviços, na segurança e setor de transporte destacamos na Tabela 2 algumas aplicações:

Tabela 2 – Aplicações *VRP*. Fonte: (GOLDBARG, 2012)

<b>Aplicação</b>	<b>Autor</b>
Distribuição de manufaturados	Perl, Daskin (1985)
Distribuição de produtos diversos	Dantzig, Ramser (1959), Klots et al. (1992)
Distribuição de bebidas	Golden, Wasil (1987); Eibl et al. (1994)
Transporte escolar	Newton, Thomas (1974), Li, Fu (2002)
Entrega de correspondência	Frederickson (1978)
Distribuição de pão	Derigs, Grabenbauer (1993)
Roteamento de helicópteros	Timlin, Pulleyblank (1990).
Transportes coletivos urbanos	Ceder e Stern (1981).
Serviços de emergência	Daskin (1987).
Distribuição de derivados de petróleo	Ronen et al. (1987), Brown e Graves (1981).
Distribuição de gás	Bell et al. (1983).
Roteamento de células de manufatura flexível	Finke e Kusiak (1985).
Entrega de correspondência bancária	Malmberg e Simons (1989).
Patrulhamento policial e de segurança	Larson (1972), Calvo e Cordone (2003).
Roteamento de navios, cabotagem e logística	Brown et al. (1987), Ronen (2002).
Roteamento de satélites	Lee et al. (2003).
Just-in-time	Vaidyanathan et al. (1999).

As seguintes restrições já alteram a forma clássica do problema de VRP como originalmente modelado, estas servem para aproximar o problema da realidade e por conseguinte tornando-o mais complexo de resolver.

## 2.6 Máquina de Estados

Máquina de estados finita *FSM-Finite State Machine* ou autômato finito é um modelo matemático usado para representar programas de computadores ou circuitos lógicos que apresenta um conjunto de estados finito. Tendo como característica fundamental que sua memória é limitada e exclusivamente organizada em torno do conceito de “estado”. O FSM pode mudar de um estado para outro em resposta a algumas entradas tal, mudança de estado é denominada transição. Podemos defini-lá por uma lista de estados, ações e condições para transições entre os estados. Uma aplicação significativa desse padrão de desenvolvimento de software está presente em jogos principalmente na parte de seleção do comportamento de agentes reativos. Trata-se da representação do estado de um objeto em um determinado período que este participa do processo, podendo o objeto transitar por vários estados dentro de um mesmo processo entretando, estando em um estado de cada vez.(CURY, 2001)

FSM podem ser representada por meio de um Diagrama de Máquina de Estado no qual demonstra o comportamento de um elemento através de um conjunto de transições de estado:

- **Estado** -É uma condição ou situação em que um objeto se encontra num determinado momento;
- **Transições** - Uma transição é um relacionamento entre dois estados, indicando a mudança de um estado para outro após a realização de certas ações. São representadas por uma linha contínua com uma seta apontando do estado origem para o estado final. Uma transição é formada por cinco partes:

**Estado de Origem** -Estado em que o objeto se encontra quando recebe um evento de ativação para realizar uma certa ação.

**Evento de Ativação** - Estímulo que ocorre em um objeto, capaz de disparar uma transição de estado.

**Condição de Proteção / Condição de Guarda** - Expressão booleana entre colchetes que é avaliada quando uma transição é iniciada pelo evento de ativação que define se a transição será ou não executada.

**Ação** - Computação que pode ocorrer em uma transição, normalmente uma chamada de método.

**Estado Destino** - Estado em que o objeto se encontra após a conclusão da transição.

Uma FSM pode ser modelado a como um grafo onde os estados são vértices e as transições são as arestas no caso, as ações são executadas na entrada ou saída dos estados.

## 2.7 Microeconomia

A Microeconomia, ou Teoria de Preços, é a parte da teoria econômica que estuda o comportamento dos agentes econômicos, os mercados nos quais interagem e principalmente a alocação de recursos. Analisa a formação de preços no mercado e os fatores econômicos que determinam tanto o comportamento dos agentes consumidores quanto dos produtores. Sendo um conjunto de teorias sobre o processo pelo qual recursos são alocados e as relações entre preço e dos mercados neste processo.

### 2.7.1 Mercado

A literatura apresenta diversas definições para o conceito de mercado. Dentre estas e neste trabalho, utiliza-se o define mercado como um sistema social e abstrato baseado na divisão das atividades e na propriedade privada dos meios de produção, onde agentes econômicos agem por conta própria para satisfazer suas necessidades interagindo com as necessidades de outros agentes. Nesta perspectiva cada agente é simultaneamente um meio e um fim, isto é, um fim útil em si mesmo e um meio para que outros agentes atendam seus próprios fins (IORIO, 2013)

O mercado é fruto das interação de vários agentes que cooperam sob o regime da divisão das atividades. As forças que determinam o estado do mercado são provenientes dos julgamentos dos agentes e das suas ações baseadas nesses julgamentos. Desta forma, o estado do mercado em um determinado momento é definido pelo conjunto de relações de trocas, proveniente da interação daqueles que estão desejosos de vender com aqueles que estão desejosos de comprar, por meio destas interações são estruturados os preços, sendo o preço o indicador utilizado pelos produtores para decidir o que produzir, como produzir e em que quantidade, sendo o ponto focal para onde convergem e de onde se irradiam as forças que balizam o mercado.

A análise do mercado é realizada por meio da observação dos mecanismos que estabelecem os preços relativos entre os produtos e serviços, tais como a oferta ou demandas, conceitos amplamente utilizados para estabelecer parâmetros de comportamento de modelos econômico abstrato do mercado. (VASCONCELLOS; GARCIA, 1998) Utiliza-se as curvas de demanda e de oferta como ferramentas para avaliar o sistema de formação de preço.

Baseado no comportamento econômico do mercado no passado, são propostos cenários, no intuito de prever as variações nas condições do mercado e os preços futuros. Inicialmente a análise é realizada nas curvas de forma separada, para posteriormente avalia-se a inteiração dos agentes fornecedores e agentes consumidores de produtos. Para a construção das curvas adotaremos a seguinte notação:

- Preço =  $P$
- Oferta =  $O$
- Demanda =  $D$

A oferta  $O$  representa a quantidade que os agentes produtores pretendem produzir e inserir no mercado depende do preço  $P$ . Assim, temos a seguinte função de oferta  $O = O(P)$  que será cada vez maior com o aumento de preços. Onde  $O' = O'(P)$  é a derivada de  $O$  com respeito à variável  $P$

$$O'(P) = \frac{dO}{dP} > 0 \quad (2.7)$$

Assim, a declinação da curva de oferta é crescente, indicando que para um preço maior novos produtos tendem a entrar no mercado, devido ao comportamento dos agentes produtores que ao disponibilizar no mercado seus produtos tem sempre a meta de obter o maior lucro<sup>1</sup> A quantidade ofertada aumenta à medida que o preço aumenta e diminui quando o preço reduz. Logo, a quantidade ofertada é diretamente proporcional ao preço do bem ou serviço, fato evidenciado no gráfico da curva de oferta da Figura 9

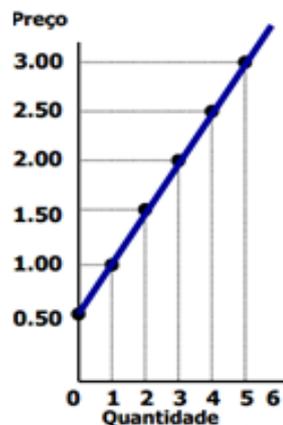


Figura 9 – Curva de Oferta

**Fonte:** (VASCONCELLOS; GARCIA, 1998)

A curva da demanda deriva de hipóteses sobre o processo de decisão dos agentes na ação de escolher entre diversos recursos. Na busca por satisfazer suas necessidades, os agentes avaliam o preço e compram o que for conveniente de modo a minimizar o investimento. Assim, a demanda por um recurso diminui quando o preço aumenta e aumenta quando o preço diminui. Podemos afirmar que a demanda é inversamente proporcional ao

<sup>1</sup> Irving (1906) definiu lucro como o resultado do uso do capital, e este é visto como estoque de riqueza capaz de gerar serviços futuros. O lucro é tido, portanto, como o fluxo de riqueza ou benefícios acima do necessário para manter o capital constante.

preço. De modo que a quantidade demandada pelos consumidores depende do preço, isto é,  $D = D(P)$  e diminui à medida que o preço aumenta. Desta forma assume-se que:

$$D'(P) = \frac{dO}{dP} < 0 \quad (2.8)$$

A função  $D = D(P)$  é denominada função demanda supostamente decrescente. Sendo o gráfico definido pelo conjunto  $(D, P) \in R \times R$  tal que  $D = D(P)$  para todo  $P \in R, P < 0$ . Na figura 10 observa-se este comportamento.

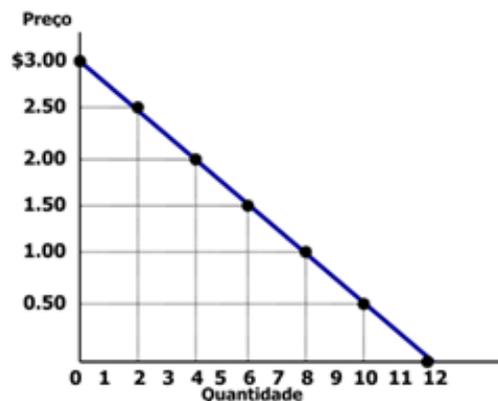


Figura 10 – Curva de Demanda

Fonte:Desenvolvida pelo autor.

### 2.7.2 Ponto de Equilíbrio

Ao avaliarmos as curvas de oferta e demanda separadamente observa-se o efeito do da variável preço na decisão em produzir (oferta) e no consumo (demanda). Desta forma, há de se considerar que exista um ponto de equilíbrio no sistema, que seria a intercessão das curvas de oferta e demanda. Afirma-se que isso ocorre quando a quantidade que os agentes consumidores desejam comprar é igual à quantidade que os agentes produtores desejam produzir. Chamaremos de  $p_e$  o preço de equilíbrio, valor pelo qual a quantidade ofertada é igual à quantidade demandada  $q_e$ . Denota-se  $q_e = O(p_e) = D(p_e)$  para a quantidade do preço de equilíbrio  $q_e$ . Podemos observar no gráfico da Figura 11 a plotagem das duas curvas e a formação do ponto de equilíbrio.

### 2.7.3 Teoria da teia de aranha

O comportamento do mercado é dinâmico, apresentar variações nas quantidades demandas, ofertadas e nos preços propostos a cada intervalo de tempo. Apresentase um modelo que pretende interpretar as flutuações periódicas de terminados tipos de mercados. O modelo da teia de aranha ou cobweb problem é uma formulação que relaciona o comportamento do preço mediante a demanda e oferta de um produtor em determinado

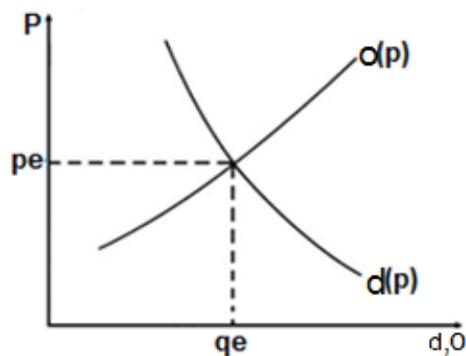


Figura 11 – Ponto de Equilíbrio

**Fonte:**(MANKIWI, )Adaptada

período de tempo, fundamentada principalmente nos preços observados no período anterior. A teoria que propõem a obtenção de um ponto de equilíbrio estável entre preços e quantidades demandadas e ofertadas depende em grande parte das inclinações das curvas de demanda e oferta do produto estudado. (SARAIVA, 2011)

As interações dos agentes econômicos no mercado tende a conduzir o preço ao equilíbrio. Suponha  $p_1 > p_e$  em um determinado tempo  $t(0)$ , configuração que incentivaria agentes produtores a produzir mais, ao ponto de superar a quantidades que agentes consumidores estariam dispostos a adquirir, por conseguinte haveria um excesso de oferta. Para que o excesso pudesse ser vendido os produtores devem reduzir o valor de  $p_1$ . A medida que  $p_1$  caísse, a quantidade demanda aumentaria e a quantidade ofertada diminuiria, até que o preço de equilíbrio,  $p_e$  fosse alcançado. Considere-se que o preço  $p_2$  em um determinado tempo  $t(0)$  como o valor inicialmente inferior ao preço de equilíbrio  $p_e$ . Esta configuração geraria uma escassez de oferta, ou seja, a quantidade de demandada excederia a quantidade ofertada isso ocasionaria uma pressão ascendente sobre os preços, à medida que os compradores se mostrassem dispostos a pagar mais pelas quantidades existentes e os produtores reagissem com aumento de preço e de produção. O preço acabaria atingindo o valor  $p_e$ . A Figura 12 ilustra o que foi discutido acima.

Em síntese, a teoria busca estabelecer que a oferta de um produto esta relacionada com a função do seu preço passado, enquanto a demanda desse mesmo bem é uma função do preço corrente. Como as decisões quanto à produção devem ser feitas antecipadamente, essas decisões baseam-se nos preços esperados. O teorema da teia-de-aranha postula que esses preços esperados são os preços correntes na época das decisões quanto à produção. Suponha que a quantidade de determinado produto vendido em um  $t$  específico dependa das decisões tomadas em  $(t - 1)$ ,  $(t - 2)$  logo, se o preço está em alta, a tendência é que os produtores desejem produzir e vender mais. Entretanto, se os preços estiverem em baixa, eles pretendem produzir e vender menos. Assim, os produtores terão que prever os preços para o próximo  $t + 1$ . Ele poderia utilizar os preços do  $t$  corrente como base para o  $(t + 1)$ .

