Pontifícia Universidade Católica
do Rio de Janeiro

Fabio Trindade Duque Estrada Regis

IMPLEMENTAÇÃO DE SOLUÇÕES DE PROBLEMAS LOGÍSTICOS EM UM SISTEMA DE INFORMAÇÃO GEOGRÁFICA PARA APOIO AO ENSINO

Dissertação apresentada como requisito parcial para obtenção do título de Mestre (opção profissional) pelo Programa de Pós-Graduação em Engenharia de Produção da PUC-Rio.

Orientador: Prof. José Eugênio Leal
Co-Orientador: Prof. Orivalde Soares da Silva Júnior

Rio de Janeiro
Abril de 2014

Pontifícia Universidade Católica
do Rio de Janeiro

FABIO TRINDADE DUQUE ESTRADA REGIS

IMPLEMENTAÇÃO DE SOLUÇÕES DE PROBLEMAS LOGÍSTICOS EM UM SISTEMA DE INFORMAÇÃO GEOGRÁFICA PARA APOIO AO ENSINO.

Dissertação apresentada como requisito parcial para obtenção do título de Mestre (opção profissional) pelo Programa de Pós-Graduação em Engenharia de Produção da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. José Eugênio Leal
Orientador e Presidente
Departamento de Engenharia Industrial - PUC-Rio

Prof. Orivalde Soares da Silva Júnior
Co-Orientador
Tecnologia em Sistemas de Comando e Controle

Prof. Ulf Bergman
Tecnologia em Sistemas de Comando e Controle

Prof. Hugo Miguel Varela Repolho
Departamento de Engenharia Industrial - PUC-Rio

Prof. José Eugênio Leal
Coordenador Setorial do Centro Técnico Científico - PUC-Rio

Rio de Janeiro, 16 de Abril de 2014

Fabio Trindade Duque Estrada Regis

Graduou-se em Engenharia de Produção na UVA (Universidade Veiga de Almeida) em 2009. Atuou em diversos projetos em logística no grupo Peugeot Citroën. Pós-graduando em Engenharia de Software com Java no Infnet.

Ficha Catalográfica

Regis, Fábio Trindade Duque Estrada

Implementação de soluções de problemas logísticos em um sistema de informação geográfica para apoio ao ensino / Fábio Trindade Duque Estrada Regis ; orientador: José Eugênio Leal ; co-orientador: Orivalde Soares da Silva Júnior. – 2014.

123 f. : il. (color.) ; 30 cm

Dissertação (mestrado)–Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Industrial, 2014.

CDD:658.5

Aos meus três maiores professores,

minha mãe Marlene, meu pai Wilson e meu irmão Márcio.

Agradecimentos

A minha família por toda a educação, amor e suporte.

Ao orientador, Prof. José Eugênio Leal, por acreditar e apoiar o trabalho desde o início e pela alta qualidade técnica na exposição das aulas durante o mestrado.

Ao coorientador, Orivalde Soares da Silva Júnior, pela paciência de entender o projeto e pelos ensinamentos fundamentais em desenvolvimento de software no OpenJUMP e em Java.

A PUC-Rio e todos os professores do mestrado em Logística pela alta qualidade nas aulas que ampliaram ainda mais minha visão sobre a arte da logística.

Aos meus colegas do mestrado por compartilhar suas experiências e conhecimentos profissionais durante as aulas.

Á Ulf Bergman e Hugo Varella Repolho pela participação na banca examinadora.

Aos amigos Prof. Conrad Peres e Jean Fonseca pelos ensinamentos em Java.

Resumo

Regis, Fábio Trindade Duque Estrada; Leal, José Eugênio. **Implementação de soluções de problemas logísticos em um sistema de informação geográfica para apoio ao ensino**. Rio de Janeiro, 2014. 123p. Dissertação de Mestrado – Departamento de Engenharia Industrial, Pontifícia Universidade Católica do Rio de Janeiro.

Esta dissertação tem como objetivo contribuir ao ensino da logística, através da construção de ferramentas computacionais didáticas e gratuitas, adicionadas a um software do tipo SIG – Sistema de Informação Geográfica, com distribuição livre, permitindo a execução prática de diversos procedimentos importantes para a solução de problemas logísticos por parte de estudantes, técnicos e demais interessados na área de logística de forma gratuita. O conhecimento necessário para a construção de soluções de problemas de gestão logística ou da gestão de uma cadeia de suprimentos são interdisciplinares, envolvendo diferentes áreas como; geografia, informática, matemática, administração, entre outras. Assim, o processo de construção do conhecimento pelos aprendizes em logística torna-se um grande desafio, pois exige diferentes habilidades do aluno que primeiramente, precisa entender os assuntos de cada área de forma separada para poder integrá-los e iniciar a construção do conhecimento. Além do fator da interdisciplinaridade que complica o processo de aprendizado, os recentes avanços na área de computação e de algoritmos para solução de problemas em logística, tem causado um distanciamento dos profissionais entre as soluções disponíveis e a sua implementação prática. Nesse sentido, esta dissertação irá apresentar a implementação de soluções para problemas logísticos através da construção de ferramentas computacionais com fins didáticos desenvolvidas através da linguagem de programação Java que serão adicionadas como *plugin* ao software OpenJUMP.

Palavras-chave

SIG; logística; zoneamento; roteirização; clusters; ponto central.

Abstract

Regis, Fábio Trindade Duque Estrada; Leal, José Eugênio (Advisor). **Implamentation of Logistic Problems Solutions Using Geographic Information System for Teaching Purposes**. Rio de Janeiro, 2014. 123p. MSc. Dissertation – Departamento de Engenharia Industrial, Pontifícia Universidade Católica do Rio de Janeiro.

The objective of this dissertation is to contribute to the teaching of logistics, through the construction of educational and free computational tools, added to a software like GIS - Geographic Information System, with free distribution, allowing the practical implementation of several important procedures for solving logistical problems by students, technicians and others interested in the logistics area for free. The knowledge required for the construction of solutions of problems of logistics management or the management of a supply chain , are interdisciplinary , involving different areas such as ; geography, computer science, mathematics, business administration, among others. Thus, the process of knowledge building by learners in logistics becomes a challenge because it requires different skills that the student first needs to understand the subjects of each area separately to be able to integrate them and begin construction of knowledge. Besides the factor of interdisciplinarity which becomes the learning process more complicate, recent advances in computing and algorithms to solve problems in logistics has caused a distancing of professionals from the available solutions and their practical implementation. In this sense, this dissertation will present the implementation of solutions to logistics problems through the construction of computational tools for didactic purposes developed through the Java programming language that will be added as *plugin* software to OpenJUMP.

Keywords

GIS; logistics; zoning; routing; clusters; central point.

Sumário

Lista de figuras

Onde há amor e sabedoria, não tem temor e nem ignorância.

(Francisco de Assis)

1 Introdução

1.1 O Problema

A formação de qualquer profissional é um dos principais fatores que tem grande influência no nível de competitividade das empresas, consequentemente, do país. Melhorar a educação do profissional fornecendo acesso ao conhecimento e aproximando as teorias aprendidas com a prática, ou seja, a realidade empresarial interessa a todos; faculdades, governo, empresas, etc.

A logística ao longo das últimas décadas construiu um grande arcabouço de teorias e técnicas para resolver grande parte dos problemas enfrentados dentro da área. Em contrapartida, criou-se um distanciamento entre a teoria e a aplicação das técnicas devido à complexidade relativamente alta e a necessidade de ter a capacidade de entender sobre diversas áreas para conseguir a implementação efetiva dessas técnicas.

Os problemas de localização, roteirização, formação de cluster e formação de zonas, são desafios comuns enfrentados pelos profissionais de logística e ocorrem com alta frequência no dia a dia empresarial devido a sua natureza dinâmica. Empresas estão constantemente revisando e criando novas estratégias para aumentar o valor percebido por seus clientes, que por sua vez tem relação direta na decisão em onde devem localizar suas fábricas, depósitos, lojas, etc. A roteirização é muito sensível a mudanças de fatores como preços, consumo, localização, etc.

Ou seja, esses tipos de problemas são constantes no ambiente empresarial. Por essa razão, o aprendizado das técnicas disponíveis que possam fornecer respostas corretas para cada problema é fundamental para profissional de logística.

1.2 Objetivo Geral

Proporcionar aos estudantes, técnicos e demais interessados na área de logística, uma ferramenta didática gratuita, representada por um sistema computacional que possua funcionalidades com base em técnicas reconhecidas para solucionar problemas logísticos de localização e roteirização, permitindo a aplicação prática da teoria aprendida pelo profissional de logística em formação e contribuindo no seu processo de aprendizado.

1.3 Objetivos Específicos

- Desenvolver um *plugin* na linguagem de programação Java que será adicionado ao software SIG, OpenJUMP*;*
- Implementar as seguintes funcionalidades no *plugin:*
    - *Procedimentos para solução de problemas de localização*
        - Definição do ponto central pela métrica Retangular
            - Método de Fibonacci
            - Método da Derivada
        - Definição do ponto central pela métrica Euclidiana
            - Método de Weisfield
    - *Procedimentos para problemas de formação de cluster*
            - Algoritmo de Kruskal adaptado para formação de vários *Clusters*
    - *Procedimentos para problemas de roteirização*
        - Métodos de Construção de roteiro
            - Método de Clarke e Wright
        - Método de Melhoria de Roteiros
            - Método 2-opt
    - *Procedimentos para solução de problemas de formação de zonas*
        - Métodos de Construção de zonas
            - Método de divisão de regiões em zonas de acordo com nível de serviço

- Método de Clarke e Wright com parâmetro de forma

1.4 Justificativa

NOVAES (2007) cita que a má qualificação dos profissionais de logística resulta em decisões deficientes na logística empresarial, por exemplo, acreditando que a compra de um software de roteirização irá resolver todos os seus problemas de distribuição em sua empresa. Ou seja, além da falta de entendimento do problema existe também a falta do entendimento da solução proposta.

O profissional de logística ao receber uma formação que permita entender o problema pelo ponto de vista acadêmico, traduzir para a realidade empresarial através de cenários, posteriormente aplicar e observar a resolução do problema através ferramentas computacionais, melhora o seu processo de aprendizado e consequentemente sua qualidade profissional.

Os sistemas computacionais atualmente são amplamente aplicados tanto no meio acadêmico, melhorando o processo de aprendizado do profissional, como no meio empresarial, em grande parte representada por soluções em software de empresas especializadas. O profissional de logística bem formando consegue selecionar melhor qual solução a ser implementada em cada momento, provendo respostas eficazes.

CARDOSO (2012) em sua análise sobre a efetividade do método de aprendizado baseado em problemas no ensino da logística declara de forma positiva que trazer problemas o mais próximo possível do mundo real e provocar a resolução por parte dos alunos é benéfico e eficiente. Em consonância com esse contexto de ensino, as ferramentas computacionais propostas nesse trabalho, poderiam ser aplicadas também de forma melhorar o processo de aprendizado do aluno.

A característica aberta e flexível de sistemas computacionais de livre distribuição, como o OpenJUMP*,* que suporta a inclusão de funcionalidades diversas, favorecem a aplicação no meio acadêmico, pois podem ser utilizados, modificados, melhorados e compartilhados sem custo algum.

As funcionalidades desenvolvidas no protótipo, conforme citado anteriormente em objetivos específicos, foram selecionadas com base nas aulas ministradas no próprio curso de mestrado.

1.5 Estrutura do Trabalho

No primeiro capitulo é apresentado como este trabalho está estruturado, seu objetivo, justificativa e limitações.

O segundo capítulo descreve os elementos que fazem parte do contexto dos problemas de localização, roteirização, formação de clusters e formação de zonas. É apresentada uma breve análise sobre a logística nas organizações brasileiras, ensino de logística no Brasil e uma revisão dos principais conceitos relacionados a distribuição física.

O terceiro capítulo aborda alguns conceitos de software de código livre, a construção de sistemas com o paradigma de orientação a objetos, a linguagem de programação Java e a caracterização e relevância do uso de Sistemas de Informação Geográfica pelas empresas. Também se apresenta o software livre OpenJUMP, usado na construção do protótipo.

O quarto capítulo descreve os métodos que foram usados no protótipo para a resolução de problemas de localização, roteirização, formação de clusters e formação de zonas.

O quinto capítulo descreve os principais componentes usados no desenvolvimento do protótipo, lista de requisitos, diagramas de caso de uso e organização do código.

No sexto e último capítulo é apresentada a conclusão sobre o trabalho desenvolvido, assim como algumas recomendações e sugestões para trabalhos futuros.

1.6 Limitações do Trabalho

O presente trabalho não irá revisar, comparar e implementar todas as técnicas disponíveis na literatura para a solução dos problemas logísticos de localização e roteirização.

Este trabalho irá abordar algumas técnicas disponíveis na literatura usadas para a solução de problemas logísticos que são imprescindíveis aos envolvidos com a área de logística.

2 Logística

A quantidade e diversidade de produtos e serviços que surgiram nas últimas décadas foram enormes. Os vários segmentos existentes em qualquer economia; têxtil, eletrônicos, alimentos, energia, e muitos outros, viram a complexidade da sua logística aumentar e, consequentemente, tiveram que enfrentar novos desafios para conseguir comercializar os seus produtos. Como consequência, por exemplo, os produtos têm um ciclo de vida cada vez menor, mudam em suas embalagens, formas, tamanhos, cores, composição, etc. Todas essas mudanças sofridas pelos produtos e a necessidade de resposta num curto espaço de tempo, impactam diretamente a logística dos mesmos aumentando a complexidade de comercialização.

BOWERSOX (2001) descreve a logística como um esforço integrado com o objetivo de ajudar a empresa a criar valor para seus clientes com o menor custo possível. Essa perspectiva está em sintonia com o conceito de que a logística pode gerar uma vantagem competitiva para empresas a partir do momento que as mesmas abordam a logística e a cadeia de suprimentos com uma visão estratégica.

Segundo o *Council of Logistics Management* (CLM);

"Logística é o processo de planejamento, implantação e controle do fluxo eficiente e eficaz de mercadorias, serviços e das informações relativas desde o ponto de origem até o ponto de consumo com o propósito de atender às exigências dos clientes."

De acordo com *Society Of Logistics Engineers* (apud KOBAYASHI, 2000) descreve oito finalidades para a logística:

1. *Right Material* (material correto)

2. *Right Quantity* (na quantidade correta)
3. *Right Quality* (com a qualidade justa)
4. *Right Place* (no lugar correto)
5. *Right Time* (no tempo correto)
6. *Right Method* (com o método correto)
7. *Right Cost* (segundo o custo justo)
8. *Right Impression* (com uma boa impressão)

Há algumas décadas, as empresas eram mais verticalizadas, ou seja, realizavam grande parte ou todas as etapas do processo de produção de um produto, sendo responsável desde a matéria-prima até a venda do produto. Após mudanças na economia, como o crescimento da concorrência, essa estrutura empresarial se modificou (NOVAES 2007).

Em busca de diferenciação e competitividade de um produto, considerando que o mesmo é resultado de uma cadeia de suprimento, é necessário mudar um paradigma no enfoque de gestão, onde antes as empresas buscavam a melhoria somente com enfoque interno, da própria empresa, buscando o ótimo local, agora as empresas devem buscar o ótimo global, através da cooperação e compartilhamento de informações para que o resultado do produto fruto de uma cadeia de suprimento seja maximizado.

Essa mudança de enfoque de gestão e todos os seus problemas intrínsecos são tratados pelo *Supply Chain Management* – SCM (Gerenciamento da Cadeia de Suprimentos) que por sua vez possui um arcabouço de metodologias e técnicas disponíveis.

Definição de SCM segundo Fórum de SCM realizado na *Ohio State University:*

SCM é a integração dos processos industriais e comerciais, partindo do consumidor final e indo até os fornecedores inicias, gerando produtos, serviços e informações que agreguem valor para o cliente.

NOVAES (2007) também define o conceito de cadeia de suprimento com base em STERN *et al.(*1996):

Uma cadeia de suprimentos é constituída por conjuntos de organizações interdependentes envolvidas no processo de tornar o produto ou serviço disponível para uso ou consumo.

CHOPRA (2003) define uma cadeia de suprimentos como todos os estágios envolvidos, direta ou indiretamente, no atendimento de um pedido de um cliente, sendo composta tipicamente por fábricas, depósitos, transportadoras, varejistas, entre outros membros.

Em resumo, o SCM traz o enfoque para todas as empresas da cadeia de suprimento para o consumidor final, exige uma forte cooperação e comunicação entre as empresas, compartilhamento de ganhos, integrações de processos, planejamento estratégico conjunto e muitos componentes que devem ser tratados para obter uma cadeia de suprimentos eficiente.

2.1 Breve análise sobre a logística nas organizações brasileiras

MACHILINE (2011) em seu trabalho que fez uma análise histórica da evolução da logística no Brasil destaca que apesar da defasagem em termos do tempo, o desenvolvimento da logística no Brasil ocorreu de forma parecida com a dos Estados Unidos. No Brasil, entre os anos de 1950 e 1960 o foco era o transporte e o conceito foi ampliado entre as décadas de 1970 e 1980 solidificando o conceito de logística empresarial. O autor destaca a década de 1990 para o Brasil que teve um salto conceitual da visão de logística para a visão de cadeia de suprimentos, expandindo a visão de logística para fora da organização, ou seja, preocupando-se também com a cadeia de fornecedores e de clientes.