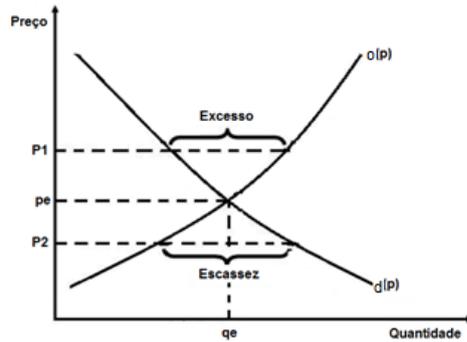


Figura 12 – Curva de Equilíbrio Avaliação  
(MANKIW, )

Para ilustrar o que está descrito acima, considere as seguintes equações de comportamento linear para a oferta e demanda: (VASCONCELLOS; GARCIA, 1998)

$$\text{Curva da Oferta} \quad O(t) = a + bpe_{(t-1)} \quad (2.9)$$

$$\text{Curva da Demanda} \quad D(t) = \alpha + \beta P_t \quad (2.10)$$

Onde  $\alpha$ ,  $\beta$ ,  $a$  e  $b$  são constantes positivas e  $t$  e  $(t - 1)$  são índices que se referem ao tempo atual e ao tempo anterior respectivamente. A condição de equilíbrio no mercado do produto deve ocorrer quando  $D(pe) = O(pe)$ . Assim,  $\alpha - \beta pe_{(t)} = a + bpe_{(t-1)}$ . Tem-se que o preço atual está relacionado ao preço  $(t - 1)$  anterior, ou seja,

$$Pe_t = \frac{(\alpha - a)}{\beta} - \frac{b}{\beta pe_{(t-1)}} \quad (2.11)$$

O preço de equilíbrio  $p_e$  é o único valor que satisfaz  $p_e = p_{(t-1)} = p_{(t)}$ , ou seja, o preço de equilíbrio é o valor que permanece inalterado ao longo dos  $t$  períodos . Implica que:

$$pe = \frac{(\alpha - a)}{\beta} - \frac{b}{\beta} xpe \quad (2.12)$$

Tem-se três possibilidades relacionadas aos valores para  $b$  e  $\beta$  São elas:

- **Convergente** ( $b > \beta$ ) O ajustamento entre preços e quantidades no modelo da teia de aranha de ciclo convergente ocorre quando a inclinação da curva de demanda do produto for menor, em termos absolutos, do que a inclinação da curva de oferta do produto, conforme ilustrado na Figura 13
- **Divergente** ( $b < \beta$ ) O ciclo divergente ocorre quando a inclinação da curva de demanda do produto for maior, em termos absolutos, do que a inclinação da curva de oferta do produto como pode ser observado na Figura 14.

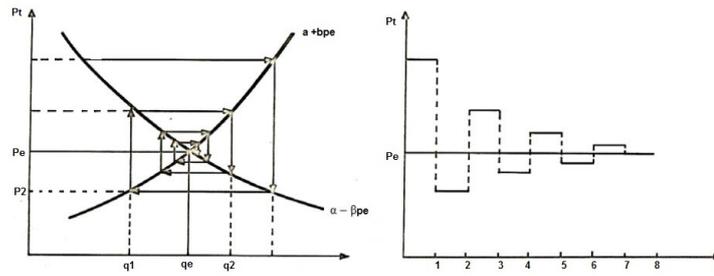


Figura 13 – Modelo da teia de aranha de ciclo convergente.

**Fonte:**(VASCONCELLOS; GARCIA, 1998) Adaptada

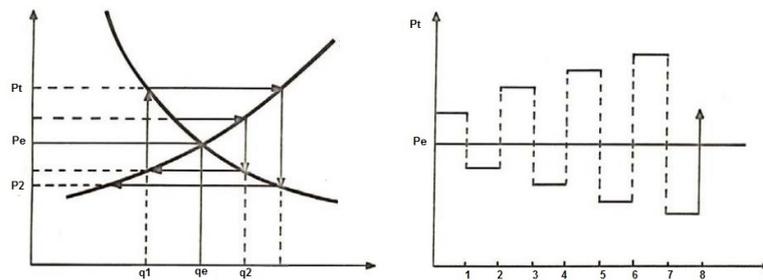


Figura 14 – Modelo da teia de aranha de ciclo divergente.

**Fonte:**(VASCONCELLOS; GARCIA, 1998) Adaptada

- **Ciclo Contínuo]** ( $b = \beta$ ) O processo de ajustamento do modelo com ciclo contínuo retrata uma situação em que as curvas de demanda e de oferta do produto apresentam, em termos absolutos, a mesma inclinação conforme pode ser observada na Figura 15

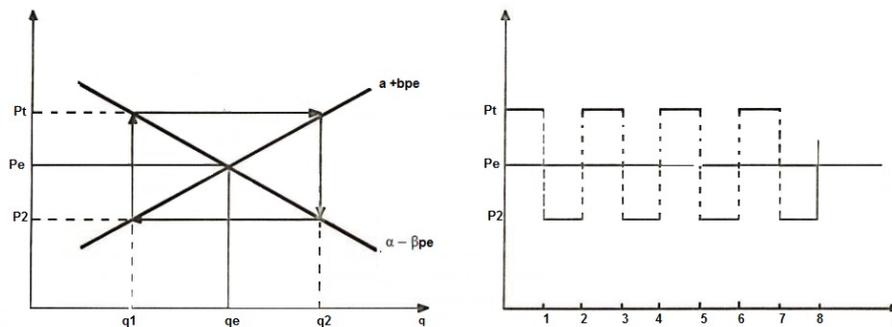


Figura 15 – Modelo da teia de aranha de ciclo contínuo.

**Fonte:**(VASCONCELLOS; GARCIA, 1998) Adaptada

## 3 Proposta

O impacto financeiro das atividades pertinentes ao transporte e distribuição de insumos representa relevante parcela na composição dos custos das empresas, visto que os investimentos relacionados ao transporte geralmente são elevados, absorvendo a maior porcentagem do investimento das atividades logísticas justificando-se, dessa forma, novos investimentos em algoritmos que minimizem tais custos (BALLOU, 2001). Estudos demonstram que aplicações de sistema para geração de itinerário de veículos proporcionam reduções de custos de transporte na ordem de 15%, sendo uma alternativa razoável para aumentar a competitividade das empresas.

Apesar dos avanços com algoritmos exatos e o melhor desempenho computacional com os hardwares atuais, não se conhece um algoritmo com tempo polinomial capaz de localizar soluções ótimas, fato que motiva o desenvolvimento de abordagem utilizando heurísticas e metaheurísticas para a solução de problemas desta natureza. Portanto, novas abordagens heurísticas para geração de rotas são de relevância significativa, devido ao tema ter aplicação nos mais diversificados cenários, como ambientes fabris, transporte urbano, distribuição de insumos, dentre outros.

A proposta central deste trabalho é o desenvolvimento de uma heurística para geração de rotas utilizando o conceito de mercado e do processo de formação de preço aplicados a um algoritmo de caminho mínimo. O preço é considerado a principal variável no processo de alocação de recursos pelo mercado, por simplificar, em uma única variável a complexidade presente na composição dos custos envolvidos na produção de determinados bens. Característica que pretende-se incorporar a solução quando se utiliza o conceito de preço na proposta. Assume-se por premissa que seja possível realizar uma abordagem utilizando o conceito de **preço** ao invés da tradicional abordagem de **custo**.

Conceitualmente, mercado é uma entidade abstrata, definida como fruto das interações entre compradores e vendedores de recursos. Somente é possível utilizá-lo por meio da simulação das interações entre os agentes econômicos e só posteriormente, analisar os resultados, frutos das interações. Por este motivo, propõe-se o desenvolvimento de um mercado virtual simulado, no qual, os produtos são trechos dos caminhos que serão vendidos por um determinado preço. Para a implementação do mercado virtual torna-se necessário a definição de um cenário para as interações entre os agentes. Com a finalidade de simplificação adotaremos como sinônimos rota e caminho.

### 3.1 Cenário

Define-se como cenário para implementação do mercado à logística de movimentação interna de insumos em uma planta fabril, composta por células de produção geograficamente separadas, um depósito central e uma base dos robôs autônomos para realizar a movimentação dos insumos. Todos os elementos interligados por uma grade, ou caminho em forma de matriz. Conforme figura 16.

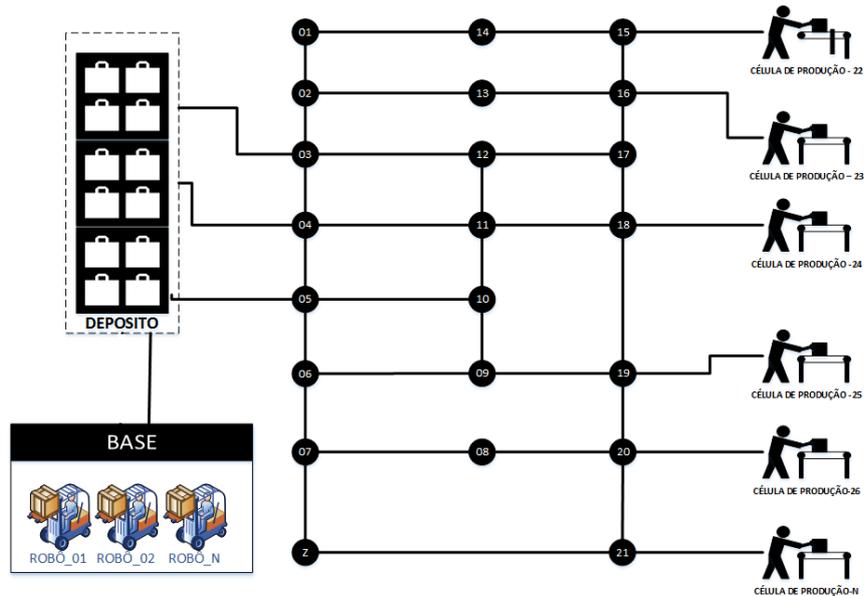


Figura 16 – Modelo Planta Fabril

Fonte: Desenvolvido pelo autor

O cenário segue o padrão dos sistemas de movimentação de matéria prima em indústria com produção no formato em célula, onde cada cliente (célula de produção) apresenta demandas individuais de insumos. Utiliza-se da teoria de grafo para a modelagem da planta fabril. Assim, será representada por um grafo:  $G = (V, A)$  no qual  $V$  é um conjunto de vértices que representam os cruzamentos entre os trechos da grade e  $A$ , um conjunto de arestas  $(i, j)$  que representam a grade. O depósito será representado pelo vértice  $v_0$  e os clientes são representados pelos vértices terminais do grafo. Ao conjunto de arestas associa-se os elementos físicos como a distância entre os vértices  $(i, j)$  denominada  $d_{(i,j)}$  e o tempo de deslocamento  $t_{(i,j)}$  ambas características consideradas fixas. Integra-se também a demanda  $de_{(i,j)}$  uma característica dinâmica do cenário.

A movimentação de mercadoria será realizada por robôs independentes em atendimento a um processo de **PCP- Planejamento e Controle de Produção**<sup>1</sup>. As soli-

<sup>1</sup> Planejamento e Controle da Produção (PCP) é uma função técnica e administrativa que tem o objetivo fazer os planos que orientarão a produção e servirão de guia para seu controle e formular os planos para organizar a aplicação dos recursos humanos e materiais de modo a controlar as ações para correções de eventuais desvios e minimizar perdas no processo produtivo

citações serão agrupadas em pacotes e expostas com o passar do tempo aos robôs, que atendem a solicitação realizando entregas de material, saindo do depósito em direção as células de produção.

A inclusão do conceito de preço na modelagem do cenário está intimamente interligada com a proposta do desenvolvimento da heurística utilizando conceito de mercado. Sendo o preço uma entidade dinâmica, utilizada pelo mercado para alocação de recursos de forma ótima ou quase ótima, investigamos a possibilidade de que esta mesma dinâmica seja alcançada pelo heurística proposta. Dentro desta perspectiva surgem alguns conceitos importante na abordagem. São eles:

- **Vendedor** - Agente responsável por determinar as característica fundamentais para a regras da formação dos preços das rotas.
- **Robô** - Agente responsável por avaliar a lista de demandas do PCP, retirar a demanda da lista, calcular quais os trechos vão compor a rota, informar seu estado e realizar a entrega das matérias primas na célula de produção. É o consumidor de rotas
- **Mercado** - Entidade abstrata proveniente das interações entre agentes econômicos (vendedor e compradores). O mercado gerencia a disposições das células de produção, demandas de cada trecho, os preços das últimas negociações, quais trechos estão em uso e demais informações referentes a composição dos recursos disponíveis para uso.
- **Produto** - São os recursos comercializados entre o vendedor e os robôs neste caso são as arestas da grade modelados como aresta  $(i, j)$  em um grafo  $G = (V, A)$  aos quais são associados os preços.

Para a implementação de um mercado é necessário criar os relacionamentos entre os agentes econômicos e mediante as interações avaliar quais caminhos foram selecionadas pelos robôs e para isto utilizaremos uma simulação de mercado centralizado com um único vendedor atendendo a vários consumidores representados pelos robôs.

Nesta abordagem a variável demanda tem papel relevante na atualização dos preços, conforme as arestas são utilizadas os preços são atualizados, tendo como referência o comportamento da demanda, o preço tende a aumentar ao passo que as arestas são selecionados para formar as rotas. Assim, a aresta selecionada em  $t$  tende a ser menos vantajosa para o próximo robô em  $t+1$ . De forma contrária, a não utilização de determinadas arestas, por serem menos vantajosa  $t$ , o que fazem com que a mesma passe por atualização de preço o que ocasionará uma redução em  $t + 1$ , fazendo com que fique mais barata. Desta forma, ainda que uma aresta seja mais longa, gradativamente poderá ser selecionada para integrar a rota.

A variável **preço**, proposta nesta abordagem, é calculada utilizando duas vertentes? o **preço inicial**  $P_{(i,j)}$  em  $t = 0$ , que tem sua composição baseada nas características físicas da aresta, e o **preço**  $P(i,j)$  em  $t > 0$ , sendo o preço inicial atualizado mediante as iterações entre os agentes do mercado conforme a simulação é executada. Apresentamos a seguir os parâmetros que compõem os preços:

- Distância =  $d_{(i,j)}$ ;
- Tempo de deslocamento =  $t_{(i,j)}$ ;
- Demanda =  $de_{(i,j)}$ ;
- Coeficiente de valorização =  $\alpha$ ;
- Coeficiente de desvalorização =  $\beta$ ;
- Taxa de variação de demanda =  $\delta_{de_{(i,j)}} = \{de_{(i,j)_{(t+1)}} - de_{(i,j)_{(t)}}\}$

Tendo como diretrizes o mercado econômico e para que a dinâmica dos preços fosse simulada tornou-se necessário estabelecer coeficientes para a atualização dos preços. As variáveis  $\alpha$  e  $\beta$  são os variáveis de valorização e desvalorização dos preços respectivamente. Sendo  $\alpha$  um valor maior ou igual a um que multiplica os preços das arestas utilizadas na rota selecionada e  $\beta$  um valor inferior a um que multiplica os preços das arestas que não fazem parte da rota selecionada. Ambas as variáveis são repassadas pelo PCP de acordo com nível de *espalhamento* dos robôs desejado.

Difine-se o **preço inicial** em  $t = 0$ , da seguinte forma:

$$P_{(i,j)} = (d_{(i,j)} \cdot lucro) \quad (3.1)$$

Onde,  $P_{(i,j)}$  é o preço em  $t = 0$ ,  $d_{(i,j)}$  a distância entre  $(i,j)$  e o lucro é um valor percentual da distância. A atualização dos **preços** para  $t > 0$  seguem a seguinte regra:

$$P_{(i,j)_{(t+1)}} = \begin{cases} se \delta > 0, & P_{(i,j)_{(t)}} \cdot \alpha \\ senão, & P_{(i,j)_{(t)}} \cdot \beta \end{cases} \quad (3.2)$$

Os robôs realizam a compra ou seleção das arestas que integram o caminho de forma independente seguindo suas premissas no intuito de atender as suas necessidades, comportamento este, similar ao de um consumidor no mercado, que compra determinado recurso utilizando parâmetros subjetivos. Por conseguinte, o robô comprará as arestas que julgar mais vantajoso avaliando o preço em um determinado  $t$ . O propósito do robô comprador é minimizar gastos, assim, o problema pode se formulado da seguinte forma:

$$Minimizar Z = \sum_{i \in N} \sum_{j \in N} P_{(i,j)} X_{(i,j)} \quad (3.3)$$

Sendo  $X$  o vetor de decisão em que,

$$X_{(i,j)} = \begin{cases} 1, & \text{se } (i,j) \text{ pertence ao caminho} \\ 0, & \text{se } (i,j) \text{ não pertence ao caminho} \end{cases} \quad (3.4)$$

A figura 17 mostramos uma possível classificação das simulações, encontradas na literatura.



Figura 17 – Visão Geral - Simulações

Fonte: Desenvolvida pelo autor

Considerando a classificação anterior podemos classificar a simulação da heurística proposta como:

- **Determinística** - Como o conjunto de dados de entradas (demandas, grid, quantidade de robôs, e número de interações) da simulação é conhecido e determinadas pelo setup, o resultado é um único conjunto de saídas.
- **Discreta** - O modelo é discreto devido a passagem de tempo ser realizada em intervalos discretos, isto é de um evento a outro. Nesta proposta o estado do sistema não se altera ao longo do intervalo entre a ocorrência de dois eventos consecutivos.
- **Dinâmicas** - Por representar um sistemas que muda mediante a mudanças nos dados de entrada.
- **Tempo Simulado** - O tempo da simulação, isto é , o tempo de cada passo é arbitrariamente definido não estando vinculado ao tempo de execução da simulação ou tempo real.

# 4 Implementação

A implementação da proposta apresentada foi realizada utilizando o paradigma de programação orientada a objeto por meio da linguagem de programação Java.

Seguiu-se a premissa previamente apresentada de utilizar os conceitos de mercado e processo de formação de preço para modelagem da heurística para geração de rotas.

O desenvolvimento dos agentes econômicos e de suas relações no mercado virtual foram implementadas em classes com atributos e métodos.

Este capítulo dedica-se a detalhar as classes e seus relacionamentos.

## 4.1 Mercado

Sendo o mercado uma entidade abstrata, fruto das interações entre aos agentes econômicos que o compõem, sua implementação segue esta premissa e basea-se no anseio de reproduzir de forma simulada tais características. Desta forma, os esforços da modelagem e da implementação centrou-se no agente vendedor, compradores e no produto e suas relações. Na Figura 18 são demonstrados os componentes básicos e suas relações no mercado. As relações entre os agentes econômicos são apresentadas de forma ilustrativa, aqui colocadas apenas por formalidade, não tratamos com rigor semântico. Posteriormente apresentaremos os diagramas das classes da simulação modelados em UML para sanar esta lacuna.

### 4.1.1 Cenário

Conforme exposto na proposta deste trabalho, o cenário implementado para simulação do mercado virtual de rotas é o de uma planta fabril e o recursos negociado são as arestas da grade que interligam o depósito central as células de produção. Na figura 19 apresenta-se uma representação gráfica da **planta fabril** com as informações físicas.

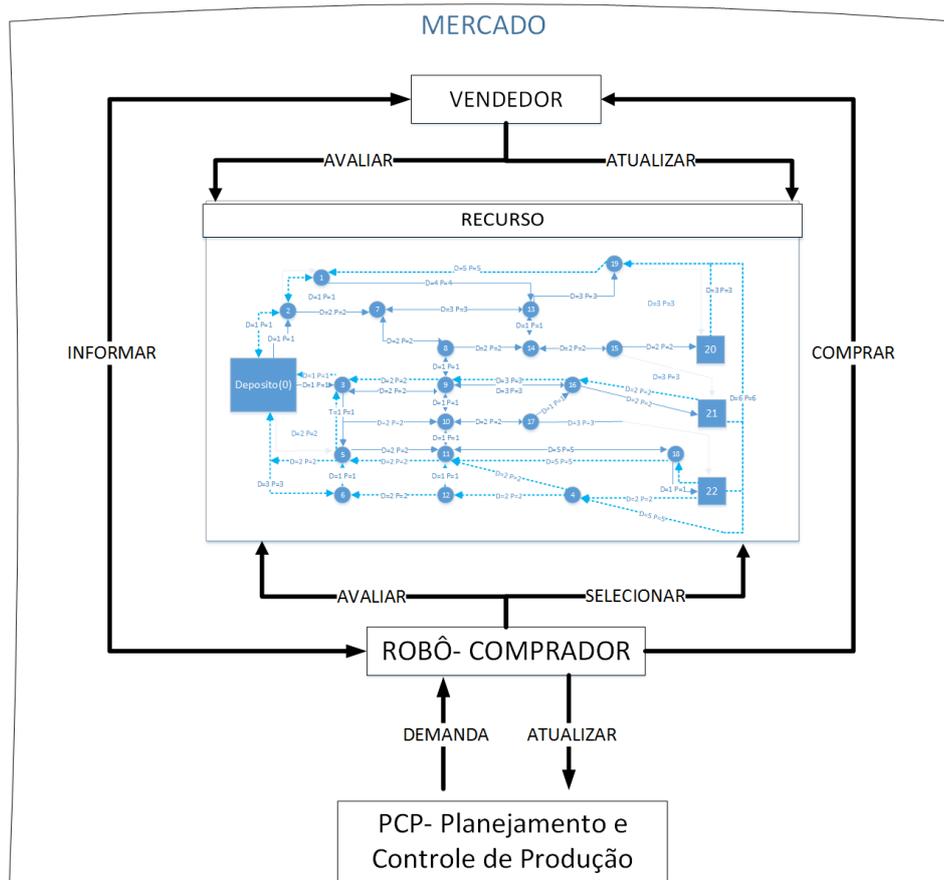


Figura 18 – Mercado Relações Básicas

Fonte: Desenvolvido pelo autor

As características foram modeladas utilizando teoria de grafo. Apresentamos a seguir os detalhes das classes, atributos os principais métodos juntamente com os diagramas de classe em UML.

#### 4.1.2 Grid\_Aresta

A classe aresta representa parte do grid que interliga as células de produção ao depósito central e tem os seguintes objetivos, modelar as características físicas da planta fabril a ser simulada e na perspectiva mercadológica, serem os produtos comercializados no processo de negociação na formação das rotas. No intuito de alinhar a modelagem com estas duas premissas foram considerados os seguintes atributos:

- **Preços:** Variável do tipo double que armazena os valores atribuídos as aresta mediante as interações dos agentes. Conforme as arestas são utilizadas este atributo sofre alterações;
- **Demanda:** Atributo do tipo inteiro que armazena os valores referente ao quantitativo de uso de uma determinada aresta, sendo contabilizado a cada instante  $t$  na simulação;

- **DemandaAnt:** Variável do tipo inteiro que armazena valores referente ao quantitativo de uso de uma determinada aresta sendo contabilizada a cada  $t - 1$  da simulação;
- **VerticeOrigem, VerticeDestino:** Atributos que referem-se a orientação da aresta são úteis para o método de localização;
- **Distância:** Atributo relacionado a distância entre as células de produção e o depósito. Embora a seleção de rotas seja realizada considerando o atributo preço, a simulação do descolamento dos robôs é influenciada diretamente pelo aspecto distância;
- **Tempo:** Variável que refere-se ao tempo necessário para um robô desloca-se por uma determinada aresta.

Os métodos implementados na classe aresta são centrados na elaboração da estrutura de dados para receber as informações das características físicas da planta fabril. Em geral são métodos acessores *getters* e modificadores *setters* de atributos. O método *String toString* se diferencia dos demais, em razão de ter como função retornar uma representação string do objeto. Relevante para a construção dos relatórios de dados da simulação.

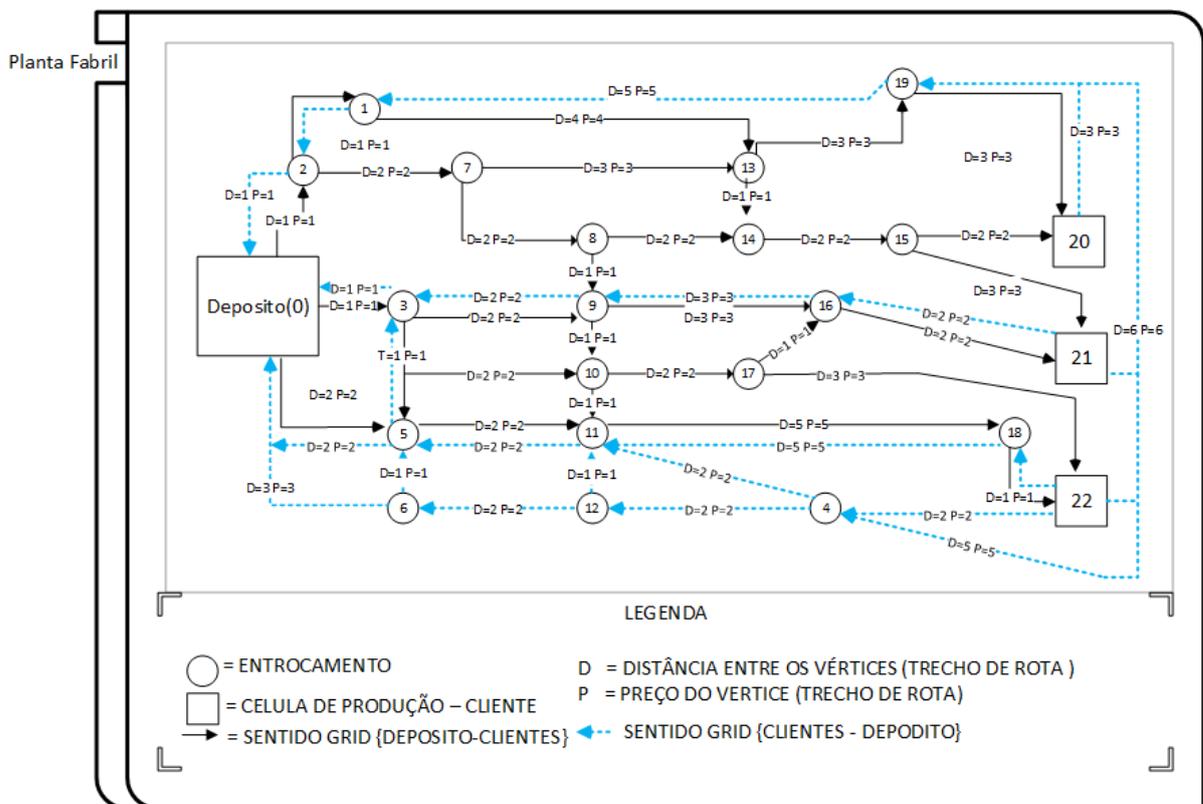


Figura 19 – Cenário Proposto

Fonte: Desenvolvido pelo autor

### 4.1.3 Grid\_Vertices

A classe foi modelada para implementar as características físicas e funcionais da planta fabril. Tais como, as células de produção (localização dos clientes), os pontos de intercessão do *grade* e deposito são modelados como vértices. Para que a classe realizasse corretamente seus objetivos incluímos os seguintes atributos:

- **ID:** Variável do tipo inteiro que guarda a identificação do vértice;
- **Distância:** Atributo relacionado a distância entre as células de produção e o deposito. Embora que a seleção de rotas seja realizada considerando o atributo preço a avaliamos as rotas pela distância percorridas;
- **Visitado:** Variável booleana utilizada no processo de rotulação dos vértices do algoritmo de dijkstra;
- **Arestas;** Lista de arestas vinculadas ao vértice implementada em Arraylist;

Os métodos implementados na classe vértices são a base para o funcionamento do algoritmo dijkstra. Alguns são métodos acessores *getters* e modificadores *setters* de atributos. Os metodos **CompareTo(Vertex vertice)** e **public boolean equals(Object obj)** são utilizados pelo dijkstra para comparação entre vértices e atributos. O método **toString()** utilizado para retornar dados dos experimento.

### 4.1.4 Planta\_fabril

Classe elaborada para armazenar e manipular vértices e arestas. Nela estão presente as informações pertinente as características físicas da planta fabril. Utiliza-se para esta junção a estrutura de dados Arraylist interface List e os seguintes métodos são utilizados para a manipulação:

- **set\_Vertice()** Método utilizado para adicionar vértice no grafo implementado pelo contruto da classe. Recebe como parâmetro um objeto da classe vertice e tem como retorno void;
- **findVertice()** Método utilizado para localizar um vértice no grafo, recebe como parâmetro o ID do vértice e tem como retorno um vértice ou null caso o vértice não seja localizado;
- **setArestas()** Método utilizado para modificar atributos de arestas no Arraylist. Recebe como parâmetro um objeto da classe vértice e tem como retorno void;

- **findAresta()** Método utilizado para localizar uma aresta no ArrayList, recebe como parâmetros o vértice origem e o vértice destino do aresta tem como retorno uma aresta ou null caso a aresta não seja localizada;

Na Figura 20 é apresentado um diagrama que contém as classes **Planta\_FabrilGrafo**, **Grid\_Vertice** e **Grid\_Aresta** que modelam a planta fabril e suas funcionalidades.