Atualmente, o Brasil passa por uma grande necessidade de demanda de infraestrutura logística como: portos, aeroportos, estradas, etc. Essa limitação de capacidade também dificulta a aplicação de boas práticas logísticas por parte das empresas. O governo brasileiro para tentar atrair capital de investimento, lançou um programa de investimentos em logística para através da concessão de incentivos e isenções fiscais, justificativas econômicas, entre outras formas, melhorar a infraestrutura logística no país. Maiores detalhes do programa no site www.logisticabrasil.gov.br.

NOVAES (2007) destaca que a falta de uma estrutura organizacional adequada é um dos limitadores do avanço da logística no Brasil. Um exemplo dessa limitação é quando uma diretoria de logística dentro de uma empresa não tem a mesma autonomia em relação às outras diretorias, ficando submissa as ações tomadas por outras áreas, dificultando a implementação de estratégias para na área.

2.2 Educação em Logística no Brasil

MACHILINE (2011) destaca que durante as últimas décadas do século XX começou a se tornar expressiva a proliferação de literatura técnica na área de logística. Entre elas algumas revistas que hoje possuem grande relevância para a comunidade logística como: Revista Tecnologista (criada em 1995), a Global Comércio Exterior e Logística (criada em 2002) e *Today Logistics and Supply Chain* (criada em 2006).

Ainda nesse período surgiram associações profissionais como a Associação de Logística Brasileira (ASLOG – Criada em 1989) e a Associação Brasileira de Movimentação e Logística (ABML – Criada em 1997). Essas associações de certa forma também ajudaram a fortalecer e difundir o conhecimento em logística no Brasil.

Segue abaixo (Figura 1) a linha do tempo com alguns fatos históricos que dentro do contexto de educação em logística no Brasil:

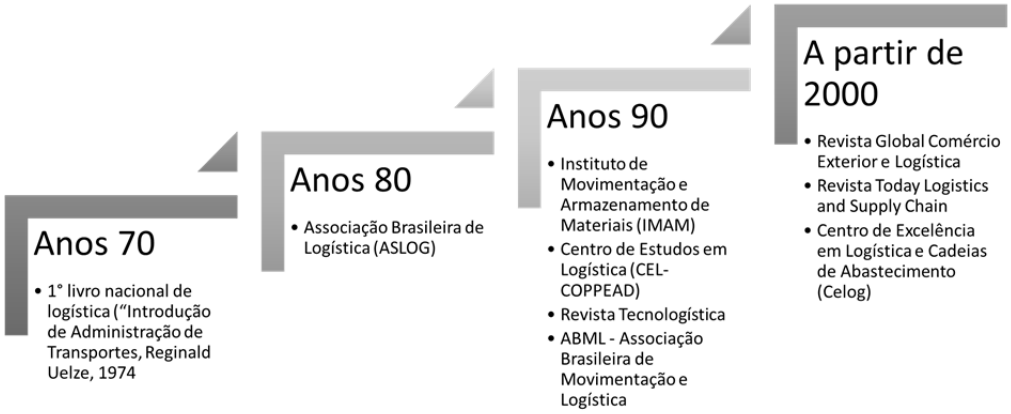


Figura 1 – Alguns fatos históricos sobre a educação em logística no Brasil.

Fonte: Adaptado de MACHLINE, 2011.

Ainda no contexto no ensino de logística, o curso de mestrado profissionalizante em logística oferecido pela PUC-RJ há mais de dez anos e possui nota máxima na avaliação do CAPES entre os anos de 2007-2009 também contribui para o ensino de logística no país. A instituição entende o *gap* de conhecimento em logística no país e propõe soluções customizadas, como por exemplo, a existência de várias turmas modo *in company* para a Petrobrás. O curso tem objetivo de oferecer boa base metodológica, que permita ao aluno desenvolver aplicações práticas e uma educação continuada de forma autodidata ou formal.

NOVAES (2001), em seu artigo onde descreve um panorama sobre os profissionais de logística no Brasil, aborda de forma clara o atraso enquanto as práticas aplicadas pelas empresas brasileiras em comparação com empresas situadas nos Estados Unidos, Europa e no Japão. Ainda de acordo com o autor essa defasagem demonstra a necessidade de profissionais qualificados em logística para atuar dentro das diversas áreas tradicionais como transporte, armazenagem, etc. e enfatiza a aplicação da modelagem matemática e da

computação proporcionando uma nova abordagem aos diversos desafios da logística.

NOVAES (2001) e ARKADER (1998) concluem que a falta de mão de obra qualificada é uma grande barreira no avanço da logística no Brasil.

NOVAES (2007) descreve sobre a mão de obra qualificada brasileira cita um exemplo que representa bem a pouca maturidade na área de logística dizendo que há diretores de empresas que creem que a compra de softwares de roteirização para suas empresas irão resolver os problemas logísticos de uma firma.

ARKADER (1998) em seu trabalho sobre a mudança no ambiente empresarial e o ensino da logística no Brasil, enfatiza a defasagem temporal no ensino da logística entre o Brasil e os Estados Unidos na década de 90 e sugere dois motivos para esse acontecimento. O primeiro motivo seria a demora em adotar o conceito de gestão de cadeia de suprimentos no Brasil e o segundo devido à própria forma como era ensinada a logística, colocando-a sempre numa visão funcional e dificultando o entendimento e a importância do assunto, o qual exige uma visão empresarial mais larga. De maneira positiva, a autora destaca o aumento de instrutores estrangeiros no Brasil que ministram cursos e seminários, ajudando a diminuir a defasagem atual do saber em logística no Brasil.

ZINN (1996, apud ARKADER 1998) enfatiza a predominância de mão de obra pouco qualificada no Brasil e que a grande maioria das escolas de ensino em logística no Brasil possuem padrões significativamente mais baixos quando comparados a escolas dos Estados Unidos, que não destacam conceitos importantes como *postponement, cross-docking* e relacionamento com operadores logísticos.

O dinamismo e a complexidade crescente da economia brasileira e os grandes desafios logísticos que eles trazem, torna-se fundamental o investimento no ensino de logística no país de modo a qualificar os profissionais da área que consequentemente trará muitos benefícios e competitividade para o país. ARKADER (1998) conclui o seu trabalho também enfatizando que a aquisição de conhecimento em logística e a capacidade de escolher ferramentas e técnicas adequadas para cada momento é fundamental para a formação do profissional de logística.

2.3 Distribuição Física

NOVAES (2007), a partir do ponto de vista da logística, define o conceito de distribuição física como sendo o conjunto de processos operacionais e de controle que permitem transferir os produtos desde o ponto de fabricação até o ponto em que a mercadoria é entregue ao consumidor final.

KOBAYASHI (2000) define três fatores fundamentais na distribuição física; clientes, produtos e estabelecimentos. O autor reconhece que existem outros fatores que impactam diretamente num sistema de distribuição física, mas esses três teriam uma importância decisiva.

BOWERSOX (2001) descreve o conceito de distribuição física como todas as atividades relacionadas com o fornecimento de serviço ao cliente, sendo que seu principal objetivo é prestar o serviço de acordo com o nível de serviço definido estrategicamente pela empresa ao menor custo possível.

2.3.1 Sistemas de Distribuição

Um sistema de distribuição física é composto pelas operações logísticas por onde um produto irá trafegar desde a saída da fábrica até o consumidor final. Essas operações podem ser compostas pelos diversos tipos de transporte, como: transporte marítimo, aéreo, terrestre, etc. e outras operações como armazenagem, troca de informações entre outras (NOVAES, 2007).

O processo de definição de um sistema de distribuição física é tão abrangente que sua definição se aproxima do próprio conceito de logística. De acordo com NOVAES (2007) o objetivo da distribuição física é o de levar os produtos certos para os lugares certos, no momento certo e com o nível de serviço desejado, pelo menor custo possível.

NOVAES (2007) descreve alguns componentes que fazem parte de um sistema de distribuição que podem ser classificados como, componentes físicos ou componentes informacionais. O autor informa que apesar de existirem diversas situações possíveis em um sistema de distribuição o autor resume os diferentes

tipos de configurações possíveis nos sistemas de distribuição física, em basicamente, dois tipos:

Tipo 1 – Sistema de Distribuição “UM PARA UM”

Nessa configuração, por exemplo, um veículo ou outro meio de transporte é totalmente carregado no ponto de origem que pode ser uma fábrica, depósito, etc., e transporta a carga para outro ponto de destino, que por sua vez, pode ser um outro depósito, loja, fábrica, etc.

Tipo 2 – Sistema de Distribuição “UM PARA MUITOS” ou compartilhada

Nessa situação, o meio de transporte terá diversos destinos para as mercadorias que estão sendo transportadas. Um caso típico desse tipo de sistema ocorre quando um veículo é carregado de mercadorias no depósito do varejista e tem que distribuir as mercadorias em diferentes lojas, executando assim, um roteiro. A Figura 2 mostra um esquema típico de distribuição.

Figura 2 - Esquema típico de distribuição um para muitos.

Fonte: NOVAES, 2007.

NOVAES (2007) descreve que esse tipo de sistema é influenciado por 15 fatores, sob o ponto de vista logístico. São eles:

1. **zonas ou bolsões de entrega** serão as áreas ou zonas de atendimento, sendo cada bolsão alocado normalmente a um veículo;
2. **distância** entre o CD e o bolsão de entrega;
3. **velocidades** operacionais médias, sendo:
    a. velocidade no percurso entre o depósito e bolsão;
    b. velocidade no percurso dentro do bolsão;
4. **tempo de parada** cada cliente;
5. **tempo de ciclo** (necessário para completar um roteiro e retornar ao depósito);
6. **frequência** das visitas as lojas ou aos clientes (diário, semanal, etc.);
7. **quantidade de mercadoria** (toneladas, *pallets,* caixas, etc.) a ser entregue em cada loja ou cliente do roteiro;
8. **densidade** da carga;
9. **dimensões e morfologia** das unidades transportadas;
10. **valor** unitário;
11. **acondicionamento (**carga solta, paletizada, a granel, etc.);
12. **grau de fragilidade;**
13. **grau de periculosidade;**
14. **compatibilidade** entre produtos de natureza diversa;
15. **custo global**.

O próprio autor reconhece que a quantidade de elementos que formam um sistema de distribuição não se esgotam na lista anterior. Outros elementos como, inserção de variáveis aleatórias nos tempos, balanceamento de horas de motoristas, condições meteorológicas, restrições geográficas, entre outros.

3 Softwares SIG, Orientação a Objetos e Java

3.1 Software de código livre

De acordo com RAINER (2011) há uma tendência dentro do setor de software para sair do software proprietário (aquele que foi desenvolvido por uma empresa e possui restrições em relação ao seu uso, cópia e modificação) e passar para o software de código aberto. O software de código aberto é de uso livre, tem direito autoral e é distribuído com termos de licença garantindo que o código fonte estará sempre disponível.

No contexto acadêmico, a aplicação e uso de software de código aberto traz diversos benefícios como a flexibilidade do número de licenças, manutenções, adição de funcionalidades, entre outros quando comparado a um software proprietário. Por exemplo, a rotatividade de alunos, a possibilidade de aplicação em zonas de ensino com pouco incentivo financeiro, a possibilidade de

customização para fins didáticos, a construção e teste de novas funcionalidades pela área científica, etc. Todos esses fatores para a adoção de softwares de código aberto pelas escolas e universidades.

3.2 Paradigma de Orientação à Objetos

PRESSSMAN (1987) e SOMMERVILLE (2010) relacionam o paradigma da orientação a objetos a visão de mundo onde todas as coisas existentes poderiam ser consideradas objetos. Essa facilidade na abstração facilita o entendimento e a modelagem de um software.

PRESSSMAN (1987) descreve que apesar de ter sido proposta a partir da década de 1960, a orientação a objetos só tornou-se largamente usada 20 anos depois.

A orientação a objetos alguns conceitos como; herança, polimorfismo, encapsulamento, entre outros que à primeira vista pode parecer trazer uma certa complexidade, mas quando conhecido a fundo descobre-se uma enorme quantidade de benefícios no seu uso.

Entre os benefícios proporcionados pela a orientação a objetos, podemos citar:

- Favorece o reuso de código
- Pode trabalhar com alto nível de abstração
- Unificação do padrão de desenvolvimento de software
- Facilidade para comunicar sobre o sistema
- Redução da dificuldade para desenvolver sistemas complexos
- Ciclo de vida mais longo para os sistemas

De acordo com TURBAN (2004) um sistema orientado para objetos é uma nova forma de programar que busca a redução de custo tanto de implantação quanto o custo de manutenção de um sistema. A tecnologia orientada a objetos favorece a modularização de um sistema, ou seja, permite o desenvolvimento de

unidades independentes de softwares que podem ser compartilhadas, compradas e/ou reutilizadas.

Para RAINER (2011) a vantagem em desenvolver um sistema orientado a objetos é porque desde o início da construção do sistema, o que importa é descobrir os aspectos do mundo real, como eles se relacionam e trazer para dentro dos sistemas. Sua utilização gera economia de tempo gasto na escrita do código e promove a reutilização de códigos. Ao contrário dos métodos tradicionais que se preocupam com a tarefa que o sistema irá executar, o que dificulta, por exemplo, futuras manutenções e melhorias no sistema.

3.2.1 Java

TURBAN (2004) define a tecnologia Java como uma "ferramenta promissora" pelas suas características como orientação a WEB, aspectos de segurança de código e por ser totalmente voltado para a tecnologia de orientação a objetos.

Segue abaixo alguns pontos fortes na adoção do Java:

- Facilidade de aprendizado - Java é a linguagem de programação escolhida em universidades e instituições de ensino em todo o mundo. O modelo do Java para gerenciamento de memória, *multithreading* e tratamento de exceções fazem que esta seja uma linguagem eficiente tanto para desenvolvedores novatos quanto para desenvolvedores experientes.
- Independência de Plataforma - O Java é executado na maioria dos hardwares e principais plataformas de sistema operacional com software JVM (*Java Virtual Machine*) mantido pela Oracle e por uma extensa comunidade.
- Baseado em padrões - A linguagem Java e a tecnologia relacionada evoluem por meio do *Java Community Process*, que é um mecanismo para o desenvolvimento de especificações técnicas para tecnologia Java.
- Prevalência mundial - Java é a plataforma de aplicativo mais popular do planeta e oferece um ecossistema para desenvolvedor vibrante impulsionado por ferramentas, livros, bibliotecas, exemplos de códigos eficientes e muito mais.

O programa base em que se apoia o protótipo, o OpenJUMP, também foi desenvolvido em Java.

Segue abaixo (Figura 3) alguns números que comprovam a expressividade da linguagem atualmente.

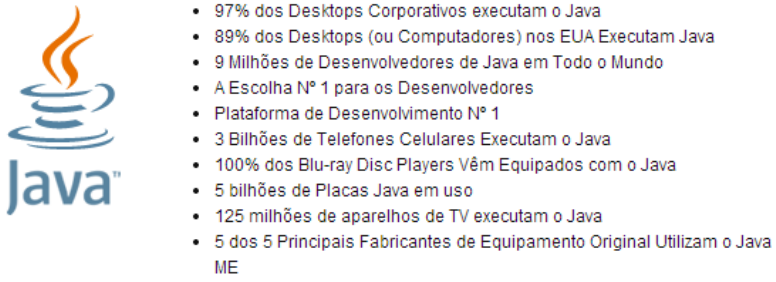


Figura 3 - Expressividade do uso do Java.

Fonte: Site ORACLE (2014).

3.3 SIG - Sistema de Informação Geográfica

3.3.1 Definição e contexto de Aplicação do SIG

LEWIS (apud ROSE, 2001) define um SIG como um sistema de gerenciamento de banco de dados computacional para capturar, armazenar, recuperar, analisar e visualizar dados espaciais.

Um SIG proporciona relacionar, de forma visual, quaisquer tipos de dados com junto a um modelo espacial. Sendo assim, sua aplicação é ampla em várias áreas como: gestão urbana, geografia, arqueologia, exploração de petróleo, logística, etc.

3.3.2 A crescente importância do SIG para Logística

Conforme bem definido por ROSE (2001), um SIG proporciona uma convergência de campos tecnológicos. Com base no aspecto da multidisciplinariedade do SIG pode-se observar sua aplicação em muitas áreas

como: planejamento urbano, geografia, agronomia, florestal, processamento de dados, arquitetura e urbanismo, gerenciamento de serviços, engenharia de transportes entre outros.

A autora também destaca que a crescente procura e adoção de SIG tem origem na necessidade de ferramentas que auxiliem os tomadores de decisão nas áreas de planejamento urbano e de transportes.

SILVA (apud ROSE 2001) faz uma revisão dos principais trabalhos que abordaram SIG e transportes. Segue a revisão conforme a Figura 4 abaixo:

Trabalho	Autor
Associação de Imagens de Satélites a Modelo Matemático para o Planejamento de Transportes	(Ferreira et al., 1994)
Geração de Imagens para Cadastro como Auxílio ao Planejamento do Transporte Público	(Nassi et al., 1994)
Avaliação dos Impactos do Crescimento Urbano sobre os Transportes	(Silva et al., 1995)
Reestruturação de um sistema de Transporte Público Urbano	(Silva e Kawamoto, 1995)
Roteirização para Distribuição de Jornais	(Silva e Grubman, 1996 e Rocha, 1996)
Localização de Pontos de Parada com Auxílio de um SIG	(Tesima e Lapolli, 1996)
Análise do Problema de Roteirização de Veículos para o Processo de Coleta e Descarga de Resíduos de Serviços de Saúde	(Graciolli et al., 1997)
Definição de Zonas de Tráfego, a partir de Setores Censitários	(Sanches, 1997)
Aplicações de SIG em coordenação Semafórica	(Oliveira e Ribeiro, 1997)
Modelos de Geração de Viagens	(Taco et al., 1997)
Avaliação da Acessibilidade aos Transportes	(Sales Filho, 1997; Raia Jr. et al.,1997; Silva,1998)
Desenvolvimento de uma Metodologia para Otimização de Frota e Redução de Custos Operacionais para Serviços de Ônibus fretados para Transporte de Funcionários	(Martins et al., 1997)

Figura 4 - Exemplos de trabalhos com SIG e transportes.

Fonte: Adaptada de ROSE (2001).

3.3.3 Escolha de um SIG

O poder de transformação do negócio na adoção de um SIG conforme mostrado anteriormente é altamente benéfico para a empresa. Porém, os custos de licenças desse software é uma barreira que impede a sua adoção em larga escala.

Através do trabalho de ROSE (2001), onde avaliou alguns SIG para aplicação na área de transportes é possível perceber a grande vantagem econômica na adoção de um SIG gratuito.

A Figura 5 contém os valores de aquisição de dois conhecidos *softwares* SIG atualizados.

Fabricante	Produto	Valor	Observação
Caliper	Standard TransCAD	12 000,00 USD	por licença
Caliper	Base TransCAD	4 000,00 USD	por licença
esri	ArcGIS Desktop	1 500,00 USD	por licença

Figura 5 - Tabela de custos de softwares SIG.

Fonte: http://www.esri.com e http://www.caliper.com (2014)

Em comparação com um SIG de livre distribuição, que é gratuito, conforme será apresentado adiante, fica clara a motivação da escolha em termos de custos. Além do custo de licença, o custo de customização do software também possui elevada importância, pois uma vez que um software é usado no ambiente de ensino, ele tem que ser flexível para sofrer melhorias e evoluir de acordo com as novas técnicas que também serão ensinadas.

3.3.4 OpenJUMP

O software OpenJUMP (Figura 6 - Imagem do Programa OpenJUMP.), inicialmente chamado de Jump (*Java Unified Mapping Plataform*) é um SIG (Sistema de Informação Geográfica) construído com a tecnologia Java. O programa foi criado pela empresa Vivid Solutions e lançado no ano de 2006, mas atualmente o sistema é mantido por um grupo de voluntários de vários países.

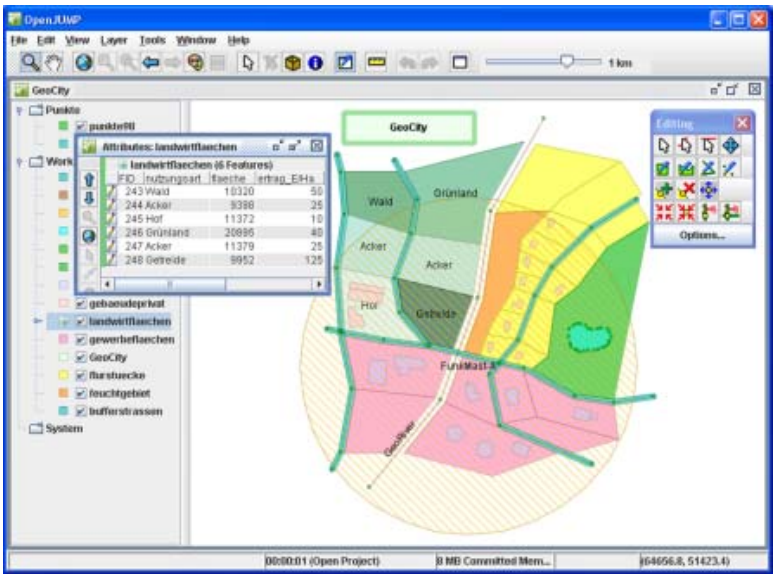


Figura 6 - Imagem do Programa OpenJUMP.

Fonte: Site OpenJUMP.

Por ser um software de código aberto (Livre) torna-se interessante o seu uso em diversos tipos de projetos, especialmente para adicionar funcionalidades ao programa, por meio de *plugins*, de forma mais rápida e eficiente.

O motivo principal para a escolha desse programa deve-se ao fato de ser gratuito, permitindo assim seu uso e adição de novas funcionalidades sem nenhum custo e universalização das soluções implementadas no software.

Outro motivo de escolha é devido ao tipo de tecnologia em que o programa foi desenvolvido, tecnologia Java. É uma tecnologia relativamente nova e robusta. Outro fator importante para sua escolha foi a disponibilidade de informação técnica fornecida pelos fabricantes do produto em sites que colabora bastante para o entendimento do funcionamento do programa e construção de novas funcionalidades.

4 Procedimentos para solução de problemas em Logística

Neste capítulo serão apresentados os procedimentos a serem implementados posteriormente pelo protótipo, para solucionar diversos tipos de problemas logísticos. O objetivo é descrever procedimentos mais comuns disponíveis na literatura técnica, ou seja, ainda existem muitos outros procedimentos disponíveis que não serão abordados neste trabalho.

Os problemas escolhidos foram selecionados por serem recorrentes no dia a dia de um profissional de logística, assim sendo de extrema importância que ele tenha algum contato durante o período de aprendizado em alguma instituição.

Os procedimentos foram escolhidos por tratarem problemas logísticos de uma forma mais simples e principalmente por ter uma razoável proximidade das soluções obtidas com métodos bem mais complexos.

Segue abaixo a lista de problemas e os procedimentos para solucioná-los:

- *Procedimentos para solução de problemas de localização*
  - Definição do ponto central pela métrica Retangular
    - Método de Fibonacci
    - Método da Derivada
  - Definição do ponto central pela métrica Euclidiana
    - Método de Weisfield
- *Procedimentos para problemas de formação de cluster*
    - Algoritmo de Kruskal adaptado para formação de vários *Clusters*
- *Procedimentos para problemas de roteirização*
  - Métodos de Construção de roteiro
    - Método de Clarke e Wright
  - Método de Melhoria de Roteiros
    - Método 2-opt
- *Procedimentos para solução de problemas de formação de zonas*

- Métodos de Construção de zonas de atendimento
    - Método de divisão de regiões em zonas de acordo com nível de serviço
    - Método de Clarke e Wright com parâmetro de forma

4.1 Procedimentos para problemas de localização

4.1.1 Problemas de localização

Em geral, esse problema ocorre quando se deseja localizar uma instalação (depósitos, fábricas, instalações, etc.) para atender um determinado conjunto de pontos, também chamado na literatura como projeto de redes. Diferentes critérios como; custo de investimento, custo de transporte, custo de estoque, relações entre instalações, nível de serviço entre outros, podem ser usados para a definição dessa instalação (BOWERSOX, 2001; LEAL, 2012).

BOWERSOX (2001) categoriza o problema de localização dentro da teoria da integração que busca dar suporte na criação e implementação de uma estratégia logística. O autor enfatiza que a definição de localização é uma questão estratégica que deve ser avaliada com regularidade. No caso em que a instalação seja um depósito, o autor descreve três classificações gerais que motivam a existência de depósitos; depósitos direcionados pelo mercado, depósitos direcionados pela produção e depósitos localizados em pontos intermediários.

De acordo com essa classificação é possível imaginar os diferentes critérios que podem existir dentro de cada classificação que buscam responder as seguintes questões:

- Qual é a quantidade de instalações que a empresa deve ter e onde devem estar localizados?
- Quais clientes cada instalação irá servir?
- Quais linhas de produtos que devem ser fabricadas ou armazenadas em cada instalação?
- Quais canais logísticos devem ser usados por cada instalação?

• Qual é a composição ideal entre integração vertical e terceirização de serviços de distribuição física para cada instalação?

Conforme descrito acima, o nível de complexidade que pode ser alcançar com a modelagem de um problema de localização é alto. Devido à grande quantidade de dados e regras que pode ser gerar no problema, as ferramentas computacionais aliadas aos métodos iterativos se tornam o único meio real com capacidade para se obter as respostas.

BALLOU (2006) descreve o problema como uma questão de compensação dos custos relevantes para determinada localização de uma instalação, que incluem:

- Custos de produção e compra
- Custos de estocagem e manuseio no armazém
- Custos fixos do armazém
- Custos de manutenção do estoque
- Custos de processamento dos pedidos de estoques e pedidos dos clientes
- Custos de transporte de entrada e saída do armazém

Existem diversos métodos matemáticos que podem ser usados num problema de localização, podendo ser caracterizados como exatos, de simulação ou heurísticos. O problema pode ser aplicado para definir localização única ou definir múltiplas localizações (BALLOU, 2006).

Conforme NOVAES (2007) o centro de gravidade, também chamado de ponto central, pode ser usado, por exemplo, no caso de aplicação do método de varredura onde o ponto de depósito está distante da área que possui os clientes. Nesse caso o centro de gravidade diminui as distorções evitando roteiros extremamente alongados. A Figura 7 mostra o ponto de centro de gravidade dentro de uma região de distribuição.

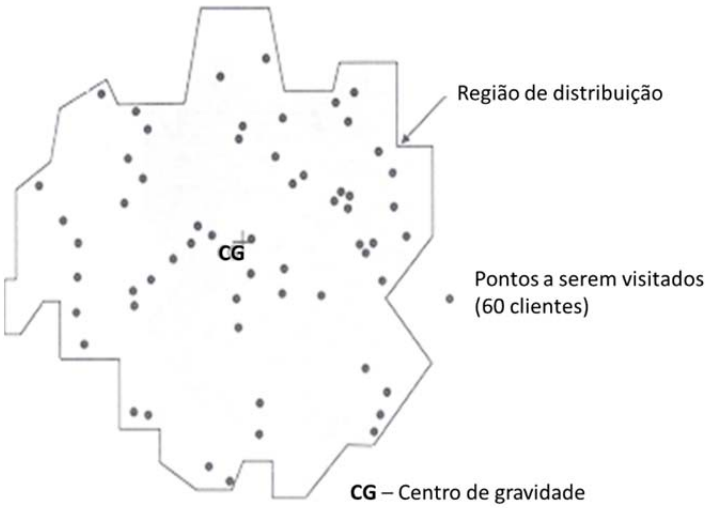


Figura 7 - Localização dos clientes e do centro de gravidade numa região de distribuição.

Fonte: NOVAES. 2007.

4.1.2 Procedimentos para solução de problemas de localização

4.1.2.1 Procedimentos para definição do ponto central

4.1.2.1.1 Definição do ponto central pela métrica Retangular

Métrica Retangular

A distância calculada pela métrica retangular, é aquela que se observa em uma rede de ruas em forma de malhas ou quadrículas regulares (LEAL, 2012). A Figura 8 mostra um exemplo de distância pela métrica retangular.

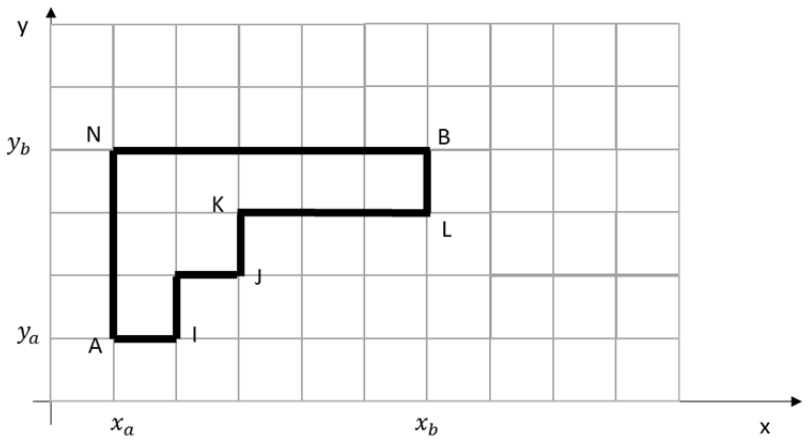


Figura 8 - Exemplo de distância entre os pontos A e B pela métrica retangular.

Fonte: LEAL, 2012.

O cálculo da distância pela métrica retangular (DR) é realizado da seguinte maneira:

$$DR_{AB} = |x_B - x_A| + |y_B - y_A| \quad (1)$$

Em geral, realiza-se o cálculo de distância pela métrica retangular quando a malha possui formas de quadriláteros bem definidos, por isso ela também é conhecida como a métrica de Manhatan, devido as formas viárias bem definidas (LEAL, 2012). Caso contrário, a diferença entre a distância real em comparação com a métrica será relevante e pode ser necessário aplicar um fator de correção. Por exemplo, NOVAES (1989) identificou que para usar o cálculo de distância pela métrica retangular na cidade de São Paulo, para aproximação a distância real deveria aplicar o fator de correção $D = 1{,}13 + 1{,}04\ DR$.

O fator de correção é aplicado tanto para a métrica retangular, como visto anteriormente, como para a métrica euclidiana que será abordada mais à frente. Torna-se evidente que para cada métrica o fator irá variar devido à própria característica da métrica e a outros fatores.

O fator de correção a ser aplicado irá depender da localidade que está sendo analisada. Caso a localidade esteja localizada, por exemplo, sobre uma região

montanhosa ou com muitas restrições de tráfego o fator de correção será diferente para ambos os casos.

Para definir um fator de correção que seja ajustado a localidade analisada, pode-se aplicar a técnica de regressão linear simples.

A regressão linear simples é método causal que busca estimar um valor de uma variável a partir de uma série de dados que é baseada no método dos mínimos quadrados. A teoria da regressão permite que se estabeleçam relações entre variáveis que se inter-relacionam e cujas informações estão disponíveis (dados pré-coletados), relações às quais se associam os modelos de regressão.

O processo de regressão significa que os pontos plotados no gráfico são definidos, modelados ou regredidos, a uma reta que corresponde à menor distância possível entre cada ponto plotado e a reta.

O modelo de regressão linear simples é composto da seguinte maneira:

$$Y_i = \alpha + \beta x_i + \epsilon_i \quad (2)$$

Seja:

Y_i - Variável que será explicada;

α - É uma constante, que representa a interseção da reta com o eixo vertical;

β - É outra constante, que representa o coeficiente angular da reta;

x_i - Variável que irá explicativa;

ϵ_i - Variável que inclui todos os fatores residuais mais os possíveis erros de medição.

O seu comportamento é aleatório. Para que essa fórmula possa ser aplicada, os erros devem satisfazer determinadas hipóteses, que são: serem variáveis normais, com a mesma variância σ^2 (desconhecida), independentes e independentes da variável explicativa *X*.

A técnica é relativamente simples e está disponível em vários programas gerenciadores de planilhas como o Microsoft Excel e em softwares estatísticos.

4.1.2.1.1.1 Definição do ponto central pela métrica Retangular com o Método de Fibonacci

O objetivo do cálculo do ponto central com a métrica retangular busca definir a melhor localização numa malha de acordo com critérios como custo, tempo, faturamento, entre outros, que por sua vez é similar ao que será apresentado adiante, porém com a métrica euclidiana (BITENCOURT, 2005).

A função a ser minimizada para encontrar o ponto central é:

$$Min\ f(x,y) = \sum_{i=1}^{N} Pi\ [|x - x_i| + |y - y_i|] \tag{3}$$

A função acima é separável em $x\ e\ y$:

$$f(x,y) = \sum_{i=1}^{N} Pi \left| x - x_i \right| + \sum_{i=1}^{N} Pi \left| y - y_i \right| \tag{4}$$

Ou seja:

$$f_1(x) = \sum_{i=1}^{N} Pi\ |x - x_i| \quad \text{e} \quad f_2(y) = \sum_{i=1}^{N} Pi\ |y - y_i| \tag{5}$$

Igual a:

$$f(x,y) = f_1(x) + f_2(y) \tag{6}$$

Segundo NOVAES (1989) o método de Fibonacci é eficiente para a determinação dos mínimos $f_1(x)\ e\ f_2(y)$. Porém após o surgimento do método da derivada (será explicado em seguida), que por sua vez é um método exato, apresentando resultados melhores do que o método de Fibonacci.