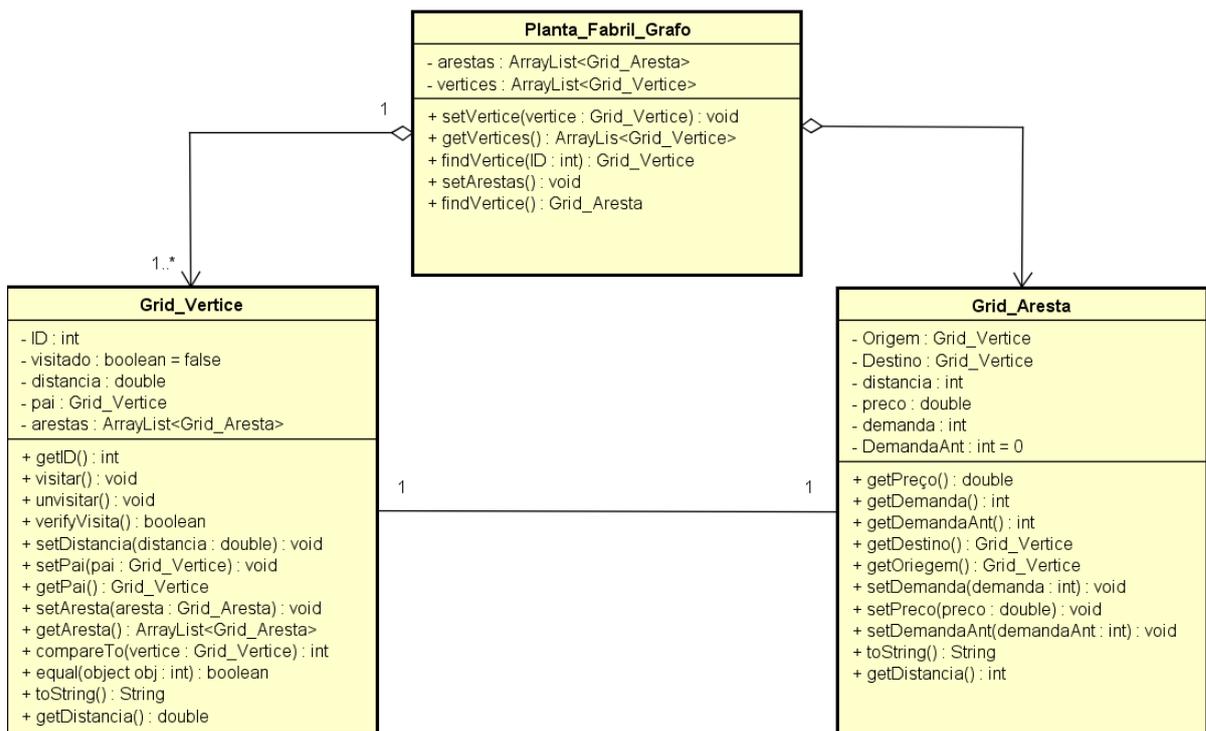


Figura 20 – Diagrama de classe UML- Planta Fabril

**Fonte:** Desenvolvido pelo autor

#### 4.1.5 Vendedor

A classe responsável por realizar a leitura dos dados de entrada do arquivo de configuração externo e com as informações instanciar e preenche as classes que modelam a estrutura física do cenário. Conforme a simulação é realizada a classe vendedor realiza a atualização das informações mercadológicas, preços e demandas dos recursos. A seguir apresenta-se aos atributos e principais métodos da classe:

- **EstruturarRecursos()** Método que instância um objeto da classe **Planta\_fabril** e preenche com informações físicas e mercadológicas da cenário a ser simulado.
- **Atualizar()** Método que realiza a atualização das informações mercadológicas na simulação, conforme os robôs selecionam as rotas, este método atualiza os preços, as

informações de demanda e aplica os coeficientes de valorização e desvalorização do preço.



Figura 21 – Classe Vendedor - Modelada em UML

**Fonte:** Desenvolvido pelo autor

#### 4.1.6 Requisicao

Classe que modela as requisições de entrega dos materiais. Os objetos instanciados por essa classe contém informações úteis aos robôs no processo de entrega dos insumos, tais quais, a origem e o destino, ID da entrega. Os métodos desta classes são todos assessores **Getters** e modificadores **Setters** de atributos apenas o método construtor da classe se diferencia dos demais.

- **getID()** Métodos utilizado para acessar os identificadores das solicitações de entrega. O método retorna um número inteiro;
- **getOrigem()** Método para acessar a informação ponto de partida do material a ser entregue, o método retorna um número inteiro que faz referência a um vértice dentro do grafo que implementa planta fabril;
- **getDestino()** Método para acessar a informação ponto de destino do material a ser entregue, o método retorna um número inteiro que faz referência a um vértice dentro do grafo que implementa planta fabril;

#### 4.1.7 PCP

Classe que realiza a leitura de um arquivo externo para obter dados de parâmetros da simulação, faz a implementação de objetos da classe requisição. Após a criação dos objetos os incluem em uma listagem para posteriormente serem acessados pelos robôs. O arquivo de configuração contém informações das entregas tais como, origem, destino, Id. A sequência em que as requisições são implementadas e inseridas na listagem que os robôs seguirão para realizá-las. A seguir pode-se observa os diagramas das classes **PCP** e **Requisição**.

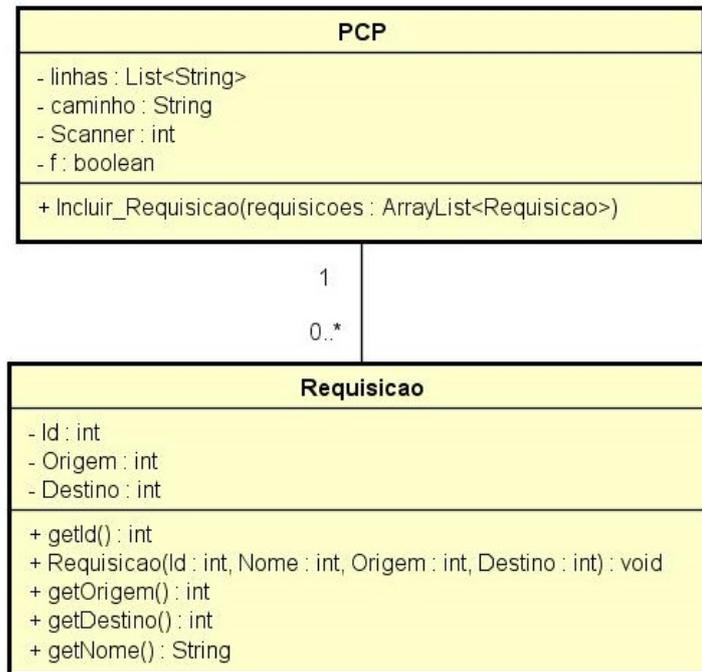


Figura 22 – Classes PCP e Requisição em UML

**Fonte:** Desenvolvido pelo autor

#### 4.1.8 Dijkstra

Classe que implementa o algoritmo para seleção das arestas que compõem as rotas. O procedimento retorna por meio da avaliação do preço de cada aresta da rota que apresenta o preço mínimo em determinado  $t$ . A modelagem segue o algoritmo tradicional do Dijkstra tem os seguintes atributos:

- **Unknom** ArrayList com os vértices ainda não visitados;
- **CurrentVertice** Vértice que está no processo de relaxamento ou rotulação;
- **Caminho** ArrayList com os vértices que vão compor a rota de preço mínimo calculada pelo algoritmo.
- **nearest** Variável que armazena a vértice mais próxima da avaliada.

A implementação é realizada por meio do método **FindMenorPreco** que recebe os seguintes parametros: (**Planta\_fabril**, **Grid\_Vertice\_Origem**, **Grid\_Vertice\_Destino**).

#### 4.1.9 Robo

Classe que modela e implementa robôs de transporte de um sistema de movimentação de insumos em uma planta fabril. Contém atributos e métodos necessário para simular a movimentação e o relacionamentos com os demais componentes do sistema. Sendo responsável por chamar os principais métodos da simulação tais como, o método para calcular as rotas e o de atualização dos preços. Para a implementação da classe utilizou-se do conceito de máquina de estado. Assim, conforme a simulação é executada, durante a entrega dos insumos os robôs variam entre alguns estados. A seguir apresentam-se os principais atributos da classe:

- **isFinished** Variável utilizada para sinalizar o momento que o sistema deve realizar a impressão dos dados de saída.
- **RequisicaoAtual** Variável que armazena a requisição que o robô está atendendo em determinado período;
- **CountdownDistancia** Variável utilizada para armazenar as distâncias que o robô devera percorre. Utiliza-se como um contado decrescente que a cada interação retira o valor percorrido pelo robô.
- **Rota** - `ArrayList` que armazena as arestas pertencente a rota selecionada pelo robô.
- **ID** e **ProximoId** - Variáveis utilizadas para armazenar a identificação do robô que está realizando a entrega.

Apresenta-se a seguir os principais métodos que modelagem o comportamento e funcionamento da classe robô são:

- **AtualizarRob** - O método `AtualizarRob` é o principal para a realização da simulação das do processo de movimentação de insumos por robôs. Neste método está inclusa uma Máquina de Estado que contém o comportamento do objeto da classe, os quais recebem como parâmetros os seguintes atributos: `ArrayList` `requisições` e um objeto instanciado do classe `vendedor`.
- **CalcularRota** - Método que tem a função de calcular a rota de preço mínimo no grid da planta fabril. Recebe com parâmetro dos seguintes atributos: `origem`, `destino` e `vendedor` tem como retorno um `ArrayList` de `Arestas`.

Conforme exposto anteriormente, a classe robô é modelada utiliza-se dos conceitos de **Maquina de Estado**. Desta forma, apresentamos a seguir os estados implementados e seus funcionamentos detalhados. Estados do robô conforme figura 23:

**DESOCUPADO** - Estado inicial dos robôs, sinaliza que estão aguardando o surgimento de requisições de entrega. A cada interação a listagem de requisições é avaliada se estiver vazia o robô continua no estado, caso, uma requisição seja incluída o robô a retira e inicia os métodos para calcular a rota, atualizar preços e modifica seu estado para **DESLOCAMENTOCLIENTE**.

**DESLOCAMENTOCLIENTE** - Este estado sinaliza que os robôs estão em processo de deslocamento entre o deposito e os clientes para realizar a entrega de insumo. A cada interação é avaliado se chegou ao seu destino e ao chegar o robô modifica seu estado para **ENTREGANDOINSUMOS**.

**ENTREGANDOINSUMOS** - Estado que sinaliza que o robô esta realizando a entrega ao cliente. Este processo tem duração de 4 interações. Ao final das interações o robô modifica para **RETORNODEPOSITO**.

**RETORNODEPOSITO** Este estado sinaliza que os robôs estão em processo de deslocamento entre o deposito e os clientes para realizar a entrega de insumo. A cada interação é avaliado se chegou ao seu destino e ao chegar o robô modifica seu estado para **DESOCUPADO**.

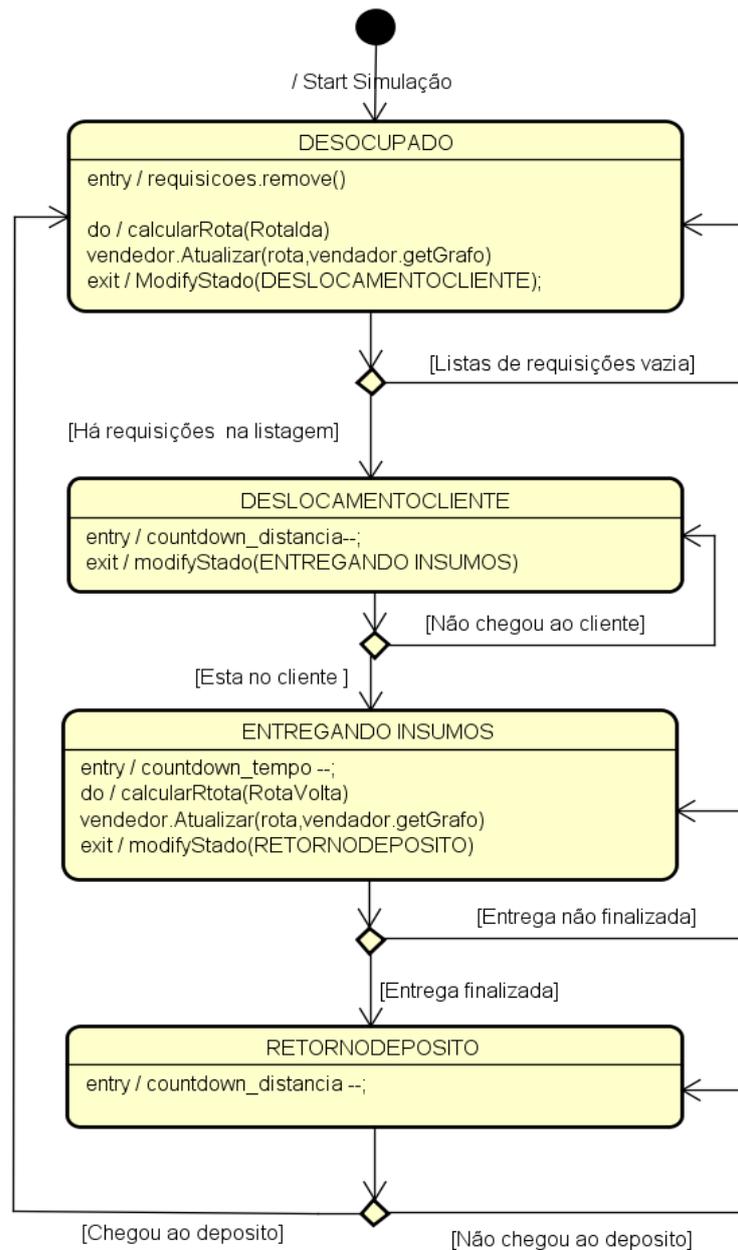


Figura 23 – Máquina de Estado - Classe Robô

**Fonte:** Desenvolvido pelo autor

#### 4.1.10 Classe Main

A classe Main é a principal da simulação sendo responsável pela coordenação das ações das demais classes e pela geração dos relatórios com dados de saída.

Os relatórios implementados são no formatos XML e TXT foi necessário para sua implementação a incluir duas bibliotecas para leitura e geração de arquivos externos são `apache.poi.hssf.usermodel` e `java.io`. Nesta classe é configurado o número de ciclos

da simulação, Quantidade de Robôs e os coeficiente para atualização dos preços e contém principal da simulação. Apresentamos na Figura 24 o diagrama de classes da simulação.

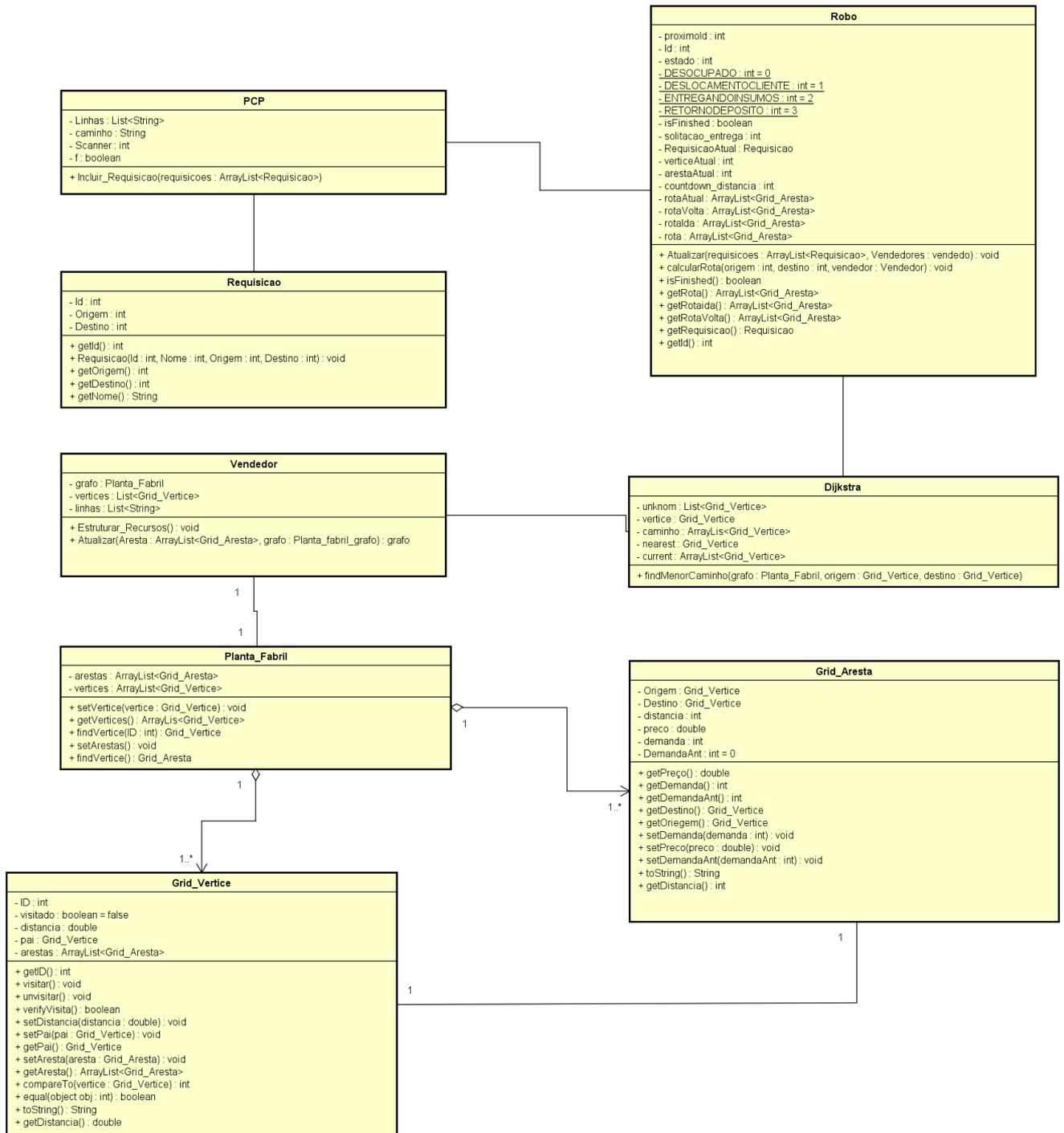


Figura 24 – Diagrama de classe completo - Simulação

Fonte: Desenvolvido pelo autor

# 5 Experimentos e Resultados

Este capítulo apresenta os experimentos concebidos no âmbito desta dissertação, os quais visam validar as hipóteses acerca da aplicabilidade dos conceitos de mercado na heurística proposta para geração de rotas. Por questão de simplicidade, na implementação da grade com os trechos que formam as rotas foram construídos caminhos distintos para a entrega do material e para retorno a base.

A heurística proposta é avaliada por meio de experimentos que simulam a movimentação de robôs durante a logística de abastecimento de célula de produção em uma planta fabril, na simulação os trechos que formam as rotas são vendidos aos robôs em um mercado virtual.

## 5.1 Função Preço X Função Custo

A inovação da heurística proposta, está presente na utilização de uma **função Preço** responsável por atualizar do preços dos trechos que constituem a grade que interligam as células de produção. Desta forma, para avaliar a performance da heurística, propomos a comparação com uma abordagem tradicional do problema que faz uso da **Função de Custo** na geração das rotas. Adotaremos as seguintes funções nos experimentos:

### 5.1.1 Função Preço

Função que depende unicamente do comportamento do mercado, sendo definida da seguinte forma:

No instante  $t = 0$ ,

$$P_{(i,j)} = (d_{(i,j)} \cdot lucro) \tag{5.1}$$

Onde,  $d_{(i,j)}$  a distância entre  $(i, j)$  e o lucro é um valor percentual da distância. A atualização do preço  $P_{(i,j)}$  para  $t > 0$  segue a seguinte regra:

$$P_{(i,j)(t+1)} = \begin{cases} \text{se } \delta > 0, & P_{(i,j)(t)} \cdot \alpha \\ \text{senão,} & P_{(i,j)(t)} \cdot \beta \end{cases} \tag{5.2}$$

### 5.1.2 Função Custo

Função que considera os parâmetros físicos e lógicos do problema, adotamos para sua composição a distância entre as vértices, o tempo de deslocamento e demanda de uso das arestas deste modo, sua formulação é a seguinte:

$$C(ij) = Pd(d_{ij}) + Pt(t_{ij}) + PD(D_{ij})^T \quad (5.3)$$

$$\text{Minimizar } Z = \sum_{i \in N} \sum_{j \in N} C_{(i,j)} X_{(i,j)} \quad (5.4)$$

Sendo  $X$  o vetor de decisão em que,

$$X_{(i,j)} = \begin{cases} 1, & \text{se } (i, j) \text{ pertence ao caminho} \\ 0, & \text{se } (i, j) \text{ não pertence ao caminho} \end{cases}$$

Sendo,

$d_{ij}$  : Distância entre os vértices  $ij$ ;

$t_{ij}$  : Tempo entre os vértices  $ij$  ;

$D_{ij}$  : Demanda entre os vértice  $ij$ ;

$T$  : Instante da simulação.

$Pd$  : Peso atribuído a distância entre os vértices  $ij$ ;

$Pt$  : Peso atribuído ao tempo entre os vértices  $ij$  ;

$PD$  : Peso atribuído a demanda entre os vértices  $ij$  ;

## 5.2 Roteiro Experimental

Para avaliar a proposta por diferentes perspectivas elaboramos dois cenários distintos para que desta forma, fosse possível aplicar as funções de custo e de preço e gerar informações para a análise da performance da heurística.

Os cenários, em resumo, são modelos com as configurações físicas da planta fabril e parâmetros do mercado virtual de rotas. Os características de cada cenário são definidas para atender seu objetivo específico.

### 5.2.1 Cenário 1

Cenário idealizado para possibilitar que os robôs utilizem rotas compartilhando trechos e desta forma, gerando uma superposição e por consequência um grande número de rotas usáveis. A estrutura do cenário é composta um deposito central, uma grade de trechos que interliga os elementos ao deposito e um cliente. O objetivo central desta estrutura é avaliar a performance da heurística no quesito distância percorrida pelos robôs. Na figura 25 são apresentados os dados do cenário tais, como preço inicial, distância e sentido dos trechos.

- Planta fabril:

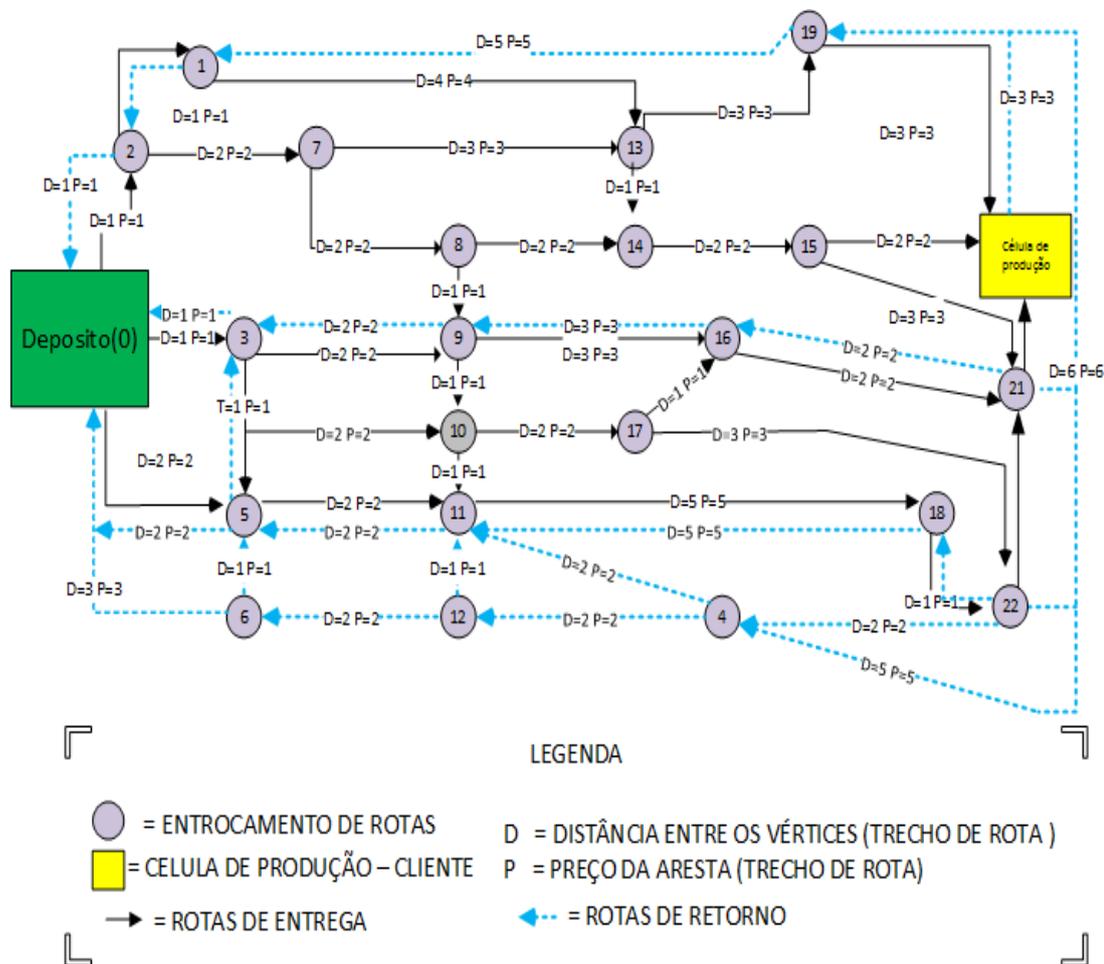


Figura 25 – Planta Fabril Rotas diversificadas

Fonte: Desenvolvido pelo autor

No intuito estabelecer um padrão para as simulações fixamos os seguintes parâmetros mercadológicos:

- Quantidade de robôs: [160]
- Número de iterações: [1500]
- Quantidade de requisições: [1000]

Entretanto, as variáveis utilizadas na atualização dos preços e dos custos são modificadas conforme tabela a seguir:

- Coeficiente de valorização -  $\alpha$

1,0	1,1	1,15	1,2	1,25	1,3	1,35	1,4	1,45	1,5	1,55	1,6	1,65	1,7
-----	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----

- Coeficiente de desvalorização -  $\beta$

1,0	0,9	0,85	0,8	0,75	0,7	0,65	0,6	0,55	0,5	0,45	0,4	0,35	0,3
-----	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----

- Peso Atribuídos  $PD$

01	02	03	04	05	06	07	08	09
----	----	----	----	----	----	----	----	----

No experimento é realizada uma simulação de atendimento a mil requisições de entrega para cada valores de  $\alpha$ ,  $\beta$  e  $PD$  com as suas respectivas funções.

### 5.2.1.1 Métricas avaliativas - Cenário 01

Conforme exposto anteriormente, o cenário 01 foi idealizado para possibilitar a avaliação das rotas tendo como parâmetro as distâncias percorridas pelos robôs. definimos as seguintes métricas estatísticas para a avaliar a proposta. Posteriormente compararemos os resultados da funções preço em detrimento a função curto.

#### 5.2.1.1.1 Somatória das distâncias

Métrica que demonstra a somatória das distâncias percorridas por todos os robôs durante a realização das entregas na simulação;

$$SD = \sum_{i=1}^N Dr_i \quad (5.5)$$

Sendo,

SD : Somatório das distâncias de todos as rotas, desde da primeira posição  $Dr = 1$  até a posição  $N$ ;

$Dr_i$  : Distâncias da rota  $i$ ;

$N$  : Quantidade de rotas;

#### 5.2.1.1.2 Desvio Médio

O desvio médio é definido como média aritmética dos desvios de cada elemento da série para a média. Utilizamos para expressar o desvio médio das distâncias das rotas em torno da média de todas as distâncias das rotas da simulação.

$$DM = \frac{\sum_{i=1}^N (Dr_i - \bar{D})}{N} \quad (5.6)$$

DM : Desvio médio das distâncias de todas as rotas.

$Dr_i$  : Distâncias da rota  $i$ ;

$N$  : Quantidade de rotas;

$\bar{D}$  : Média aritmética das distâncias das rotas;

#### 5.2.1.1.3 Desvio padrão populacional

O desvio padrão populacional trata-se de uma medida que expressa o grau de dispersão de um conjunto de dados. Avaliaremos por meio dessa métrica as distâncias percorridas pelos robôs investigando o quanto é uniforme. Quanto mais próximo de 0 for o desvio padrão, mais homogêneo são os dados.

$$DP = \sqrt{\frac{\sum_{i=1}^N (D_i - \bar{D})^2}{N}} \quad (5.7)$$

Sendo,

$DP$  : Desvio Padrão populacional das distâncias das rotas do conjunto de rotas;

$\sum_{i=1}^N$  : Somatório das distâncias de todas as rotas;

$D_i$  : Distância da rota  $i$  no conjunto de rotas;

$\bar{D}$  : Média aritmética das distâncias das rotas

$N$  : Quantidade de rotas;

## 5.2.2 Cenário 2

Cenário idealizado com o intuito de avaliar o balanceamento das rotas, isto é, como ocorre a distribuição dos robôs na planta fabril durante o processo de entrega dos insumos. Sendo Composto por um deposito central, uma grade de com sete rotas fixas e independentes e uma célula de produção. A implementação da rotas tem a seguinte lógica: a segunda rota tem o dobro do tamanho da primeira e a terceira o dobro do tamanho da segunda e assim assim sucessivamente até sétima. Observa-se na figura 26 os dados do cenário 02.