O processo proposto por Fibonacci também é iterativo e aplicado igualmente para x e y. Segue abaixo (Figura 9) o pseudocódigo de aplicação do método descrito por BITENCOURT (2005) e LEAL (2012):

1: **Procedimento proposto por Fibonacci**

2:Início

3: Leia *K:* /* Obs: *K* é o número de iterações que determinará a precisão final do cálculo */

4: Encontre o extremo inferior de *x*;

5:*Xi*:= extremo inferior;

6: Encontre o extremo superior de *x*;

7:*Xs*:=extremo superior;

8: *S1*:=*Xi;*

9: *S2*:=*Xs;*

10: *F*:= $\frac{2}{(1+\sqrt{5})}$;

/* Obs: *F* é o inverso da razão da seção áurea */

/* Obs: O segmento *S2*-*S1* é dividido em três partes pelos pontos *R1 e R2* */

11: *i*:=1; /* Obs: *i* é o contador de iterações */

12: Enquanto $i \leq K$ faça

13: *R1*:=(1-*F*)×(*S2*-*S1*)+*S1*;

14: *R2*:=*F*×(*S2*-*S1*)+*S1*;

15: *G1*:= $f_1(R1)$;

16: *G2*:= $f_1(R2)$;

17: Se *G1*≤*G2* então

18: *S2*:=*R2*

19:Senão

20:*S1*:=*R1*

21:Fim-se

22: *I*:=*i*+1;

23: Fim enquanto

24: *Xc*:= $\frac{(S1+S2)}{2}$

/* Obs: *Xc* é a abcissa do ponto central procurado */

25:Fim

Figura 9 - Procedimento proposto por Fibonacci

Após o término do cálculo anterior para encontrar a o valor da abcissa, repete-se igualmente o procedimento para agora para encontrar o valor do eixo das ordenadas.

BITENCOURT (2005) define duas formas de definir a precisão para o método de Fibonacci. A primeira, como citada anteriormente é definindo o número de iterações (*K)* onde a precisão (*ε)* final será definida por:

$$\varepsilon = F^k = \left(\frac{2}{1+\sqrt{5}}\right)^k \tag{7}$$

A segunda é fazendo o processo inverso da primeira, ou seja, define-se primeiro a precisão e calcula-se a quantidade de iterações (*K)* que serão necessárias para atingi-la conforme a fórmula abaixo:

$$K = \frac{\log(\varepsilon)}{\log(0{,}618034)} \tag{8}$$

Lembrando que é necessário se calcular separadamente, ou seja, para x e y, tanto na primeira quanto pela segunda forma.

4.1.2.1.1.2 Definição do ponto central pela métrica Retangular com o Método da Derivada

Também é possível calcular o ponto central com a métrica retangular através do método da derivada, que por sua vez é mais eficiente do que o método de Fibonacci. O procedimento para este método busca encontrar um ponto onde a derivada da função nomeada como *DIF*, muda de sinal, por isso o nome método da derivada (LEAL, 2012).

Definição do método:

- Tem-se um conjunto de pontos, com coordenadas (x_i, y_i)
- Para cada ponto, existe um peso (p_i) associado

A função objetivo a ser minimizada é a mesma que foi apresentada anteriormente para o método de Fibonacci. Deseja-se encontrar os valores da coordenada de um ponto central que minimiza a função:

$$F(x,y) = \sum_{i=1}^{N} P_i \left[|x - x_i| + |y - y_i| \right] \qquad 9)$$

O método funciona da seguinte maneira:

- Calcula-se a derivada da função em cada intervalo e descobre-se o ponto limite entre dois intervalos, onde a derivada passa de valor negativo em um intervalo a não negativo no intervalo seguinte.
- Como a função é separável, pode-se encontrar o mínimo de cada parcela da função, ou seja, encontrar o valor de x e depois o valor de y que minimizam cada parcela da função objetivo. A modelagem a seguir para a variável x também serve para a variável y. A função é contínua e por partes com descontinuidade em cada ponto entre dois intervalos.

Considerando um exemplo, seja uma função de quatro pontos, ordena-se em ordem crescente da coordenada em questão, ou seja, x ou y.

No intervalo à esquerda dos pontos a função assume o valor:

$$F_1 = P_1 \times (x_i - x) + P_2 \times (x_2 - x) + P_3 \times (x_3 - x) + P_4 \times (x_4 - x) \qquad (10)$$

A derivada com relação a x, neste intervalo, é:

$$DIF_1 = \frac{\partial F_1}{\partial x} = -P_1 - P_2 - P_3 - P_4 = \sum_{i=1}^{4} P_i \qquad 11)$$

No intervalo seguinte entre x_1 e x_2, a função assume o valor:

$$F_2 = P_1 \times (x_i - x) + P_2 \times (x_2 - x) + P_3 \times (x_3 - x) + P_4 \times (x_4 - x) \quad 12)$$

A derivada neste intervalo é:

$$DIF_2 = \frac{\partial F_2}{\partial x} = +P_1 - P_2 - P_3 - P_4 = \sum_{i=1}^{4} P_i + 2 \times P_i \quad 13)$$

Em geral, como entre dois intervalos a derivada sofre uma variação igual a duas vezes o valor do peso no ponto, em algum ponto, a derivada no intervalo seguinte terá passado de um valor negativo para um valor não negativo.

Observações:

- Se a derivada passou para um valor positivo, o ponto de mínimo da função é único, e o valor da variável correspondente a este mínimo, coincide com o valor da coordenada neste ponto.

- Se a derivada passou para o valor zero, o mínimo da função terá o mesmo valor em todo o intervalo entre o ponto que torna a variável zero e o ponto seguinte, onde a derivada se torna positiva, conforme a Figura 10.

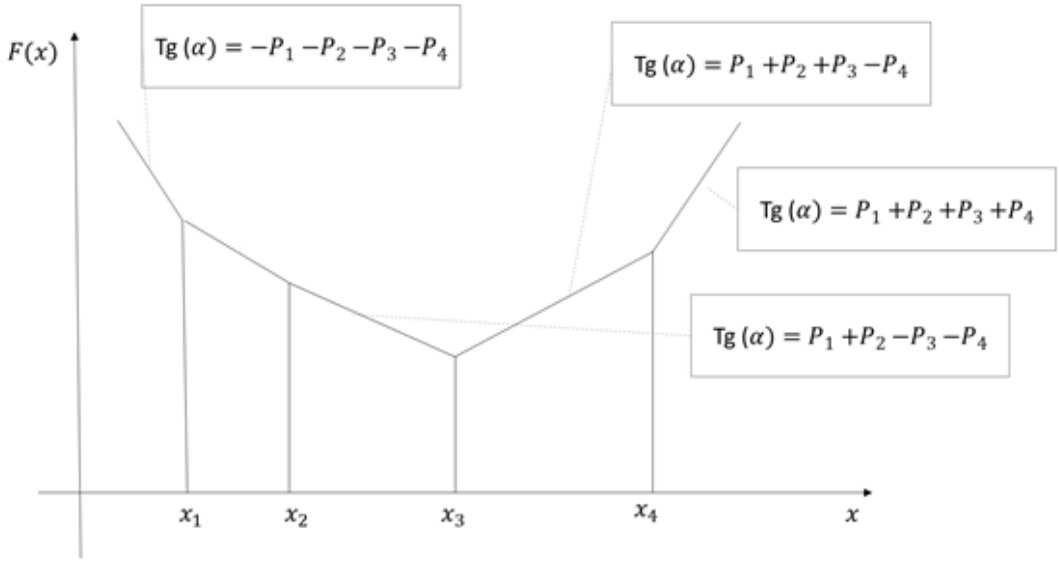


Figura 10 - Gráfico exemplo com intervalos da função de ponto central retangular

Fonte: LEAL, 2012.

Assim, pode-se definir o método da derivada como um método exato que retorna o valor mínimo da coordenada do ponto que torna mínimo o valor da função.

Escrito de outra forma, segue abaixo o procedimento para se alcançar os mínimos de x e y, pelo método da derivada é descrito por LEAL (2012):

1: **Método da derivada**

2: Ordena-se os pesos de *x* e *y* em ordem crescente;

3: Calcula-se o $DIF = -\sum_{i=1}^{n} P_i$

4: $i = 0$;

5: Enquanto $DIF < 0$;

6: Início:

7:$i = i + 1$;

8:$DIF := DIF + 2 \times p_i$;

9: FIM Enquanto

10: $x_{Min} = x_i$;

Figura 11 - Método da derivada

Após o término, repete-se o processo acima para o eixo y.

4.1.2.1.2 Definição do ponto central pela métrica Euclidiana

Métrica Euclidiana

A métrica euclidiana representa a localização dos pontos da rede através de duas coordenadas associadas aos eixos tradicionalmente denominados de x e y (Leal, 2012). Nesta métrica, a distância euclidiana (DE) entre dois pontos, A e B, é calculada da seguinte maneira:

$$DE_{AB} = \sqrt{[(x_B - x_A)^2 + (y_B - y_A)^2]} \quad 14)$$

Exemplo de distância entre dois pontos pela métrica euclidiana na Figura 12:

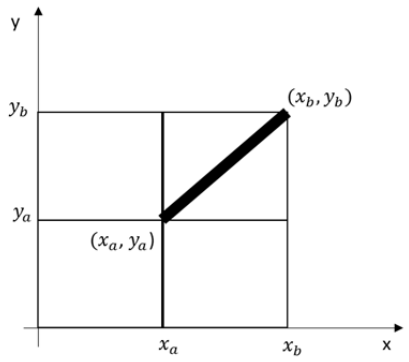


Figura 12 - Exemplo de distância entre os pontos A e B pela métrica euclidiana.

Fonte: LEAL, 2012.

Devido a forma da métrica euclidiana, que é uma linha reta ou também chamada de linha aérea, se difere da distância real. Para melhorar a aproximação até a distância real é comum usar um fator de correção. O valor desse fator irá variar de acordo com vários fatores, como a geografia da área que está sendo estudada, enquanto a forma das ruas de uma cidade, entre outros.

NOVAES (1989) sugere alguns fatores de correção em diversas situações:

Fator de correção para rodovias:

$$D = 23{,}9 + 1{,}11 \times DE \text{ (Para distância acima de 60 km)} \quad (15)$$

$$D = 1{,}48 \times DE \text{ (Para distâncias abaixo de 60 km)} \quad (16)$$

Fator de correção para ferrovias:

$$D = 9,\ 8 + 1{,}25 \times DE \quad (17)$$

Fator de correção para vias urbanas:

$$D \cong 1{,}3 \times DE \quad (18)$$

4.1.2.1.2.1 Definição do ponto central pela métrica Euclidiana com o Método de Weisfield

Segundo BITENCOURT (2005) o primeiro método iterativo elaborado para calcular o ponto central com a métrica euclidiana foi criado por Weisfield em 1937. O objetivo do cálculo é encontrar as coordenadas x e y de um ponto central para servir um conjunto de pontos i, onde cada ponto possui um peso Pi associado (LEAL, 2012).

A função a ser minimizada é:

$$Min\ f(x,y) = \sum_{i=1}^{N} Pi\sqrt{[(x - x_i)^2 + (y - y_i)^2]} \quad (19)$$

O ponto mínimo é obtido igualando as derivadas em relação à x e y iguais a zero:

$$\frac{\partial f(x,y)}{\partial x} = 0 \qquad \frac{\partial f(x,y)}{\partial y} = 0 \quad (20)$$

Chega-se as equações:

$$x = \frac{\sum_{i=1}^{N} Pixi \Big/ \overline{DE_i}}{\sum_{i=1}^{N} Pi \Big/ \overline{DE_i}} \qquad y = \frac{\sum_{i=1}^{N} Piyi \Big/ \overline{DE_i}}{\sum_{i=1}^{N} Pi \Big/ \overline{DE_i}} \tag{21}$$

Sendo que:

$$\overline{DE_i} = \sqrt{[(x - x_i)^2 + (y - y_i)^2]} \tag{22}$$

Cada eixo representa as coordenadas do ponto central. Como $DE_i = f(x, y)$, não há solução analítica direta, o que sugere uma resolução via método iterativo.

Esse método também pode ser resolvido com através do *Solver*, funcionalidade que faz parte do Microsoft Excel®.

Segue abaixo o processo iterativo usado para calcular o ponto central pela métrica euclidiana pelo método de Weisfeld (BITENCOURT 2005, LEAL 2012):

1: **Método de Weisfield**

2: Supõe-se que $DE_1 = DE_2 \ldots DE_N = \overline{DE}$

3: Elimina-se o DE no numerador e no denominador e pode-se calcular os valores iniciais de x e y através das seguintes fórmulas:

$$x^0 = \frac{\sum_{i=1}^{N} Pixi}{\sum_{i=1}^{N} Pi} \quad e \quad y^0 = \frac{\sum_{i=1}^{N} Piyi}{\sum_{i=1}^{N} Pi}$$

4: Calcula-se os valores $\sqrt{[(x - x_i)^2 + (y - y_i)^2]}$ utilizando os valores determinados anteriormente para x^0 e y^0.

5: Usa-se os valores determinados para $\sqrt{[(x - x_i)^2 + (y - y_i)^2]}$, para calcular x^1 e y^1 com as seguintes fórmulas:

$$x^1 = \frac{\sum_{i=1}^{N} Pixi \Big/ \sqrt{[(x^0 - x_i)^2 + (y_0 - y_i)^2]}}{\sum_{i=1}^{N} Pi \Big/ \sqrt{[(x^0 - x_i)^2 + (y_0 - y_i)^2]}}$$

e

$$y^1 = \frac{\sum_{i=1}^{N} Piyi \Big/ \sqrt{[(x^0 - x_i)^2 + (y_0 - y_i)^2]}}{\sum_{i=1}^{N} Pi \Big/ \sqrt{[(x^0 - x_i)^2 + (y_0 - y_i)^2]}}$$

6: Deve-se retornar as etapas 3 e 4 até que as iterações sucessivas obedeçam a um critério de convergência. Podendo ser definido como um valor de precisão (ε).

Figura 13 - Método de Weisfield

O método iterativo termina quando o valor de precisão (ε) é alcançado, conforme abaixo:

$$\frac{\left|x^{(k+1)} - x^{(k)}\right|}{x^{(k)}} \leq \varepsilon \quad e \quad \frac{\left|y^{(k+1)} - y^{(k)}\right|}{y^{(k)}} \leq \varepsilon \tag{23}$$

Esse valor de precisão (ε) é alcançado quando a diferença relativa entre dois pontos sucessivos da iteração for menor, ou igual, ao valor da precisão definida, devendo assim, encerrar o procedimento iterativo.

4.2 Procedimentos para problemas de formação de Clusters

4.2.1 Formação de Clusters - Clusterização

Clusterização, definindo de forma simples, é o processo de dividir os dados de todo o conjunto em diferentes grupos. Ela representa uma das principais etapas de processos de análise de dados, denominada análise de clusters (JAIN et al., 1999)

A análise de clusters é um dos tópicos de pesquisa mais importantes nas áreas de mineração de dados, reconhecimento de padrões, aprendizado de máquina e outros **(**L. HUANG et al., 2012). Ele tem amplas perspectivas de aplicação. É largamente aplicada nas áreas de negócio, biologia, análise climática, sociologia, análise de dados espaciais, web mining, processamento de imagens entre outros.

O algoritmo de Kruskal com uma pequena adaptação pode ser usado para a identificação de *clusters*. A seguir será mostrada a versão original do algoritmo de Kruskal e posteriormente a versão adaptada que permite a identificação de *clusters*.

4.2.1.1 Identificação de Árvore Geradora Mínima com o Algoritmo de Kruskal

Também chamada de Árvore de Extensão Mínima, é formada a partir de um grafo conectado e não orientado. Uma árvore de extensão mínima é um grafo parcial, que possui todos os vértices conectados e sem ciclos. O algoritmo de Kruskal pode ser aplicado para identificar uma Árvore Geradora Mínima. A Figura 14 mostra um exemplo de árvore geradora mínima num grafo com 250 vértices.

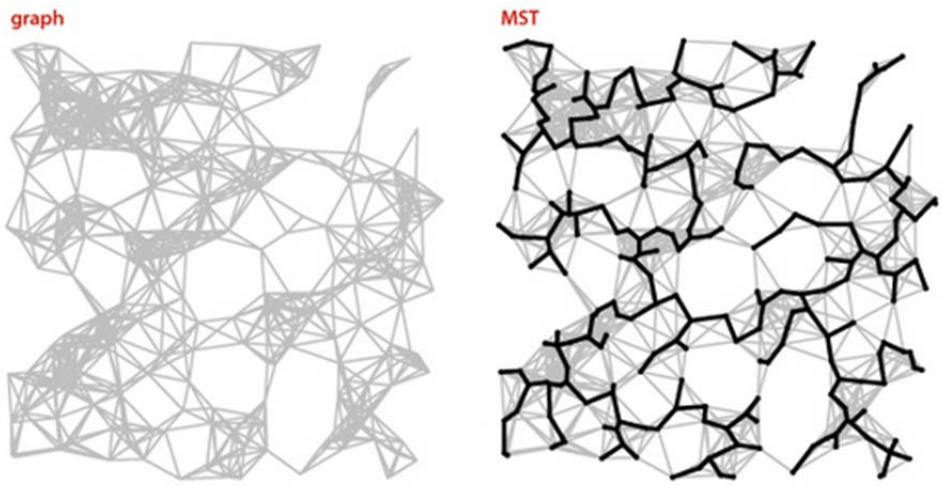


Figura 14 - Exemplo de uma Árvore Geradora Mínima (*Minimum Spanning Tree* – MST) com 250 vértices.