- Planta fabril:

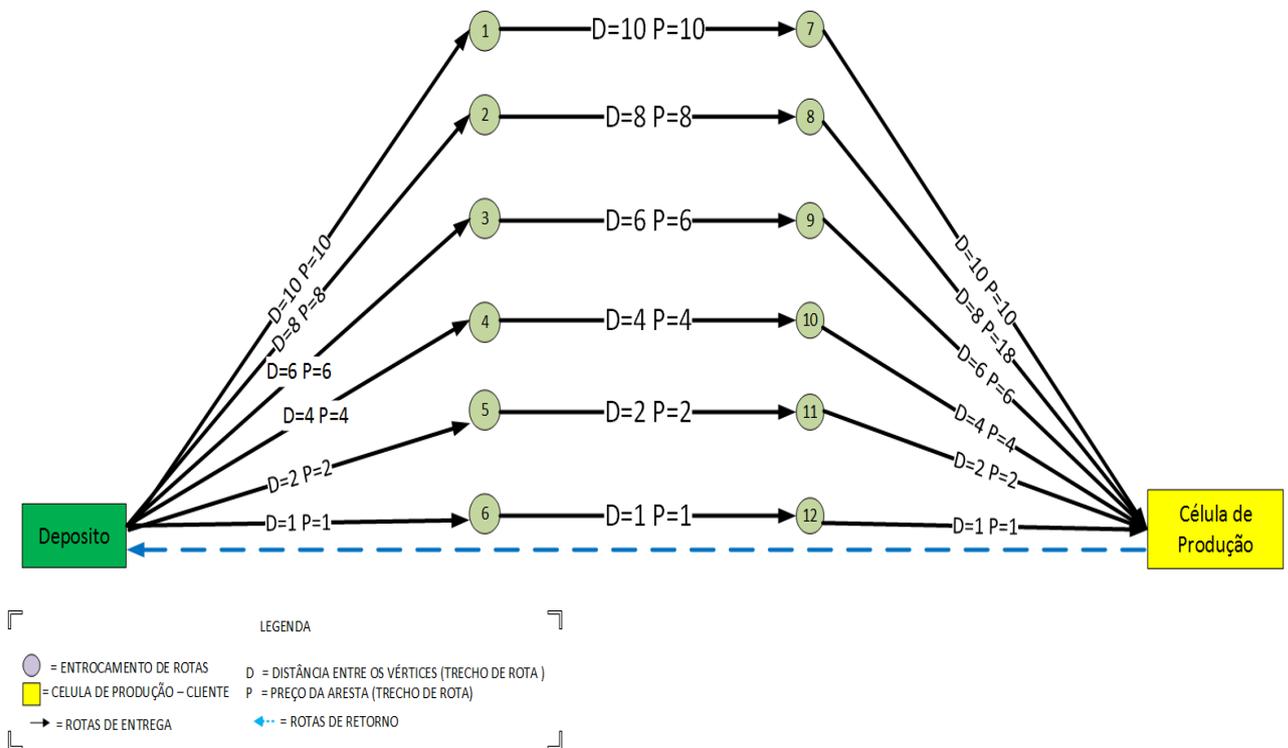


Figura 26 – Planta Fabril Rotas Fixas

Fonte: Desenvolvido pelo autor

No intuito estabelecer um padrão para as simulações fixamos os seguintes parâmetros mercadológicos:

- Quantidade de robôs: [160]
- Número de inteirações: [1500]
- Quantidade de requisições: [1000]

Entretanto, as variáveis utilizadas na atualização dos preços e dos custos são modificadas conforme tabela a seguir:

- Coeficiente de valorização -  $\alpha$

1,0	1,1	1,15	1,2	1,25	1,3	1,35	1,4	1,45	1,5	1,55	1,6	1,65	1,7
-----	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----

- Coeficiente de desvalorização -  $\beta$

1,0	0,9	0,85	0,8	0,75	0,7	0,65	0,6	0,55	0,5	0,45	0,4	0,35	0,3
-----	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----

- Peso Atribuídos  $PD$

01	02	03	04	05	06	07	08	09
----	----	----	----	----	----	----	----	----

No experimento é realizada uma simulação de atendimento a mil requisições de entrega para cada valores de  $\alpha$  ,  $\beta$  e  $PD$  utilizando as suas respectivas funções.

### 5.2.2.1 Métrica avaliativa - Cenário 02

Conforme exposto anteriormente, o cenário 02 foi idealizado para possibilitar a avaliação do balanceamento das rotas tendo como parâmetro a distribuição dos robôs. Define-se a seguinte métricas para a avaliar esta características:

#### 5.2.2.1.1 Densidade de utilização das rotas

O objetivo é mensurar o quanto uma determinada rota é utilizada em detrimento as demais rotas durante o processo de entrega dos insumos. Estabelecemos para atender este objetivo a seguinte métrica:

$$DU = \frac{DR}{\sum_{i=1}^N DR_i} \cdot 100\% \quad (5.8)$$

Sendo,

$\sum_{i=1}^N DR_i$  : Somatório número de robôs que utilizaram determinada rotas, desde da primeira posição  $i = 1$  até a posição  $N$ ;

$DR_i$  : Quantidade Demanda da rota  $i$  no conjunto de rotas;

$DD$  : Densidade de demanda

$N$  : Quantidade de rotas;

As métricas avaliativas são individuais de cada cenário e suas resultados são contabilizados para posteriormente serem plotados e anualizados.

### 5.2.3 Roteiro da Simulação

O comportamento e as interações dos agentes econômicos nas simulações dos experimentos seguem as seguintes etapas:

**Setup** - Etapa que antecede as simulações na qual são definidos os parâmetros mercadológicos e estruturais dos cenários:

- Quantidade de Robôs;
- Número de iterações;
- Coeficiente  $\beta$  de degradação do preço no tempo;
- Coeficiente  $\alpha$  de acréscimo do preço no tempo;
- Grafo com as informações físicas da planta fabril.

**Etapa 1** - A classe PCP lê os parâmetros inserido no Setup e mediante as informações, inclui as requisição de entrega na programação dos robôs.

**Etapa 2** - Os robôs aguardam as requisições no estado denominado desocupado e conforme as requisições são inseridas na programação, retiram para atendê-las. Solicitam do vendedor os preços de todos os trechos que formam a grade que interligam as células de produção.

**Etapa 3** - O vendedor realiza uma leitura do estado atual da grade de trechos observando o histórico de uso, as características físicas e quais preços foram utilizados nas vendas anteriores e munido desta informações formula o preço para cada aresta da grade tendo no final um grafo de preço que é repassado ao robô.

**Etapa 4** - Munido do grafo de preços, o robô calcula por meio de um algoritmo de Caminho Mínimo [**Dijkstra**] qual rota apresenta o preço mínimo. Após, a seleção da rota, o robô informa ao vendedor quais arestas utilizará para atender sua solicitação.

**Etapa 5** - O vendedor recebe a informação de qual rota foi selecionada e atualiza os preços das arestas da planta fabril.

**Etapa 6** – Após selecionar a rota, o robô modifica seu estado para DeslocamentoCliente e inicia seu percurso pela primeira aresta com que compõe a rota selecionada. A cada interação o robô se desloca um passo e avalia a sua localização até que esteja no destino e atualize seu estado para EntregandoInsumos. Aguarda por duas interações, calcula a rota de retorno a base e modifica o estado para Retornobase e inicia seu percurso pela primeira aresta da rota de retorno. Avalia a sua localização até que esteja na base e atualize seu estado para desocupado. Após o retorno a base o robô modifica seu estado para desocupado. O ciclo repete-se até que um dos critérios de interrupção seja alcançado.

**Etapa 7** – Os critérios de interrupção da simulação são a conclusão de todas as entregas ou a execução de todas as interações, ambos os critérios são repassados na etapa de setup. Ao finalizarmos as entregas ou realizarmos todas as interações dois relatórios são gerados com os dados da simulação.

### 5.3 Ambiente computacional

Todos os nossos experimentos são executados em um laptop Hewlett - Packard Company HP, com processador Intel(R) Core(TM) i54200M CPU @ 2.13GHz e 8GB de memória RAM. O sistema operacional utilizado é o Windows 7 professional , 64 bit. Os códigos foram implementados usando a linguagem de programação Java utilizando o IDE Netbeans, versão 8.2

### 5.4 Resultados

Uma vez definidos os experimentos com seus cenários levantou-se as curvas características das funções para cada métrica afim de avaliar a heurística baseada em mercado em detrimento ao modelo tradicional baseado em custos.

### 5.4.1 Coeficiente de valorização $\alpha$ verso deslocamento dos robôs - Função Preço

Na construção deste experimento fixamos  $\beta$  em  $[0,90]$  fato que implica em uma desvalorização de 10% no preço das arestas não utilizadas a cada atualização do mercado. As simulações foram realizadas com o coeficiente de valorização  $\alpha$  variando no intervalo de  $[1,0, 1,75]$ . Todas as simulações tiveram como base o cenário 1. Na Figura 27 pode-se observar o comportamento do sistema mediante as variações de  $\alpha$ .

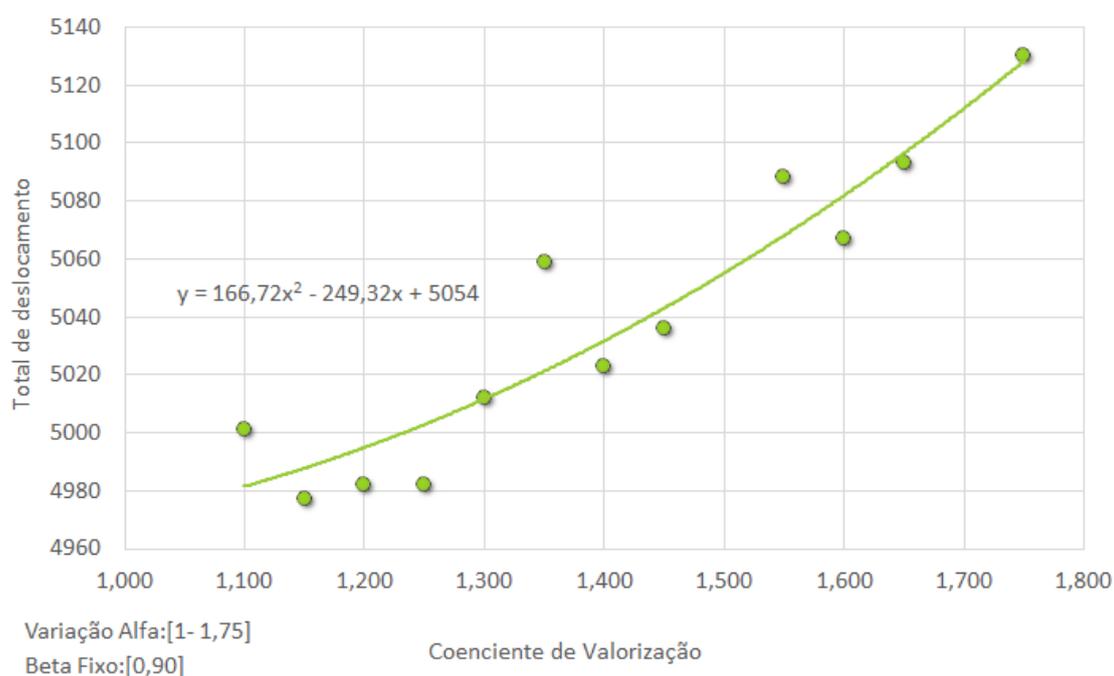


Figura 27 – Relação entre  $\alpha$  e deslocamento dos robôs

**Fonte:** Desenvolvido pelo autor

Nota-se que o crescimento do valor de  $\alpha$  resulta no aumento no descolamento total dos robôs, isto ocorre porque quando  $\alpha$  tem um valor elevado, o preço de determinada aresta cresce rapidamente quando utilizada fazendo com que rotas mais longas e, portanto, menos atrativas, sejam utilizadas. Deduzimos que se trata de uma relação diretamente proporcional não linear.

### 5.4.2 Peso da demanda x deslocamento dos robôs - Função Custo

No desenvolvimento deste experimento fixamos os pesos da distância e do tempo em 2 e variamos o valor atribuído ao peso da demanda no intervalo de [1;9]. As simulações são realizadas no cenário 1. Nota-se que o deslocamento é diretamente proporcional ao peso da demanda.

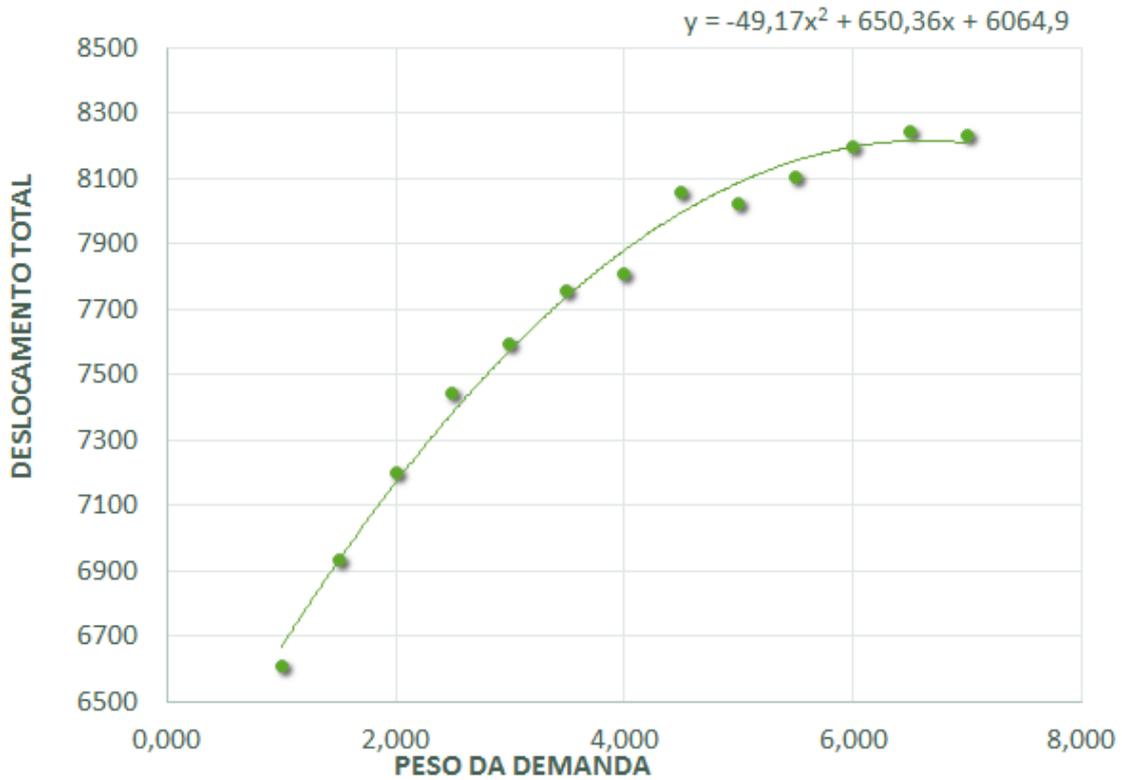


Figura 28 – Peso demanda x deslocamento robôs - Função Custo

**Fonte:** Desenvolvido pelo autor

Observa-se na curva resultante plotada na Figura 28, que a relação entre o peso da demanda e o total de deslocamento não é linear. Conforme o peso atribuído a demanda aumenta há um crescimento significativo no total do deslocamento dos robôs. Conclui-se que o comportamento é similar ao encontrado quando utilizamos a função preço ao variarmos o coeficiente de valorização  $\alpha$ .