Fonte: *Princeton University – Department of Computer Science* (2014)http://www.cs.princeton.edu/courses/archive/fall05/cos226/lectures/mst.pdf

De acordo com COLIN (2007) a árvore de extensão mínima é caracterizada como mínima quando todos os vértices estão conectados por arcos cuja soma de seus pesos é a menor possível. Ainda segundo o autor há muitas aplicações deste

tipo de problema, sempre associadas à construção de redes com comprimento mínimo, ou seja, problemas onde há um conjunto de nós que precisam ser conectados uns aos outros com o custo mínimo.

WU e CHAO (2005) destacam o uso de árvore geradora mínima é usado amplamente por empresas de telecomunicações no planejamento de redes, definindo a malha de conexão num conjunto de sites usando a menor quantidade de cabeamento.

Algoritmo de Kruskal

O algoritmo de Kruskal foi desenvolvido por Joseph Kruskal em 1956. O resultado do algoritmo de Kruskal é uma Árvore Geradora Mínima, explicada anteriormente, cujo objetivo é basicamente definir um conjunto de arestas, que possuem um peso associado, tal que conecte todos os nós com o menor somatório dos valores de arestas.

Ele é baseado na teoria dos grafos o seu objetivo é formar, a partir de um grafo conexo com pesos, uma árvore geradora mínima, também chamada de árvore de extensão mínima. O procedimento retorna um subconjunto das arestas que forma uma árvore que inclui todos os vértices, onde o peso total, dado pela soma dos pesos das arestas da árvore, é minimizado.

Caso o procedimento seja aplicado em um grafo desconexo, ou seja, um grafo que possui dois ou mais componentes conexos, então o procedimento retorna uma floresta geradora mínima. Uma floresta é um grafo cujas componentes conexas são árvores.

O algoritmo de Kruskal é um exemplo de um algoritmo guloso. Assim seu procedimento busca sempre ótimos locais com a esperança que essas escolhas levem a um ótimo global.

MORENO e RAMÍREZ (2011) descrevem o funcionamento do algoritmo de Kruskal da seguinte maneira (Figura 15 e Figura 16):

Procedimento Kruskal

Entrada: $G = (V,E)$: grafo

1: $T \leftarrow 0$ /*conjunto de arestas*/

2: $u \leftarrow$ vértice inicial qualquer

3: $U \leftarrow \{u\}$ /*conjunto de vértices*/

4: enquanto $U \neq V$

5: $(u,v) \leftarrow$ aresta de menor custo, tal que $T \cup \{(u,v)\}$ não possui ciclos.

6: $T = T \cup \{(u,v)\}$

7: $U = U \cup \{ v\}$

8: fim

Saída: T árvore geradora mínima

Figura 15 - Procedimento Kruskal

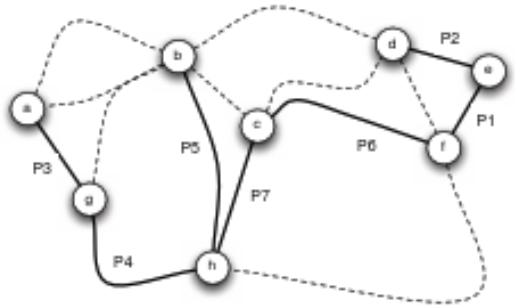


Figura 16 - Exemplo de passos para criar uma Árvore Geradora Mínima (linha em negrito).

Fonte: MORENO e RAMÍREZ, 2011.

4.2.1.2 Identificação de *Clusters* com o Algoritmo de Kruskal adaptado

O algoritmo de Kruskal pode ser usado em problemas de análise de *clusters* através de uma pequena modificação no seu procedimento. Através dessa modificação, e definindo previamente a quantidade de clusters que se deseja encontrar, o procedimento irá formar a quantidade de clusters simplesmente excluindo os arcos (também chamado de arestas) de maior peso até atingir o número de clusters definido.

Assim, o seu funcionamento é basicamente conforme descrito anteriormente pela versão padrão do algoritmo de Kruskal, porém ao término do processo, sendo "*k"* a quantidade de cluster que se deseja criar, deve-se excluir *k-1* arestas para encontrar a resposta.

L. HUANG et al. (2012) aplicaram o procedimento de Kruskal para melhorar um procedimento que define a localização de clusters dentro de uma região de pontos. O Kruskal foi usado para gerar uma solução inicial que consistia basicamente em gerar uma quantidade pré-definida de clusters que depois seria melhorada com a execução de outro procedimento chamado de *K-means* que busca um melhor ajusto da localização de um cluster entre seus pontos. Duas vantagens principais foram obtidas através da aplicação do Kruskal; estabilidade e acurácia das soluções geradas.

4.3 Procedimentos para problemas de roteirização

4.3.1 Problemas de Roteirização de Veículos - PRV

O problema de roteamento de veículos clássico (PRV) é baseado em definir rotas que serão percorridas por uma frota homogênea que possui alguma capacidade, que inicia e retorna a um único depósito. A solução deve satisfazer a demanda de atendimento, podendo ser coletas ou entregas de produtos para um conjunto de clientes que possuem localização espacial definida e demanda conhecida. Cada cliente (ponto ou nó) deve ter sua demanda totalmente atendida por um único veículo. As rotas devem ser definidas de forma a não violar as restrições de tempo e de capacidade do veículo.

Segundo a definição de PARTKA e HALL (apud NOVAES*,* 2007*)* um problema de roteirização é definido por três fatores: decisões, objetivos e restrições. As decisões dizem respeito à seleção e agrupamento de clientes a serem visitados, quantidade de veículos e motoristas, programação e o sequenciamento de visitas. O principal objetivo é maximizar o nível de serviço prestado aos clientes, mantendo os custos mais baixos possíveis. E por último devem-se respeitar as restrições definidas na roteirização, que geralmente incluem: não

exceder o limite de recursos disponíveis, completar as rotas, respeitar limites de tempo de jornada de trabalho, limites de carga do veículo, restrições de tráfegos em determinadas ruas, entre muitas outras possíveis.

NOVAES (2007) cita alguns exemplos onde o processo de roteirização de veículos é frequentemente aplicado:

- Entrega em domicílio de produtos comprados na internet;
- Distribuição de CDs para as lojas de varejo;
- Distribuição de bebidas em bares e restaurantes;
- Distribuição de dinheiro para caixas eletrônicos e bancos;
- Distribuição de combustíveis para postos de gasolina;
- Coleta de lixo urbano;
- Entrega domiciliar de correspondência, entre outros.

Roteirização Sem Restrições ou Problema do Caixeiro Viajante - PCV

O conceito de "Roteirização sem Restrições", também chamado de Problema do Caixeiro Viajante - PCV na literatura técnica, descrito por NOVAES (2007), existe quando a separação de clientes, pelos diversos roteiros já foi previamente realizada e todas as restrições de tempo, capacidade, etc. já foram resolvidas. A partir desse momento, o problema a ser resolvido é o de encontrar a sequência de visitas que possua a menor distância possível dentro do bolsão. A Figura 17 mostra um exemplo de roteiro simples.

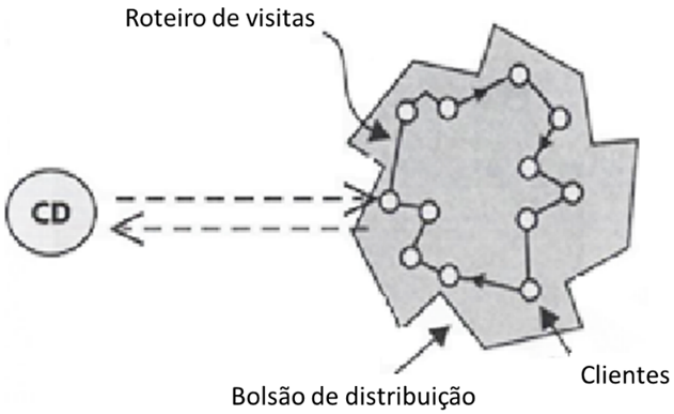


Figura 17 - Roteiro simples (12 clientes) num bolsão de distribuição.

Fonte: NOVAES, 2007.

Entre as diversas formas existentes para resolver um PCV, pode-se agrupar elas em duas categorias:

Categoria 1 – Métodos de Construção do Roteiro;
Categoria 2 – Métodos de Melhoria do Roteiro;

O objetivo dos métodos que pertencem a categoria 1 é de prover uma solução inicial que depois será revisada pelos métodos de melhoria da categoria 2.

Roteirização com Restrições

É comum realizar a roteirização de veículos sem que nenhuma definição de zonas ou bolsões tenha sido feita. Nesse cenário é preciso roteirizar ao mesmo tempo em que ocorre a divisão de zonas ou bolsões.

Segundo NOVAES (2007), muitos métodos para resolver esse tipo de roteirização envolvem modelos matemáticos razoavelmente complexos. O autor define que os métodos de *varredura* e de *Clarke e Wright,* os quais foram implementados no protótipo e que são relativamente simples, eficazes e largamente utilizados.

Como exemplo do nível de complexidade que pode chegar o tratamento da roteirização com restrições pode-se citar o trabalho de SILVA (2013) onde tratou o problema de roteirização dinâmica com janelas de tempo usando seis algoritmos

baseados no método de inserção, do vizinho mais próximo, de otimização por colônia de formigas, e das versões sequencial e aleatória do método de busca em vizinhança variável.

4.3.2 Procedimentos para solução de problemas de roteirização

4.3.2.1 Métodos de Construção de Roteiros

4.3.2.1.1 Método de Clarke e Wright - CW

O método de CW, também conhecido como o método de economias, possui bons resultados na aplicação de problemas de roteirização, sendo largamente usado em softwares de roteirização. Segundo LIU e SHEN (1999), o método criado por CLARKE e WRIGHT (1964) foi desenvolvido para resolver o problema clássico de roteirização de veículos.

NOVAES (2007) destaca o fato do método de CW possuir somente 2% de erro médio em relação ao ótimo, sendo um bom desempenho em comparação com outros métodos.

O objetivo do método é minimizar a distância e indiretamente, minimizar a quantidade de veículos usados. Além de ser relativamente de fácil implementação, o método é bem flexível e permite, por exemplo, a inclusão de diversas restrições sem prejudicar a qualidade dos resultados (NOVAES, 2007; BALLOU, 2006).

Ocasionalmente, o método CW pode gerar roteiros que se cruzam, que sugerem uma melhoria nos roteiros. Mas essa melhoria, em geral, pode ser facilmente alcançada através dos métodos de melhoria de roteiros.

O método de CW é baseado no conceito de ganho, que é definindo por *L*

$$L = d_{D,i} + d_{D,j} - d_{i,j} \tag{24}$$

Onde:

$\mathcal{D}$ – depósito, local de saída e retorno do veículo

i e j – dois pontos no mapa, ou seja, possíveis clientes

d – distância

L – ganho

O processo começa atribuindo um veículo a cada cliente, formando rotas simples entre o depósito para todos os clientes. Seja, $c_{i,j}$ o custo de viagem partindo de um cliente i a um cliente j ou $d_{i,j}$ a distância da viagem partindo de um cliente i a um clientes j, segundo definição de LIU e SHEN (1999), duas rotas contendo os clientes i e j podem ser combinadas, desde que i e j estejam ou na primeira ou na última posição de suas respectivas rotas e que a demanda total das rotas combinadas não ultrapasse a capacidade do veículo.

Em cada iteração, todas as combinações de rotas possíveis são analisadas através da fórmula de ganho. As duas rotas que renderem o maior ganho de combinação são unidas. Por ser sempre escolhida a maior economia dentre as possíveis, a função de escolha é dita gulosa. Como a cada nova combinação de subrotas as economias são novamente calculadas e atualizadas para a próxima combinação de subrotas, o método é dito iterativo, (LIU e SHEN, 1999). Na Figura 18 pode ser observado a evolução do método.

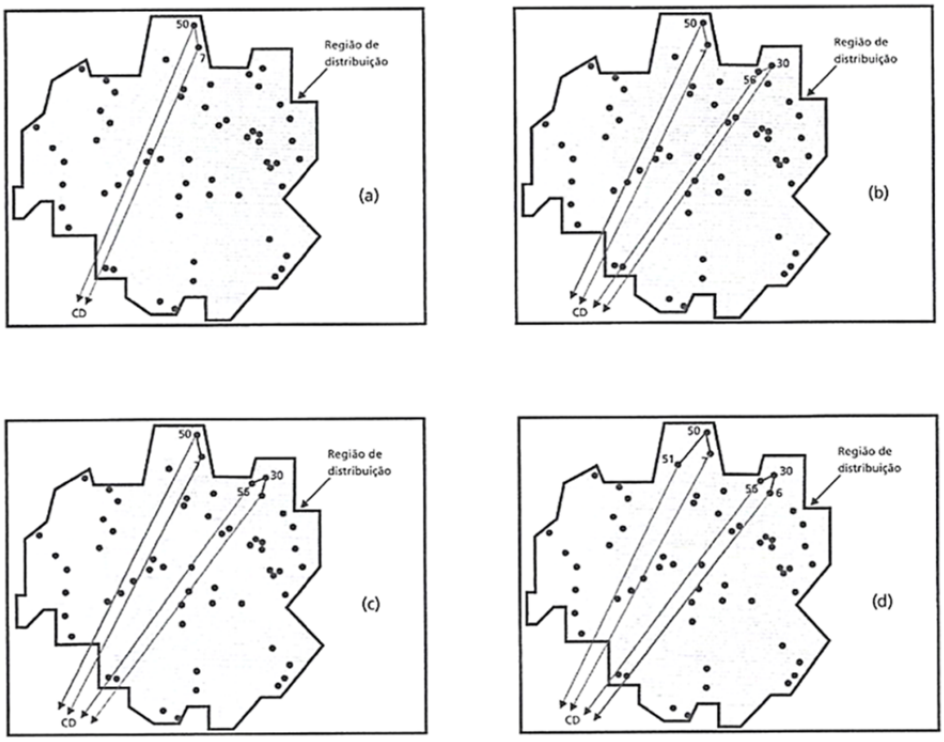


Figura 18 - Evolução do método de Clarke e Wright.

Fonte: NOVAES, 2007.

Descrição do processo do método de CW em seis etapas segundo Novaes (2007):

1: **Método de Clarke e Wright em seis etapas**

2: Etapa 1 – Combina-se todos os pontos (que representam os clientes) dois a dois e calcula-se o ganho para cada combinação.

3: Etapa 2 – Ordenam-se todas as combinações i,j de forma decrescente segundo os valores dos ganhos (L).

4: Etapa 3 – Começamos com a combinação de dois nós que apresentou o maior ganho. Posteriormente, na análise de outras situações, começa a percorrer a lista de combinações de cima para baixo, seguindo a ordem decrescente de ganhos.

5: Etapa 4 – Para um par de pontos (i,j), tirado da sequência de combinações, verifica-se se os dois pontos já fazem parte de um roteiro iniciado:

(a) Se i e j não foram incluídos em nenhum dos roteiros já iniciados, cria-se então um novo roteiro com esses dois pontos;

(b) Se o ponto i já pertence a um roteiro iniciado, verificar se esse ponto é o primeiro ou último desse roteiro (não contando o CD). Se a

resposta for positiva, acrescentar o par de pontos (i,j) na extremidade apropriada. Fazer a mesma análise com o ponto j. Se nenhum dos dois pontos satisfizer essa condição separadamente, passar para o item (c);

(c) Se ambos os pontos i e j fazem parte, cada um deles, de roteiros iniciados, mas diferentes, verificar se ambos são extremos dos respectivos roteiros. Se a resposta for positiva, fundir os dois roteiros num só, juntando-os de forma a unir i a j. Caso contrário, passar para a etapa 5.

(d) Se ambos os nós i e j pertencerem a um mesmo roteiro, passar para a etapa 5.

6: Etapa 5 – Cada vez que acrescentar um ou mais pontos num roteiro ou quando fundir dois roteiros num só, verificar se a nova configuração satisfaz as restrições de tempo e capacidade. Se atender aos limites das restrições, a nova configuração é aceita.

7: Etapa 6 – O processo termina quando todos os pontos (clientes) tiverem sido incluídos nos roteiros.

Figura 19 - Método de Clarke e Wright em seis etapas

O método de CW possui consegue reduzir a complexidade de resolução de um problema de roteirização de veículos, obtendo resultados bem próximos ao ótimo.

4.3.2.2 Métodos de Melhorias de Roteiros

O objetivo dos métodos de melhoria, como o nome da própria categoria já explicita é o de aperfeiçoar os roteiros resultantes dos métodos de construção de roteiro, por exemplo, reduzindo ainda mais a distância total percorrida no roteiro.

De acordo com NOVAES (2007) os dois métodos de melhoria de Roteiros mais utilizados são o *2-opt* e o *3-opt,* desenvolvidos por LIN e KERNIGHAN (1973).

4.3.1.2.1 Método 2-opt

O método consiste em tentativas sucessivas em formar dois arcos novos, ou seja, que não existiam no roteiro inicial, que promovam uma redução na extensão total do roteiro. A Figura 21 mostra uma iteração do método.

NOVAES (2007) descreve o processo de aplicação do método em três etapas:

1: **Metodo 2-opt em três etapas**

2: Etapa 1 - Seleciona-se um roteiro, de preferência que tenha sido gerado por um método de construção de roteiro.

3: Etapa 2 - Remove-se dois arcos do roteiro.

Inicia-se as tentativas de formar diferentes arcos com os quatro nós que pertenciam aos arcos removidos.

Caso encontre uma nova formação de arcos que diminua a extensão total do roteiro, adota-se essa formação como o roteiro principal e retorna a etapa 2.

Caso contrário, continua-se com o roteiro anterior e repete-se a etapa 2.

4: Etapa 3 - O processo termina ao testar todas as combinações de trocas possíveis.

Figura 20 - Método 2-opt em três etapas

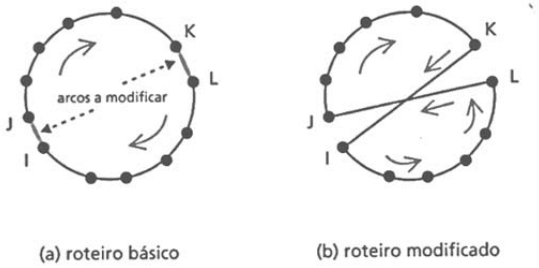


Figura 21 - Dois pares de nós (I-J e K-L) rearranjados no método 2-opt, para solução do PCV.

SILVA (2013) descreve que o método 2-opt consiste em remover um par de arcos e posteriormente, inserir outro par de arcos. Na Figura 22 contém o detalhamento do funcionamento do algoritmo feito pelo autor.

1: **Procedimento 2-opt(s)**

2: **Para** *r1* = 1... v **faça**

3:**Para** *i*=1 até o tamanho de *r1* ***faça***

4: **Para** j=tamanho de *r1* até 1 **faça**

5://Avaliar o custo *L* de uma solução *s'* vizinha de *s*, após remover os arcos (*i,i*+1) e (*j,j*-1) e inserir os arcos (*i,j*-1*)* e (i+1,*j)*

6: **Se** *Ls'* < *Ls** **E** *s'* é factível **então**

7: $s^* \leftarrow s'$

8: **Retorne** s

Figura 22 - Funcionamento do algoritmo 2-opt.

Fonte: SILVA, (2013).

4.4 Procedimento para problemas de formação de zonas

4.4.1 Procedimentos para formação de zonas - Zoneamento

Os procedimentos para formação de zonas podem ser executados antes da etapa de roteirização.

Segundo NOVAES (1989, apud LEAL, 2012), o procedimento de formação de zonas, basicamente, tem-se um depósito central localizado em certo ponto geográfico, que deve servir a uma zona de coleta ou distribuição atendendo as seguintes condições:

- A região geográfica é dividida em zonas;
- Cada zona será servida por um veículo e uma equipe;
- Os veículos partem do depósito e retornam após realizar o serviço;
- O serviço de cada veículo é descrito por um roteiro que inclui um número dado de locais de parada dentro da zona a ser servida, ordenados em uma certa sequência;
- O serviço, seja de coleta ou distribuição, demanda um tempo total desde a partida do veículo do depósito até sua volta, denominado tempo de ciclo.

O procedimento pode ser dividido em quatro passos básicos:

1: Procedimento de formação de zonas

2: Definição da divisão da região em zonas;

3: Seleção de veículos e da equipe para executar a tarefa;

Estimativas de parâmetros como a quilometragem média, o tempo de ciclo e os componentes dos tempos associados ao serviço além dos custos associados;

4: Parâmetros do nível de serviço como a fração de serviço não atendida.

Figura 23 - Procedimento de formação de zonas

4.4.1.1 Procedimento de formação de zonas por nível de serviço

O procedimento técnico para a formação de zonas envolve uma série de pressupostos, sendo assim é necessário introduzir alguns conceitos relacionados conforme abaixo.

Todas as definições e modelagens matemáticas que serão apresentadas adiante foram retiradas de notas de aula de LEAL (2012).

Os conceitos que serão abordados são:

- Tempo de Ciclo
- Quilometragem
- Distância Média Entre Pontos
- Restrições por Capacidade e Tempo
  - Restrição por capacidade Física
  - Restrição por Tempo
- Cálculo de Área Admissível na Restrição por Tempo
- Cálculo de Área Admissível na Restrição por Carga

4.4.1.1.1 Componentes envolvidos para o dimensionamento de zonas

Segue abaixo na Figura 24 um esquema gráfico abordando as variáveis envolvidas no procedimento.

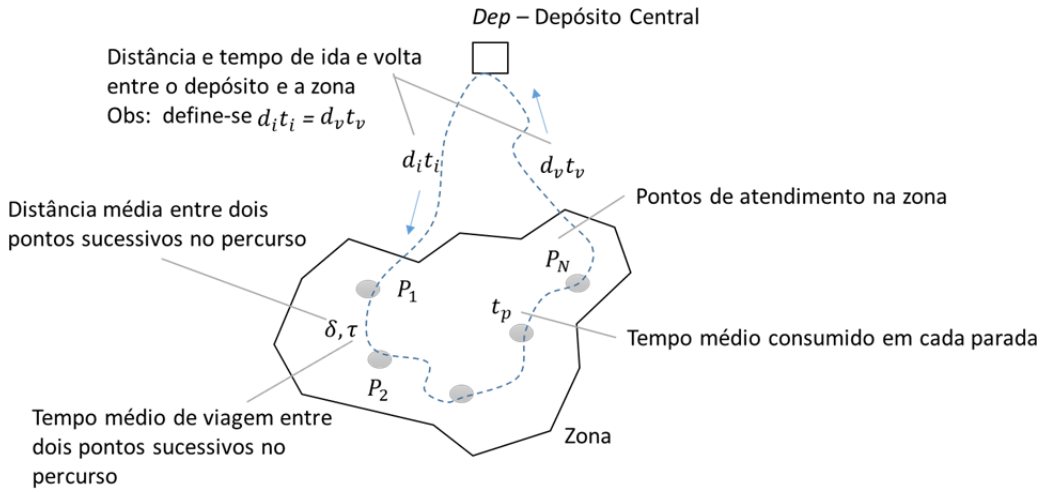


Figura 24 - Exemplo de um veículo saindo do depósito e percorrendo os pontos de atendimento numa zona e seus componentes.

Fonte: Adaptado LEAL, 2012.

4.4.1.1.1.1 Tempo de Ciclo - TC

O tempo de ciclo pode ser obtido com a seguinte fórmula:

$$TC = t + T^*_P + T^*_\tau + t \quad 25)$$

Onde:

- T^*_p é o somatório do tempo de todas as paradas na zona.

É uma variável aleatória, pois o número de pontos N varia aleatoriamente, assim como o tempo consumido em cada parada.

$$T^*_P = t_p^{(1)} + t_p^{(2)} + \cdots + t_p^{(N)} \quad 26)$$

- N é uma variável aleatória com média $E[N]$ e variância $Var[N]$.

- t_p (tempo de parada) é uma variável aleatória com média $E[t_p]$ e variância $Var\ [t_p]$.

Sendo assim, pode-se estimar a média e a variância da variável T^*_p que é composta por duas variáveis aleatórias:

$$E[T^*_p] = E[N] + E[t_p] \tag{27}$$

$$Var\ [T^*_p] = E[N]\ \times Var\ [t_p] + E^2[t_p]\ \times Var\ [N] \tag{28}$$

Este tipo de fórmula proposta por NOVAES (1989) se aplica a estes tipos de variáveis compostas do produto de duas outras variáveis aleatórias.

Analogamente podemos estimar a média e a variância do tempo total de percurso $E\ [T^*_\tau]$ e Var $[T^*_\tau]$, respectivamente.

$$E[T^*_\tau] = E[N] \times E[\tau] \tag{29}$$

$$Var[T^*_\tau] = E[N] \times Var\ [\tau] + E^2[\tau]\ \times Var\ [N] \tag{30}$$

Assim, a média e a variância do TC dentro da zona pode ser estimado conforme abaixo:

$$E[TC] = 2 \times E[t] + E[\ T^*_p] + E[T^*_\tau] \tag{31}$$

$$Var[TC] = 2 \times Var\ [t] + Var[T^*_p]\ \times Var[T^*_\tau] \tag{32}$$

4.4.1.1.1.2 Quilometragem

A quilometragem total percorrida dentro da zona pode ser definida pela seguinte fórmula:

$$DC = d + D^*_{\delta} + d \quad 33)$$

Porém trata-se aqui também de uma variável aleatória, já que o número de pontos e a distância de percurso entre pontos são pontos são variáveis aleatórias.

Seja:

- $E\,[T^*_{\tau}]$ valor esperado da distância total percorrida dentro da zona

- $Var\,[T^*_{\tau}]$ variância da distância total percorrida dentro da zona

A distância total percorrida dentro da zona é a soma das distâncias entre paradas, derivada do número médio de pontos e da distância média entre pontos.

$$E[\,D^*_{\delta}] = E[N] \times E[\delta] \quad 34)$$

Logo a variância é definida como:

$$Var[\,D^*_{\delta}] = E[N] \times Var[\delta] + E^2[\delta] \times Var[N] \quad 35)$$

Para encontrar a média e variância da quilometragem total percorrida, deve-se agora, acrescentar as distâncias de ida e de volta da zona.

Seja:

- E[d] distância média de ida ou de volta entre a zona e o depósito (di = dv)

- E [DC] valor esperado da distância total percorrida

- Var [DC] variância da distância total percorrida

Então:

$$E[DC] = 2 \times E[d] + E[D^*_{\delta}] \quad (36)$$

$$Var[DC] = 2 \times Var[d] + Var[D^*_{\delta}] \quad (37)$$

Assim, chega-se ao valor esperado e da variância da distância total percorrida no serviço de uma zona.

A fórmula anterior pressupõe o conhecimento da distância entre pontos que será apresentado a seguir.

4.4.1.1.1.3 Distância Média Entre Pontos

A distribuição das distâncias entre um ponto qualquer e o seu vizinho mais próximo segue uma distribuição de Poisson.

Conforme definido por STEIN (apud LEAL 2012), a fórmula para encontrar a distância média para pontos distribuídos segundo uma distribuição de Poisson com densidade λ, segue conforme abaixo:

Seja a distância média dada pela fórmula:

$$\bar{\delta} = k \times \lambda^{-\frac{1}{2}} \quad \text{com: K} \geq 0{,}5 \quad (38)$$

STEIN parte de *L*, que seria a extensão total do roteiro dentro de uma zona com *N* pontos. Ele define:

$$\lim_{N \to \infty} \left(\frac{E[L]}{\sqrt{N}} \right) = k \times \sqrt{A} \quad (39)$$

Como *k* sendo uma constante e *A* (área compacta e convexa), para STEIN: k≈ 0,765.

$$E[L] \approx k \times \sqrt{NA} \tag{40}$$

Mas sabe-se também que:

$$E[L] = N \times \bar{\delta} \tag{41}$$

$$\text{Logo} \rightarrow \bar{\delta} = k \times \sqrt{\frac{A}{N}} = \ k \ \times \lambda^{-\frac{1}{2}} \tag{42}$$

A densidade pode ser expressa como:

$$\lambda = \frac{N}{A} \tag{43}$$

Então a fórmula de distância média entre pontos de assemelha a de distância média ao ponto mais próximo:

$$\bar{\delta} = 0{,}765 \times \lambda^{-\frac{1}{2}} \ \approx \ \bar{d} = 0{,}5 \times \lambda^{-\frac{1}{2}} \tag{44}$$

A diferença está na constante das fórmulas.

4.4.1.1.1.4 Restrições por Capacidade e Tempo

Em problemas de distribuição, devido as diversas incertezas como: clima, tráfego, atrasos de clientes, problemas com o veículo, entre outros, realizar uma

abordagem estocástica aproxima o modelo planejado a realidade do dia a dia de distribuição.

Sendo assim, devem-se considerar três aspectos no dimensionamento da distribuição:

1. Capacidade Física dos Veículos – Dada a variabilidade da carga a ser transportada.
2. Tempo máximo de jornada de tripulantes – A tripulação (motorista e ajudantes) tem limitações de tempo de trabalho definidas por leis e sindicatos.
3. Desequilíbrio de produção entre veículos em zonas mais próximas e mais longínquas – Deve-se tentar equilibrar a carga de trabalho entre as zonas.

4.4.1.1.1.4.1 Restrição por Capacidade Física

O objetivo é incluir na modelagem a restrição em relação ao peso ou ao volume da carga e a variabilidade associada.

Seja:

- ∇ a capacidade útil do veículo

- U_i o volume associado ao *i*-ésimo cliente. *U* é uma variável aleatória com média $E[U_i]$ e variância $Var[U]$

- W a carga útil total ocupada pelas cargas de todos os clientes. *W* é uma variável aleatória com média $E[W]$ e variância $Var[W]$

Onde:

$$W = U_1 + U_2 + U_3 + \cdots + U_N$$

45)

$$E[W] = E[N] \times E[U]$$

46)

$$Var[W] = E[N] \times Var[U] + E^2[U] \times Var[N]$$

47)

Devido as diferentes formas das embalagens transportadas, plano de carga, sequenciamento de clientes e outros fatores, frequentemente ocorre o problema de quebra de estiva, que acaba reduzindo a capacidade total do veículo. Portanto, deve-se considerar também no dimensionamento um percentual (p) referente à redução de capacidade do veículo.

Seja:

- p valor em percentual de perda de capacidade

- R coeficiente de redução de capacidade

Então:

$$R = 1 - \frac{p}{100}$$

48)

Para a carga transportada, vale a restrição de capacidade:

$$W \leq R\nabla$$

49)

Supõe-se que W segue uma distribuição normal conforme a Figura 25. A média de carga total dos cliente é $E[W]$. A figura mostra o valor onde W iguala a capacidade útil do veículo.

O número de ocorrências de não atendimento dos clientes está associado à área rabiscada. O objetivo da análise é calcular o nível de serviço associado ao atendimento, como a probabilidade de que a capacidade do veículo seja ultrapassada.

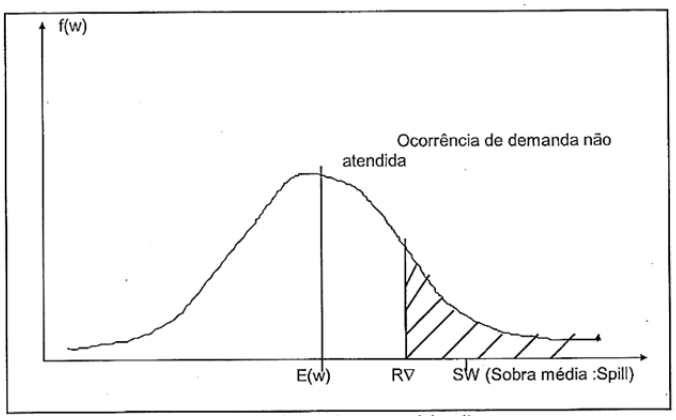


Figura 25 - Distribuição de carga total dos clientes.

Fonte: LEAL, 2012.

Para realizar a análise, trabalha-se com a distribuição normal normalizada, transformando a variável original em uma variável padronizada η.

Aplicando a tabela de distribuição normal obtém-se f(η), $\phi(\eta)$ *e* $\phi^*(\eta)$, onde:

- f(η) é a ordenada do gráfico

- $\phi(\eta)$ área da distribuição normal a esquerda do ponto η

- $\phi^*(\eta)$ área da distribuição normal a direita do ponto η

A variável original a ser transformada é W e o valor da carga a ser comparado é $R\nabla$ (capacidade real do veículo). Sendo assim, a variável padronizada é definida por:

$$\eta = \frac{R\nabla - \mathrm{E[W]}}{\sigma_W} \tag{50}$$

Onde:

$$\sigma_W = \sqrt{Var[W]} \tag{51}$$

Entrando com o valor de η na tabela, obtém-se os valores de f(η) $e\ \phi^*(\eta)$. Este último indica a probabilidade da capacidade ser ultrapassada pela demanda W.

Pode-se calcular também de forma inversa, verificando qual seria a capacidade necessária para atender a um determinado nível de serviço. Nesse caso, a fórmula para encontrar a capacidade que respeita o nível de serviço seria:

$$Então:\ \ R\nabla = \eta \times \sigma_W + E[W] \tag{52}$$

$$Então:\quad R = \frac{\eta \times \sigma_W + E[W]}{\nabla} \tag{53}$$

Da mesma forma, pode-se também a partir de uma capacidade fixa, calcular o número de clientes que podem ser atendidos, respeitando o nível de serviço definido previamente. Para isso, deve-se recordar que:

$$E[W] = E[N] \times E[U] \quad e \tag{54}$$

$$Var[W] = E[N] \times Var[U] + E^2[U] \times Var[N], \quad ou \tag{55}$$

$$Var[W] = E[N] \times (Var[U] + E^2[U]), \tag{56}$$

se N seguir a distribuição de Poisson.