### 5.4.3 Coeficientes x Desvio médio das distâncias das rotas - Função Preço

Este experimento tem como objetivo avaliar o comportamento das rotas no quesito distâncias percorridas observando os desvios médios das rotas mediante a variação dos coeficientes de atualização dos preços  $\alpha$  e  $\beta$ . A base para a simulação é o cenário 1. Nota-se na Figura 29 que o aumento na diferença absoluta entre os coeficientes geram uma redução no desvio médio das distâncias das rotas o que demonstra que as rotas geradas tem o tamanho mais próximos da media das distâncias da amostra.

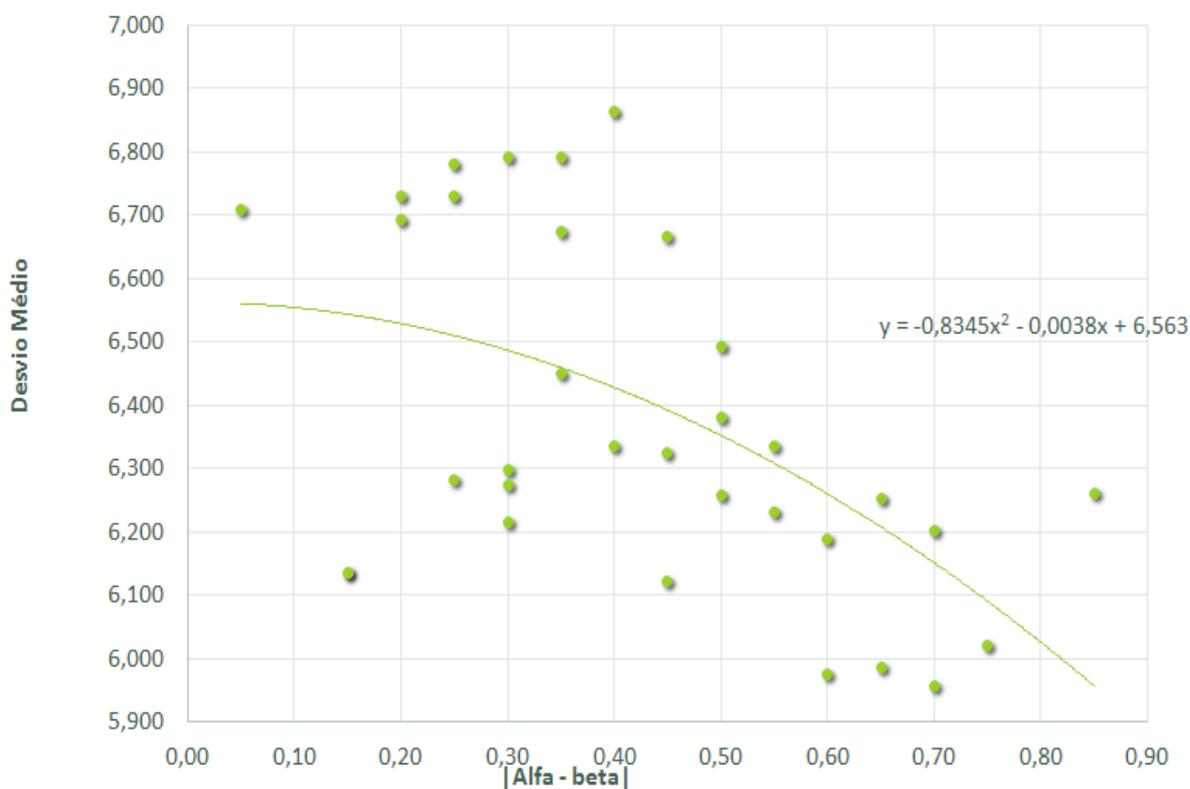


Figura 29 – Desvio Médio - Função Preço

**Fonte:** Desenvolvido pelo autor

Conclui-se que quanto maior for a diferença absoluta entre os coeficientes  $\alpha$  e  $\beta$  mais as distâncias das rotas estão agrupadas entorno do valor médio da amostra.

#### 5.4.4 Peso demanda x Desvio médio das rotas - Função Custo

Avalia-se neste experimento a relação entre a variação do peso atribuído à variável demanda em detrimento aos desvios médios das distâncias das rotas. Para tal, adotamos o seguinte procedimento: fixamos o peso da distância e do tempo em 2 e variamos o valor atribuído ao peso da demanda no intervalo de [01 , 08].

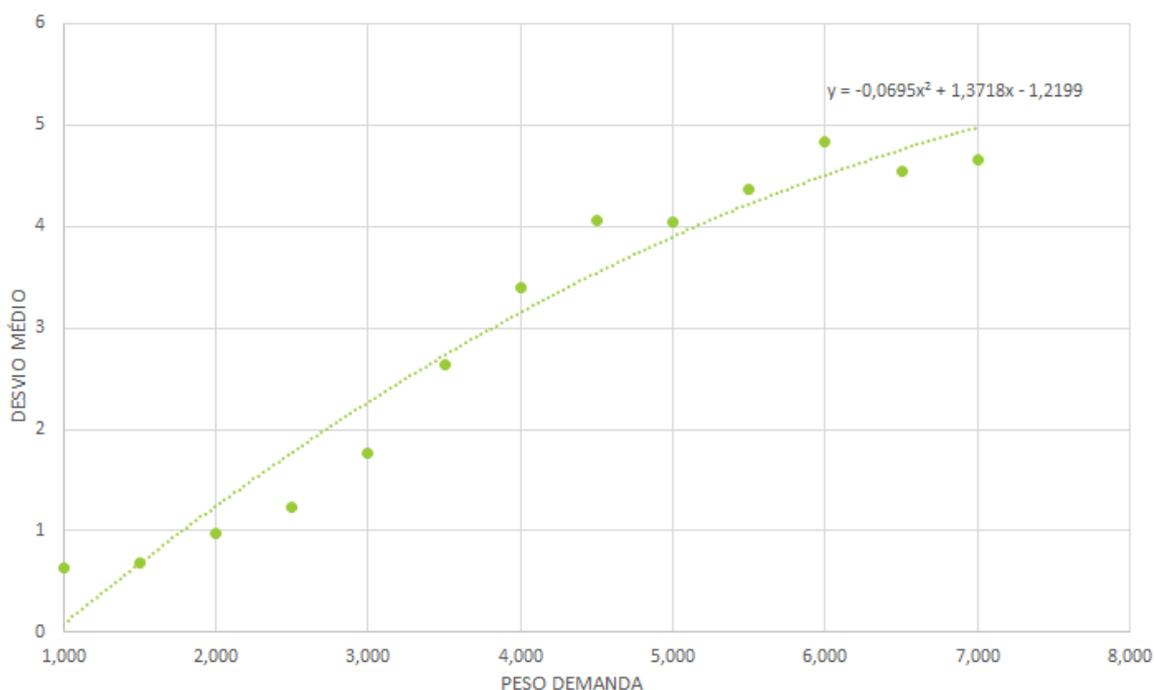


Figura 30 – Desvio Médio - Função Custo

**Fonte:** Desenvolvido pelo autor

Observa-se na Figura 30 que o desvio médio para a função custo apresentou resultados diretamente proporcionais ao peso da demanda. Desta forma, conforme o valor atribuído ao peso da demanda cresce ocorre um aumento no desvio médio padrão das distâncias das rotas de forma quase linear, comportamento contrário ao encontrado na função preço. Conclui-se que aos variarmos positivamente o peso atribuído à demanda as distâncias das rotas ficam menos agrupadas afastando-se da média da amostra.

### 5.4.5 Coeficientes x Desvio padrão das distâncias das rotas-Função Preço

Avalia-se, no experimento, a relação entre o desvio padrão amostral das distâncias das rotas em relação aos coeficientes de atualização dos preços no mercado, para tanto altera-se os valores da diferença absoluta entre os coeficientes e observa-se o comportamento do desvio padrão amostral das distâncias.

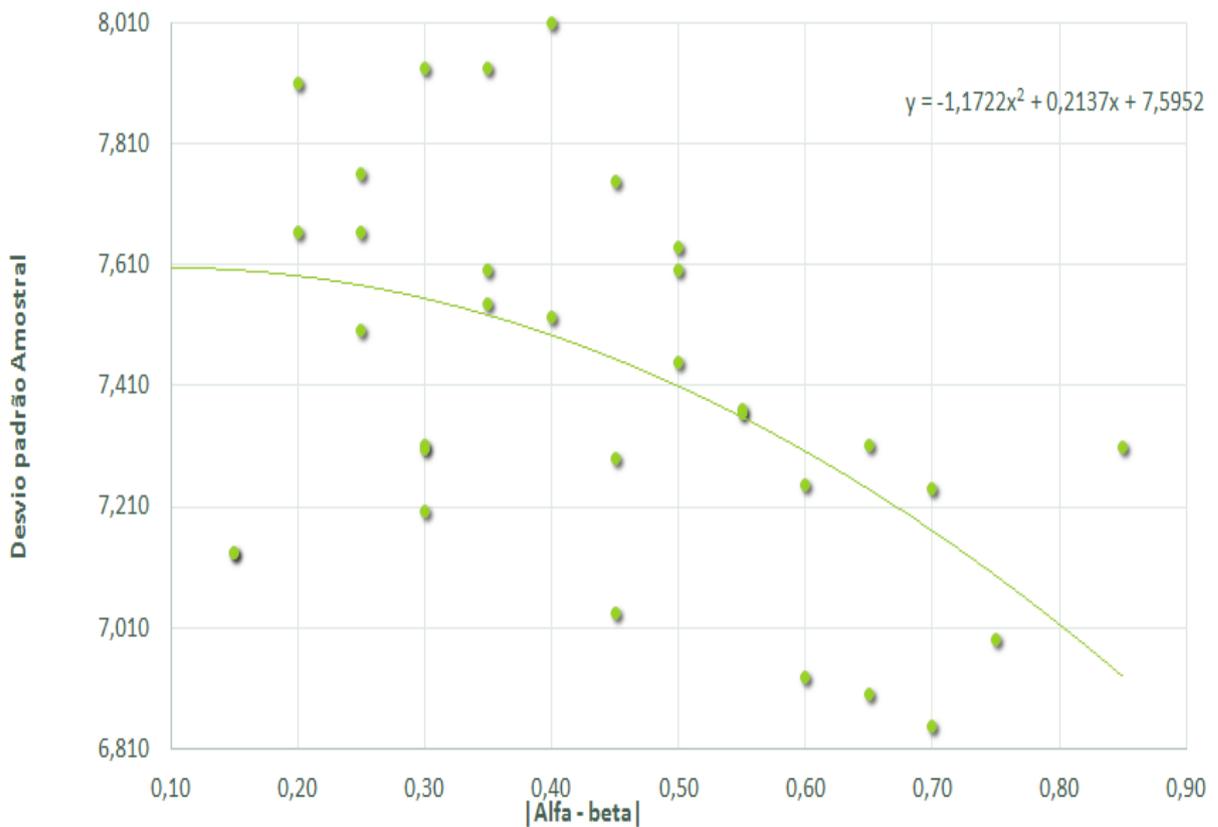


Figura 31 – Desvio Padrão - Função Preço

**Fonte:** Desenvolvido pelo autor

Na figura 31 é possível observar que conforme os valores da diferença absoluta entre os coeficientes aumentam o desvio padrão amostral diminui e com isto, conclui-se que a variabilidade das distâncias reduz fato que produz um agrupamento das rotas mais próximo da média.

### 5.4.6 Avaliação dos densidade de rotas mediante a variação dos coeficientes de atualização de preços - Função Preço

Este experimento tem o objetivo fomentar a avaliação do processo de distribuição dos robôs na planta fabril durante a simulação de entrega dos insumos. Sendo realizado tendo no cenário 02.

Conforme exposto no item 5.2.2, as rotas do cenário 02 são fixas, não havendo a possibilidade de superposição de trechos fato que viabiliza a observação da densidade de uso das rotas.

Durante as simulações foram adotados os seguintes parâmetros: o coeficiente  $\alpha$  foi fixado em 1.001 e variou-se o coeficiente  $\beta$  no intervalo de [0,99 - 0,93] ao passo de 0,01 utilizando a função preço. A cada passo, foi realizada uma simulação de atendimento a mil entregas de insumos. No gráfico da figura 32 são plotados no eixo  $y$  o percentual de uso de cada rota para cada variação de  $\beta$  e no eixo  $x$  estão dispostas as rotas fixas.

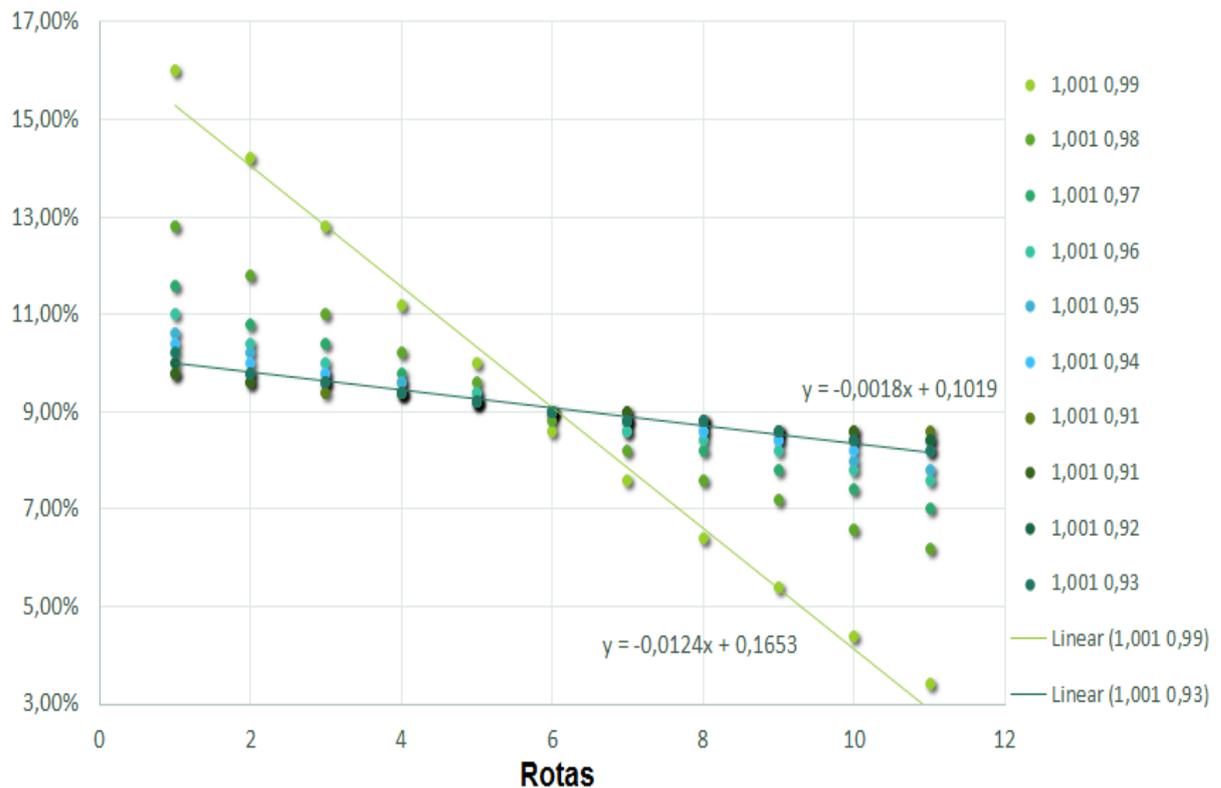


Figura 32 – Distribuição das rotas fixas - Função Preço

Fonte: Desenvolvido pelo autor

Observando o gráfico pode-se concluir que conforme o valor de  $\beta$  aumenta ocorre um espalhamento dos robôs na planta fabril e rotas menos vantajosas são utilizadas ao ponto de evitar congestionamento nos trechos mais curtos. Por conseguinte, podemos relacionar o valor de  $\beta$  o controle de espalhamento dos robôs.