Pode-se escrever em função de *N*:

$$\eta = \frac{R\nabla - E[N] \times E[U]}{\sqrt{E[N] \times (Var[U] + E^2[U])}} \tag{57}$$

Para simplificar o nome das variáveis, fazemos:

$$U = E[U] \qquad N = E[N] \qquad VarU = Var[U] \qquad Cap = R\nabla \tag{58}$$

Rearranjando a fórmula, por fim temos:

$$a = U^2 \quad b = \eta^2 \times (VarU + U^2) + 2 \times Cap \times U \quad c = Cap^2 \tag{59}$$

Resolve-se a equação de segundo grau chegando a dois valores de *N*, que correspondem ao valor absoluto de η definido.

É importante notar que embora existam dois valores na resposta, apenas o menor valor entre eles atende ao nível de serviço desejado, portanto é o valor que deve ser considerado.

4.4.1.1.1.4.2 Restrição por Tempo - Limitação de Tempo de Ciclo por Jornada de Trabalho

A análise da consequência da limitação por jornada de trabalho sobre o serviço de distribuição parte da definição dos seguintes limites:

Seja:

- H_0 jornada normal de trabalho, em horas.

- H_1 jornada máxima de trabalho, em horas. Este valor inclui as horas normais e as horas extras.

Seja *t*, o tempo total gasto no serviço de distribuição, têm-se três possíveis situações:

1. Nível Normal: $t \leq H_0$. Onde o tempo total gasto está abaixo ou igual ao limite da jornada.

2. Serviço com Horas Extras: $H_0 < t \leq H_1$. Onde o tempo total gasto ultrapassou a jornada normal, porém ainda está dentro do tolerado devido as horas extras.

3. Nível Crítico: $t > H_1$. Onde o tempo total ultrapassou a jornada máxima permitida. Não pode ser aceito.

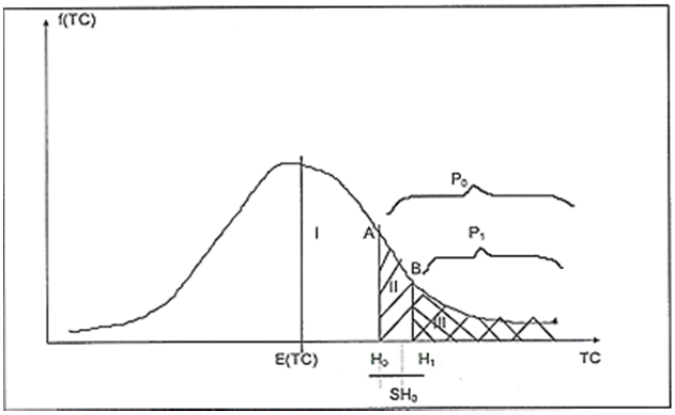


Figura 26 - Distribuição do tempo de ciclo.

Fonte: LEAL, 2012.

Na Figura 26 acima, seja:

- P_0 indica a probabilidade do tempo de ciclo ultrapassar H_0

- P_1 indica a probabilidade do tempo de ciclo ultrapassar H_1

Como o serviço completo implica em cumprir o tempo de ciclo, P_0 e P_1 podem ser vistos como níveis de serviço, quando se trabalha respectivamente com tempos de ciclo, sem usar horas extras H_0 e usando horas extras H_1.

$$P_0 = \int_{\eta_0}^{\infty} f(x)dx = 1 - \int_{-\infty}^{\eta_0} f(x)dx = 1 - \emptyset(\eta_0) = \emptyset^*(\eta_0) \qquad 60)$$

$$P_1 = \int_{\eta_1}^{\infty} f(x)dx = 1 - \emptyset(\eta_1) = \emptyset^*(\eta_1) \qquad 61)$$

Sendo η o valor da variável padronizada, a ser usada na distribuição normal normalizada.

$$\eta = \frac{H - E[TC]}{\sigma_{TC}} \tag{62}$$

Aplicando os valores de H_0 *e* H_1, obtêm-se respectivamente os valores de η_0 e η_1 e quando aplicados na tabela de distribuição normal normalizada, obtém-se as ordenadas e as probabilidades P_0 e P_1.

Da mesma maneira que foi realizado anteriormente com a restrição de carga, também podemos realizar o cálculo inverso e achar o tempo de ciclo a partir de um determinado nível de serviço.

Supondo que σ_{TC} é fixo, pode-se calcular *E(TC)*.

$$E[TC] = H - \eta \times \sigma_{TC} \tag{63}$$

Para calcular a área de serviço que corresponde a um nível de serviço dado é necessário reescrever as fórmulas em função de área e da densidade de pontos por área e período.

A fórmula do tempo de ciclo para a distribuição para uma zona desde um depósito pode ser escrita em função da área e da densidade de pontos por km^2 e período.

Seja:

$$N = A \times \lambda \;\; e \;\; \bar{\delta} = 0{,}765 \times \lambda^{-\frac{1}{2}}, \tag{64}$$

$$pode\ escrever:$$

$$TC = 2 \times t + A \times \lambda \times \frac{0{,}765}{\sqrt{\lambda} \times v} + A \times \lambda \times tp \quad 65)$$

$$TC = 2 \times t + A \times (\lambda \times \frac{0{,}765}{\sqrt{\lambda} \times v} + \lambda \times tp) \quad 66)$$

Supondo que *N* segue uma distribuição de Poisson, então o σ_{TC} pode ser escrito como:

$$\eta = \frac{H - E[TC]}{\sigma_{TC}} \quad 67)$$

$$\sigma_{TC} = \sqrt{2 \times var(t) + A \times \lambda \times \left(\tau^2 + var(\tau)\right) + A \times \lambda \times (tp + var(tp))} \quad 68)$$

Supondo um coeficiente de variação da distância média entre pontos igual a 0,523, então:

$$\sigma_\tau = 0{,}523 \times \frac{0{,}765}{\sqrt{\lambda} \times v} = \frac{0{,}4}{\sqrt{\lambda} \times v} \; logo : var_\tau = \frac{0{,}16}{\lambda \times v^2} \quad 69)$$

Sendo v a velocidade média de viagem dentro da zona.

Usando *E [*τ] *e var* [τ], pode-se estimar o σ_{TC} como:

$$\sigma_{TC} = \sqrt{2 \times var(t) + A \times \left(\frac{0{,}745}{v^2} + \lambda \times (tp^2 + var(tp))\right)} \quad 70)$$

Chamando:

$$K1 = \sqrt{\lambda} \times \frac{0{,}765}{v} + \lambda \times tp \quad 71)$$

$$K2 = \frac{0{,}745}{v^2} + \lambda \times (tp^2 \times var(tp)) \quad 72)$$

Tomando os valores de tempo em minutos e a velocidade em Km/h, tem-se que definir constantes de conversão de unidades, como:

$$K1 = \sqrt{\lambda} \times \frac{0{,}745 \times 60}{v} + \lambda \times tp \quad \text{e} \quad K2 = (\frac{0{,}745 \times 3600}{v^2} + \lambda \times (tp^2 \times var(tp))) \quad 73)$$

Definindo um nível de serviço, estas fórmulas podem ser usadas para estimar a área admissível em um serviço de distribuição para garantir uma jornada sem horas extras (dentro de H_0) ou com horas extras (dentro de H_1). Nível de serviço vai estar associado a uma variável padronizada η. Com ela. Pode-se obter *TC* admissível para garantir o nível de serviço.

$$\eta = \frac{H - E[TC]}{\sigma_{TC}} \quad ou \quad TC = H + \eta \times \sigma_{TC} \quad 74)$$

Usando *K1 e K2* como definido anteriormente, e colocando as horas em minutos, tem-se:

$$A \times K1 - (H \times 60 - 2 \times t) = \eta \times \sqrt{2 \times var(t) + A \times K2} \quad 75)$$

Elevando-se ambos os termos ao quadrado e rearranjando:

$$A^2 \times K1^2 - A \times (2 \times K1 \times (H \times 60 - 2 \times t) + \eta^2 \times K2) + (H \times 60 - 2 \times t)^2 - \eta^2 \times 2 \times var(t) = 0 \quad 76)$$

Chamando:

$$a = k1^2 \quad b = -(2 \times k1 \times (H \times 60 - 2 \times t) + \eta^2 \times k2)$$

77)

$$\text{e} \qquad c = (H \times 60 - 2 \times t)^2 - \eta^2 \times 2 \times var(t)$$

78)

Encontra-se o valor da área que atende ao valor de η, correspondente ao nível de serviço desejado, resolvendo a equação de segundo grau:

$$A = \frac{-b \pm \sqrt{b^2 - 4 \times a \times c}}{2 \times a} \qquad 79)$$

Assim, como aplicado no caso da restrição por carga, dos dois valores que forem gerados da fórmula acima, deve-se considerar sempre o menor valor. Pois ele que irá garantir o nível de serviço definido.

4.4.1.1.1.5 Divisão da região em zonas

A divisão da região em zonas está ligada diretamente ao nível de serviço estabelecido pelas restrições de tempo e capacidade que foram vistas anteriormente. A partir do nível de serviço definido, encontra-se pela distribuição normal normalizada o valor de η (variável normal padronizada). Este valor é usado para calcular a área admissível segundo as restrições de tempo e capacidade.

Resumindo, o cálculo da área admissível por tempo seria:

Define-se:

$$K1 = \sqrt{\lambda} \times \frac{0{,}745}{v} + \lambda \times tp \quad \text{e} \quad K2 = \frac{0{,}745}{v^2} + \lambda \times (tp^2 \times var(tp)) \qquad 80)$$

Chamando:

$$a = k1^2 \qquad b = -(2 \times k1 \times (H \times 60 - 2 \times t) + \eta^2 \times k2) \tag{81}$$

$$c = (H \times 60 - 2 \times t)^2 - \eta^2 \times 2 \times var(t) \tag{82}$$

$$\text{área admissível por tempo} = A = \frac{-b - \sqrt{b^2 - 4 \times a \times c}}{2 \times a} \tag{83}$$

E o cálculo da área admissível por carga seria:

Define-se:

$$a = U^2 \quad b = \eta^2 \times (VarU + U^2) + 2 \times Cap \times U \quad c = Cap^2 \tag{84}$$

$$N = \frac{-b - \sqrt{b^2 - 4 \times a \times c}}{2 \times a} \tag{85}$$

$$\text{área admissível por carga} = A = \frac{N}{\lambda} \tag{86}$$

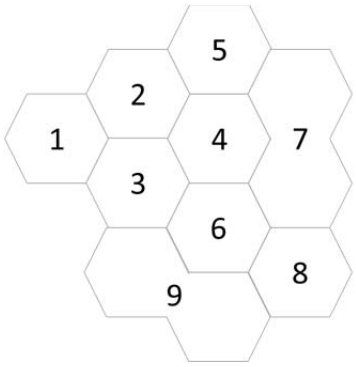
Para definir o melhor valor de área admissível é recomendável que se calcule por tempo e por carga e considere o menor valor entre eles.

Procedimento de formação de zonas – Zoneamento

Segundo LEAL (2012) o procedimento inicia-se selecionando uma região dividida em distritos, com área de cada distrito, valores de densidade e de distância ao depósito, que vão se refletir em tempos de viagem depósito-zona.

4.4.1.1.1.5.1 Zoneamento por Vizinhança

Esse método é baseado numa matriz quadrada de nXn, sendo n o número de distritos. A matriz pode receber os valores 0 ou 1. Caso o distrito i seja vizinho do vizinho j a posição na matriz tem o valor 1, caso contrário o valor é 0. Após formada a matriz, inicia-se um procedimento "guloso", agregando os distritos até o limite da área total admissível.



	Distrito 1	Distrito 2	Distrito 3	Distrito 4	Distrito 5	Distrito 6	Distrito 7	Distrito 8	Distrito 9
Distrito 1	0	1	1	0	0	0	0	0	0
Distrito 2	1	0	1	1	1	0	0	0	0
Distrito 3	1	1	0	1	0	1	0	0	1
Distrito 4	0	1	1	0	1	1	1	0	0
Distrito 5	0	1	0	1	0	0	1	0	0
Distrito 6	0	1	1	1	0	0	1	1	1
Distrito 7	0	0	0	1	1	0	0	1	0
Distrito 8	0	0	0	0	0	1	1	0	1
Distrito 9	0	0	1	0	0	1	0	1	0

Figura 27 – Informações sobre distritos e exemplo de uma matriz de vizinhança.

Fonte: Adaptado de LEAL, 2012.

O procedimento "guloso" de formação de zonas é caracterizado por agregar distritos, a partir de um distrito qualquer selecionado, até que se atinja a área admissível.

Considerando a Figura 27, uma área admissível de 4 e começando pelo distrito 1 tem-se:

Vizinhos de 1 são: 2 e 3 Total da área: 1,7

Vizinhos de 2 são: 4 e 5 Total da área: 2,9

Vizinhos de 3 são: 6 e 9 Total da área: 3,9

Vizinho de 4 é: 7 Total da área: 4,58. Não é admissível.

Os distritos 7 e 8 que sobraram, poderiam ser agregados em outro setor vizinho, ou definidos como uma zona de área 1,4.

Para melhorar a resposta final, pode-se aplicar um procedimento de troca de distritos até melhorar o valor final.

4.4.1.1.1.5.2 Zoneamento equilibrado

Para reduzir a necessidade de ajustes após a primeira divisão, pode-se, a priori, estimar a área de cada zona, a partir da área total da região e da área admissível por zona.

Calcula-se o número de zonas com a fórmula:

$$número\ de\ zonas = NZ = \left[\frac{AT}{A^*}\right]^+ \quad 87)$$

Onde:

- AT: área total da zona;

- A^*: área admissível da zona;

- $[x]^+$: inteiro maior ou igual a x;

A área média por zona é então:

$$área\ média\ por\ zona = AM = \left[\frac{AT}{NZ}\right] \quad 88)$$

Este procedimento garante uma distribuição mais equilibrada. Eventualmente, procedimentos de melhorias podem ainda ser aplicados para encontrar uma solução melhor.

4.4.1.1.2 Formação de Zonas

4.4.1.1.2.1 Varredura Sobre Eixo

O objetivo do procedimento é de agrupar distritos, pré-definidos, de modo a formar subzonas. O procedimento é "guloso", ou seja, agrupam-se distritos até que o somatório das áreas dos distritos agrupados atinja um determinado valor limite. Esse valor limite é a área admissível definida.

Informações necessárias para execução do procedimento:

- Área admissível
- Mapa que contenha os distritos
- Ângulo de abertura do triangulo
- Distrito depósito

Os componentes gerais envolvidos no procedimento podem ser observados na Figura 28 abaixo.

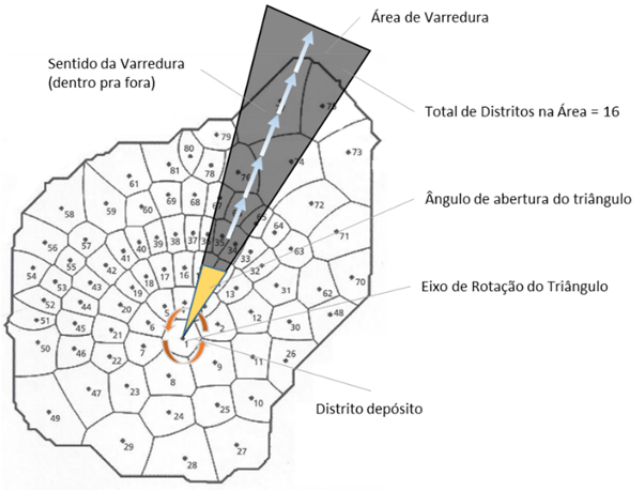


Figura 28 - Componentes envolvidos no procedimento de formação de zonas varredura sobre eixo.

Segue abaixo a descrição das etapas de funcionamento do procedimento:

1: **Procedimento de varredura sobre eixo para formação de zonas**

2: Etapa 1 – Verificar se existe algum distrito no mapa no qual sua área seja igual ou maior do que a área admissível definida. Caso exista, retira-se esses distritos do mapa para posteriormente receber um tratamento diferente.

3: Etapa 2 – Inicia-se a construção de um polígono triângulo. A base de rotação do triângulo deverá estar localizada no centroide do distrito definido como depósito. Para definir o comprimento do triângulo, verifica-se a distância entre o centroide do depósito e do distrito mais longe do mesmo, essa distância será o comprimento do triângulo.

4: Etapa 3 – Com o triângulo construído, verifica-se quais distritos possuem interseção com o triângulo e constrói-se uma lista com eles.

5: Etapa 4 – Com a lista de distritos que possuem interseção, ordena-se a mesma em ordem crescente pelo critério de distância entre cada distrito e distrito-depósito.

6: Etapa 5 – Com a lista ordenada, inicia-se a formação de uma subzona agregando os distritos até que se atinja o valor da área admissível definida. Após atingir o valor da área admissível, inicia-se a construção de uma nova subzona a partir do distrito imediatamente posterior ao último distrito selecionado na última subzona.

7: Etapa 6 – Encerra-se a formação da subzona caso não tenha mais distritos disponíveis. Caso a última subzona não consiga atingir o valor limite para formação de uma subzona, deve-se considerar uma subzona.

8: Etapa 7 – Rotacionar o triângulo na mesma quantidade de graus definida para o ângulo de abertura do triângulo e retorna-se a Etapa 3. Encerra-se essa etapa quando a rotação do triângulo atingir 360 graus.