### 5.4.7 Avaliação dos densidade de rotas mediante a variação peso da demanda - Função Custos

Neste experimento foi realizado um processo similar aos da seção 5.4.6 entretanto, avalia-se a distribuição dos robôs na planta fabril mediante a variações nos pesos da variável demanda presente no cálculo da função custos.

Utiliza-se os seguintes parâmetros: fixamos o peso 2 para a variável distância e tempo e variamos a demanda no intervalo [01 , 11] ao passo de 1. A cada passo foi realizada uma simulação de atendimento a mil entrega de insumos e no gráfico da Figura 33 são plotados no eixo *y* os percentuais de uso de cada rota para cada variação da *demand* e no eixo *x* estão dispostas as rotas fixas.

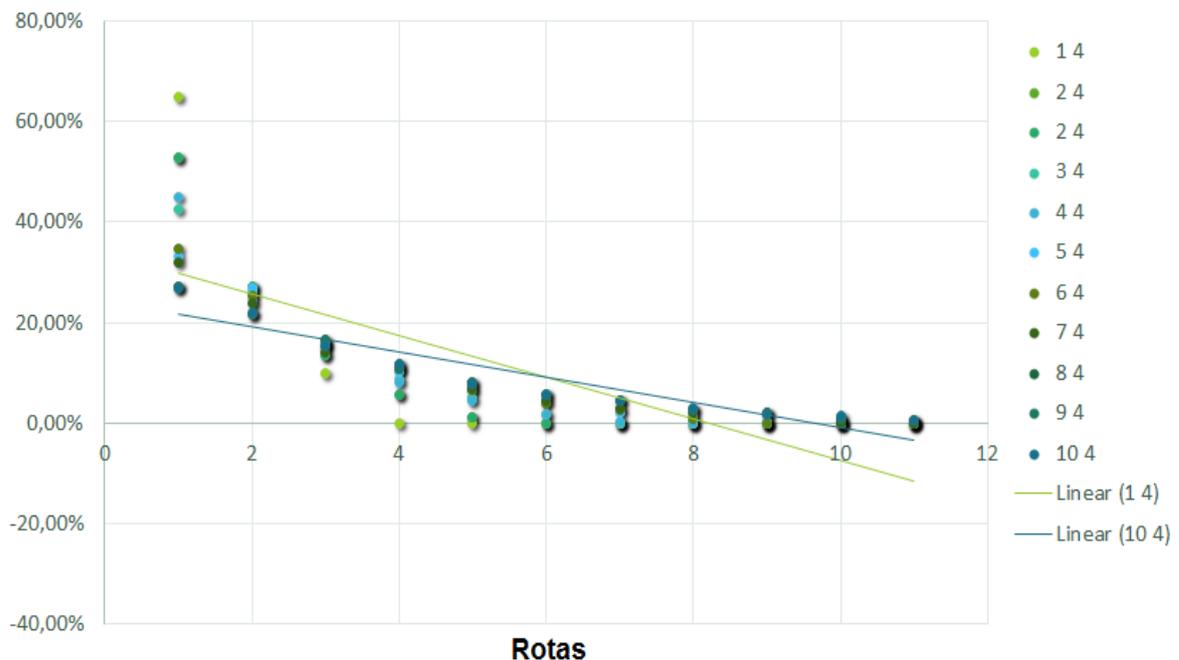


Figura 33 – Distribuição das rotas fixas - Função Custo

Fonte: Desenvolvido pelo autor

Observando o gráfico da figura 33 conclui-se que conforme o peso atribuído a demanda aumenta ocorre um espalhamento dos robôs na planta fabril de tal forma, que as rotas menos vantajosas são utilizadas. Quando comparamos os resultados obtidos pela função custo em detrimento a função preço nota-se que ao aplicarmos a função preço há um controle mais gradativo do espalhamento das rotas na planta fabril.

## 6 Análise e Conclusão

Nesta dissertação foi desenvolvida uma heurística baseada nos conceitos de mercado e processo de formação de preço para a geração de rotas. Sendo o mercado um conceito abstrato, somente é possível analisá-lo pela avaliação das interações dos agentes econômicos que o compõem. Assim, para que fosse possível utilizar os conceitos de mercado, se fez necessário a implementação de um mercado virtual de rotas e a simulação de um espaço fabril de entrega de insumos.

Na modelagem do problema não houve a inclusão de restrições para controle de fluxo e balanceamento de rotas pois, projetava-se avaliar se ao adotarmos a variável preço como métrica na seleção das rotas e implementássemos uma estrutura de mercado, se isto seria suficiente para a limitação do fluxo e o balanceamento das rotas.

Os resultados apresentados capítulo 5.4 demonstram que houve o balanceamento das rotas dentro da planta fabril o qual pode ser refinado de acordo com a necessidade, por meio da adequação dos valores dos coeficientes  $\alpha$  e  $\beta$  do processo de atualização de preços.

Constata-se que a abordagem utilizando conceitos de mercado é uma alternativa para solução de problemas de alocação de recursos, tendo como principal característica a capacidade de simplificar o processo avaliativo pois, tende a resumir em única variável a complexidade do processo produtivo. A heurística proposta seguiu esta premissa, uma vez que o preço basicamente é definido tendo como métrica apenas as interações entre os agentes econômicos que compõem o mercado. Mesmo considerando que o preço inicial seja composto por características físicas da planta fabril, a partir do início da simulação tais características tendem a ter pouca relevância na seleção das rotas e, por ser o preço apenas um número, a composição da função objetivo do problema em questão é simplificada consideravelmente.

Nota-se que a abordagem heurística proposta, que utiliza o mercado de como modelo, é uma alternativa para solução de problemas de alocação de recursos, uma vez que tem como principal característica é a capacidade de simplificação do processo avaliativo pois, resumir em única variável, o preço, a complexidade do processo.

Em relação à contribuição a heurística podemos afirmar que mostrou-se funcional e aplicável na geração de rotas, mesmo sendo uma proposta simples, apresentou bom desempenho nos experimentos realizados de forma que, os testes realizados foram no simulador do mercado virtual. Em síntese, podemos inferir que a heurística atende à proposta deste trabalho, que foi a implementação do paradigma de linguagem orientada a objeto. Aplicada distribuição de insumos em um espaço fabril.

## 6.1 Propostas para trabalhos Futuros

Quanto às principais propostas para desenvolvimentos futuros relacionados aos trabalhos nesta dissertação, podemos citar:

- Implementar uma heurística utilizando o conceito multi-agentes com a estrutura de processamento descentralizada por meio do processamento distribuído para a formação de um mercado para negociação e alocação de recursos escassos com vários vendedores negociando;
- Desenvolver uma função custo para servi de comparação que não considere a velocidade dos robôs constante, vinculado-a ao quantitativo de robôs utilizando a mesma rota em determinado instante.
- Implementar um controle dos coeficientes de atualização dos preço no intuito de modelar o comportamento do processo de alocação dos recursos que através da análise das rotas na saída do simulado.
- Desenvolvimento de um sistema para visualização do funcionamento do simulador para construção de uma melhor compreensão do funcionamento do mercado virtual;

# Referências

BALLOU, R. H. *Gerenciamento da cadeia de suprimentos: planejamento, organização e logística empresarial*. [S.l.]: Bookman, 2001. Citado na página 30.

BOCK, F.; KANTNER, H.; HAYNES, J. *An algorithm (the r-th best path algorithm) for finding and ranking paths through a network*. [S.l.]: Armour Research Foundation, 1957. Citado na página 15.

CACCALANO, L.; CUNHA, C. B. d. Routing of vehicles for material delivery to assembly lines. *Gestão e Produção*, SciELO Brasil, v. 22, n. 4, p. 846–860, 2015. Citado 2 vezes nas páginas 20 e 21.

CLARKE, G.; WRIGHT, J. W. Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, *Inform*s, v. 12, n. 4, p. 568–581, 1964. Citado na página 21.

CLARKE, S.; KRIKORIAN, A.; RAUSEN, J. Computing the n best loopless paths in a network. *Journal of the Society for Industrial and Applied Mathematics*, SIAM, v. 11, n. 4, p. 1096–1102, 1963. Citado na página 15.

COCO, A. A.; NORONHA, T. F. de; SANTOS, A. C. Heurística de refinamento para o problema de caminho mais curto robusto. *Proceedings of the XLVI Simpósio Brasileiro de Pesquisa Operacional (SBPO)*, Salvador, Brasil, p. 1913–1923, 2014. Citado na página 16.

CURY, J. E. R. Teoria de controle supervisorio de sistemas a eventos discretos. *V Simpósio Brasileiro de Automação Inteligente (Minicurso)*, 2001. Citado na página 23.

DANTZIG, G. B.; RAMSER, J. H. The truck dispatching problem. *Management science*, *Inform*s, v. 6, n. 1, p. 80–91, 1959. Citado na página 21.

DENG, Y. et al. Fuzzy dijkstra algorithm for shortest path problem under uncertain environment. *Applied Soft Computing*, v. 12, n. 3, p. 1231 – 1237, 2012. ISSN 1568-4946. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1568494611004376>>. Citado na página 13.

DIJKSTRA, E. W. A note on two problems in connexion with graphs. *Numerische Mathematik*, v. 1, n. 1, p. 269–271, Dec 1959. ISSN 0945-3245. Disponível em: <<https://doi.org/10.1007/BF01386390>>. Citado 3 vezes nas páginas 14, 15 e 17.

EKSIOGLU, B.; VURAL, A. V.; REISMAN, A. The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering*, Elsevier, v. 57, n. 4, p. 1472–1483, 2009. Citado 2 vezes nas páginas 21 e 22.

FEOFILOFF, P.; KOHAYAKAWA, Y.; WAKABAYASHI, Y. Uma introdução sucinta á teoria dos grafos. 2011. Citado 3 vezes nas páginas 4, 5 e 7.

GAREY, M. R.; JOHNSON, D. S. A guide to the theory of np-completeness. *WH Freeman, New York*, v. 70, 1979. Citado na página 10.

- GOLDBARG, M. C. *Otimização combinatória e programação linear: modelos e algoritmos*. Brasil, Rio de Janeiro: Elsevier, 2012. ISBN 978-85-352-1520-5. Citado 11 vezes nas páginas 4, 5, 6, 7, 11, 14, 15, 17, 19, 21 e 22.
- GOMES, M.; VALENTE, D.; FREITAS, R. de. Algoritmos online para o problema do caminho mínimo dinâmico e suas aplicações no tráfego urbano. Citado na página 13.
- HANDLER, G. Y.; ZANG, I. A dual algorithm for the constrained shortest path problem. *Networks*, Wiley Online Library, v. 10, n. 4, p. 293–309, 1980. Citado na página 14.
- HERNANDES, F.; BERTON, L.; CASTANHO, M. J. d. P. O problema de caminho mínimo com incertezas e restrições de tempo. *Pesquisa Operacional*, SciELO Brasil, v. 29, n. 2, p. 471–488, 2009. Citado na página 16.
- IORIO, U. J. Dez lições fundamentais de economia austríaca. *São Paulo: Instituto Ludwig von Mises Brasil*, 2013. Citado na página 24.
- IRVING, F. The nature of capital and income. *Macmillan*, p. 1–21, 1906. Citado na página 25.
- JAEGERE, N. D.; DEFRAEYE, M.; NIEUWENHUYSE, I. V. The vehicle routing problem: state of the art classification and review. KU Leuven-Faculty of Economics and Business, 2014. Citado na página 22.
- JAFFE, J. M. Algorithms for finding paths with multiple constraints. *Networks*, Wiley Online Library, v. 14, n. 1, p. 95–116, 1984. Citado na página 14.
- JOKSCH, H. C. The shortest route problem with constraints. *Journal of Mathematical analysis and applications*, Elsevier, v. 14, n. 2, p. 191–197, 1966. Citado na página 14.
- LARSON, R. C.; ODONI, A. R. *Urban operations research*. [S.l.: s.n.], 1981. Citado na página 18.
- LAWLER, E. L. A procedure for computing the k best solutions to discrete optimization problems and its application to the shortest path problem. *Management science*, INFORMS, v. 18, n. 7, p. 401–405, 1972. Citado na página 15.
- MANKIW, B. B. Ng introdução á economia: São paulo: Pioneira thomson learning, 2009. pinho, db; vasconcellos, mas de (orgs.). *Manual de Economia: equipe de professores da USP*, v. 5. Citado 2 vezes nas páginas 27 e 28.
- MATAI, R.; SINGH, S.; MITTAL, M. L. Traveling salesman problem: an overview of applications, formulations, and solution approaches. In: *Traveling salesman problem, theory and applications*. [S.l.]: InTech, 2010. Citado na página 20.
- RUSIC, j. L. *The Bottleneck traveling salesman Problem And Some Variations*. Tese (Doutorado) — Universidade de São Paulo, Faculdade de Ciências e Tecnologia, janeiro 2015. Citado na página 20.
- SARAIVA, A. Microeconomia. Saraiva, 2011. Citado 2 vezes nas páginas 2 e 27.
- UCHOA, J. S. *Caminhos mínimos com recursos limitados*. Tese (Doutorado) — Universidade de São Paulo, Faculdade de Ciências e Tecnologia, Set 2012. Citado 3 vezes nas páginas 11, 14 e 15.

VASCONCELLOS, M. A. S.; GARCIA, M. E. Fundamentos de economia. *São Paulo: Saraiva*, v. 2, 1998. Citado 4 vezes nas páginas 24, 25, 28 e 29.

YEN, J. Y. Finding the k shortest loopless paths in a network. *management Science, INFORMS*, v. 17, n. 11, p. 712–716, 1971. Citado na página 15.