Figura 29 - Procedimento de varredura sobre eixo para formação de zonas

A evolução gráfica do procedimento de formação de zonas por varredura sobre eixo pode ser observado conforme abaixo na Figura 30.

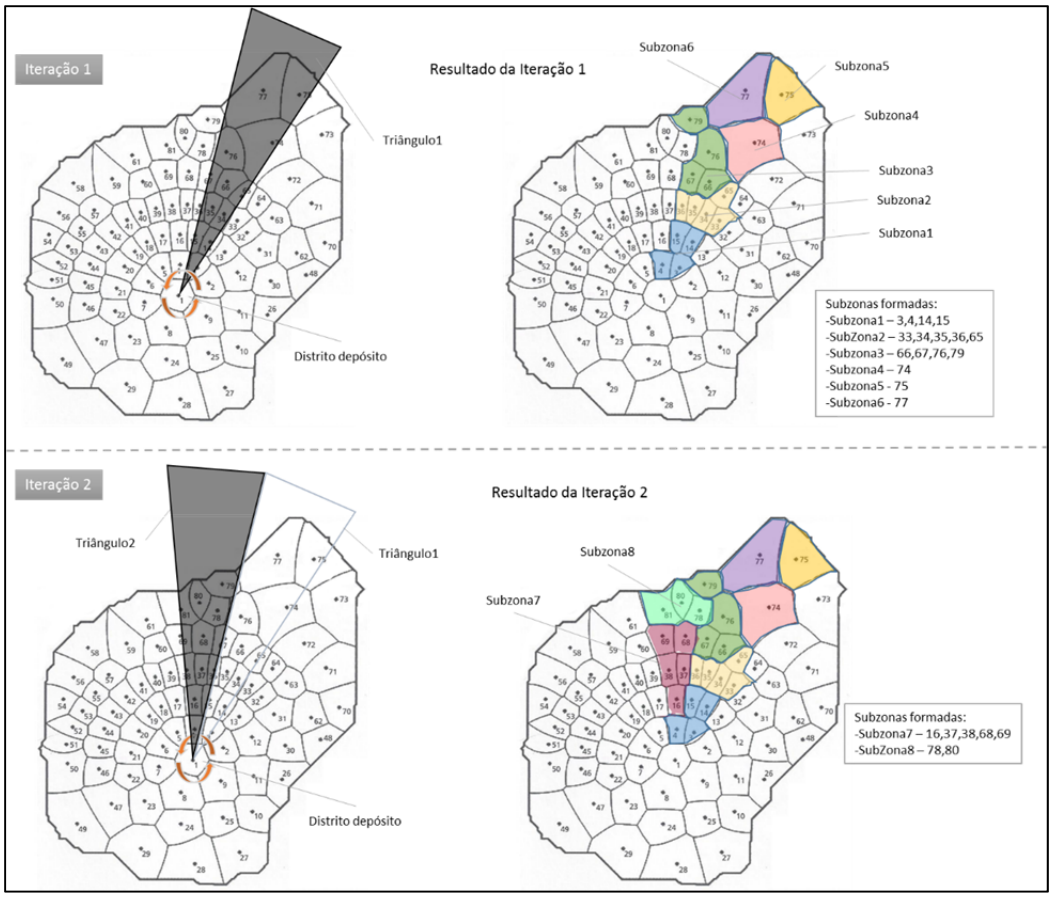


Figura 30 - Evolução do procedimento de formação de zonas por varredura sobre eixo.

4.4.1.1.2.2 Varredura Sobre Eixo Setorizado

O procedimento de formação de zonas por varredura sobre eixo setorizado é parecido com o mostrado anteriormente. Porém, nesse procedimento o triângulo usado para demarcar a região selecionada para construção de zonas será segmentado numa quantidade de setores a ser definida. A Figura 31 abaixo apresenta uma representação do triângulo dividido em setores.

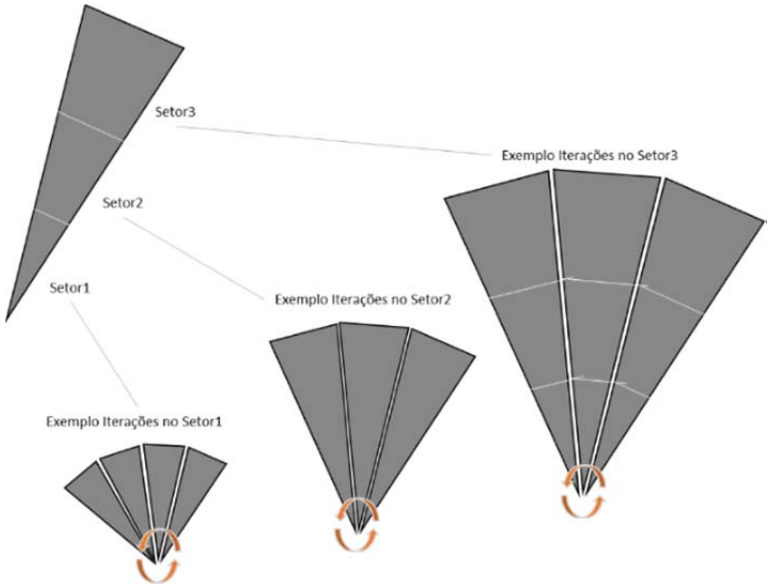


Figura 31 - Exemplo de triângulo dividido em três setores.

Ao invés de construir somente um triângulo, com um comprimento definido e rotacioná-lo como realizado no procedimento anterior, agora cada setor será rotacionado, partindo do setor mais próximo da base de rotação, por exemplo, no caso da Figura 31 seria o setor 1. E após a rotação do primeiro setor atingir 360 graus, inicia-se a rotação do setor imediatamente maior, até que o mesmo atinja também 360 graus de rotação e novamente seleciona-se o setor imediatamente maior até não existir mais setores.

Segue abaixo a descrição das etapas de funcionamento do procedimento conforme a Figura 33. As etapas são similares ao procedimento anterior com exceção das etapas 2 e 7 que sofreram uma pequena modificação e das etapas 2.a e 8 que foram adicionadas.

1: **Procedimento varredura sobre eixo setorizado para formação de zonas**

2: Etapa 1 – Verificar se existe algum distrito no mapa no qual sua área seja igual ou maior do que a área admissível definida. Caso exista, retira-se esses distritos do mapa para posteriormente receber um tratamento diferente.

3: Etapa 2 – Inicia-se a construção de um polígono triângulo. A base de rotação do triângulo deverá estar localizada no centroide do distrito definido como depósito. Para definir o comprimento do triângulo, verifica-se a distância entre o centroide do depósito e do distrito mais longe do mesmo e divide pela quantidade de setores definidos. O resultado dessa divisão será o valor incremental de comprimento para cada setor.

4: Etapa 2.a – Cria-se o triângulo até o limite de comprimento do setor.

5: Etapa 3 – Com o triângulo construído, verifica-se quais distritos possuem interseção com o triângulo e constrói-se uma lista com eles.

6: Etapa 4 – Com a lista de distritos que possuem interseção, ordena-se a mesma em ordem crescente pelo critério de distância entre cada distrito e distrito-depósito.

7: Etapa 5 – Com a lista ordenada, inicia-se a formação de uma subzona agregando os distritos até que se atinja o valor da área admissível definida. Após atingir o valor da área admissível, inicia-se a construção de uma nova subzona a partir do distrito imediatamente posterior ao último distrito selecionado na última subzona.

8: Etapa 6 – Encerra-se a formação da subzona caso não tenha mais distritos disponíveis. Caso a última subzona não consiga atingir o valor limite para formação de uma subzona, deve-se considerar uma subzona.

9: Etapa 7 – Rotaciona-se o triângulo na mesma quantidade de graus definida para o ângulo de abertura do triângulo e retorna-se a Etapa 3.

10: Etapa 8 - Quando a rotação de triângulos do setor atingir 360 graus, aumenta-se o comprimento do triângulo até o limite do setor imediatamente superior e retorna-se a Etapa 2.a. Encerra-se o procedimento quando não houver mais setores a serem verificados.

Figura 32 - Procedimento varredura sobre eixo setorizado para formação de zonas

A evolução gráfica do procedimento pode ser observada na Figura 33 e na Figura 34 abaixo:

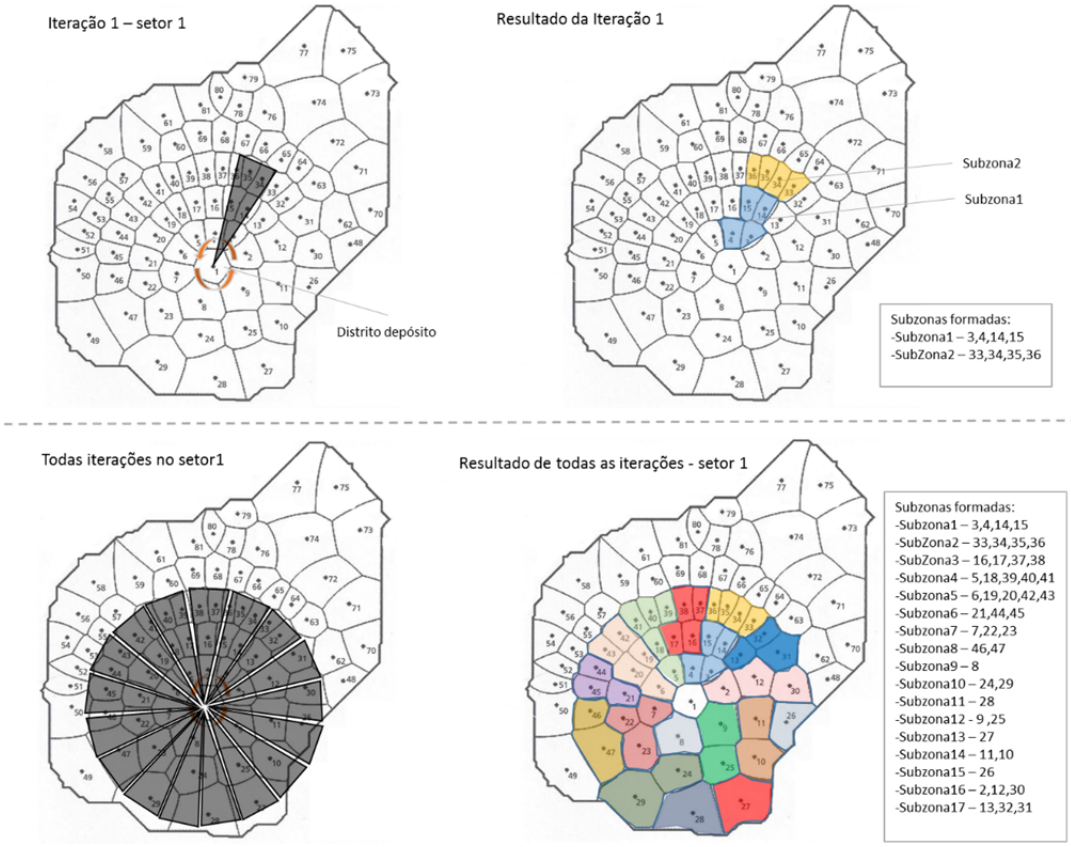


Figura 33 - Evolução do procedimento de varredura sobre eixo setorizado dentro do setor 1.

A Figura 35 apresenta o exemplo de uma zona que teve o método aplicado com uma definição de três setores. Pode-se observar a sequência de varredura do método até abranger toda a zona, partindo do setor mais próximo do depósito e incrementando até alcançar o último setor.

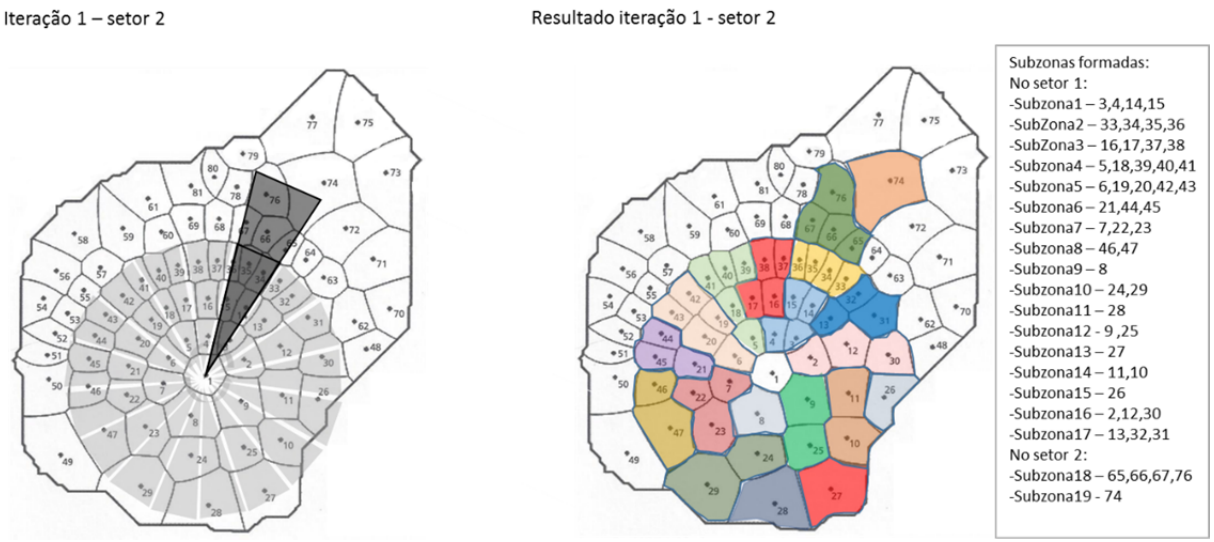


Figura 34 - Evolução do procedimento de varredura sobre eixo setorizado dentro do setor 2.

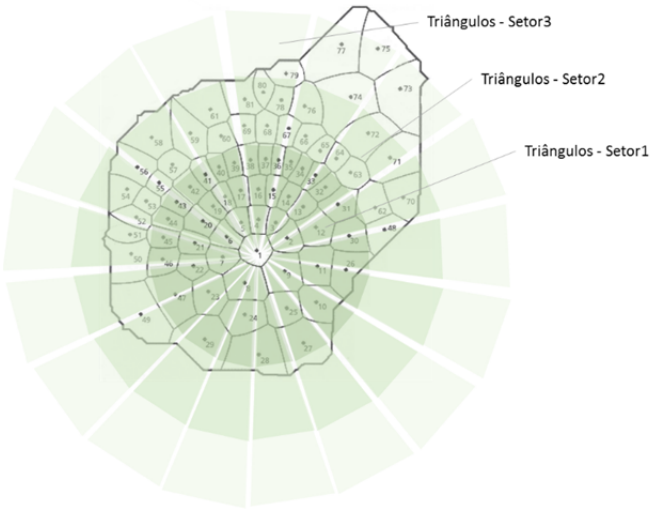


Figura 35 - Visão final dos triângulos gerados pelo procedimento de varredura sobre eixo setorizado.

4.4.1.2 Procedimento de formação de zona com Clarke e Wright

4.4.1.2.1 Método de Clarke e Wright com Parâmetro de Forma – CWPF

Existe uma variação do método de CW, que definiremos como método de Clarke e Wright com parâmetro de forma, que por sua vez é mais difundido na literatura do que a própria forma original do método proposta, que possui uma pequena diferença em relação ao cálculo de ganho, pois adiciona um parâmetro de forma, representado aqui pelo fator α, na parte da fórmula que reduz o ganho, com o objetivo de modificar a forma geométrica da solução, podendo ser mais agrupada ou mais dispersa.

YELLOW (1970) em seu trabalho sobre a modificação no método de economias (método de Clarke e Wright) propõe que o fator α pode variar de 1 a 2 dependendo de qual seja o objetivo a ser alcançado.

O cálculo de ganho para o método CWPF é estabelecido da seguinte forma:

$$L = d_{D,i} + d_{D,j} - \alpha \times d_{i,j} \qquad 89)$$

Onde:

$\mathcal{D}$ – depósito, local de saída e retorno do veículo

$i\ e\ j$ – dois pontos no mapa, ou seja, possíveis clientes

d – distância

L – ganho

A inclusão do parâmetro de forma, fator α, modifica o objetivo original do método CW que ao invés de minimizar a distância, a prioridade do método CWPF é atuar na forma geométrica que a solução vai assumir, ou seja, na compactação (formas com um perímetro menores) ou descompactação (formas com perímetros maiores) dos roteiros gerados.

De acordo com BITENCOURT (2005) os efeitos causados pela inserção do fator α na fórmula de ganho são:

Para α > 1: Quando utilizados valores de α acima de 1, o resultado do método tende a formar *clusters*, ou seja, favorece a união de pontos que estão muito distantes da origem, quando as distância entres eles são relativamente pequenas em comparação a distância até a origem.

Para α < 1: Para valores de α menores que 1, os resultados possuem favorecem a formação de roteiros ordenados, agregando primeiro os pontos mais próximos da origem e depois agrupando gradativamente os pontos mais distantes da origem.

A Figura 36 mostra as diferenças enquanto a forma causadas pela variação do valor do fator α:

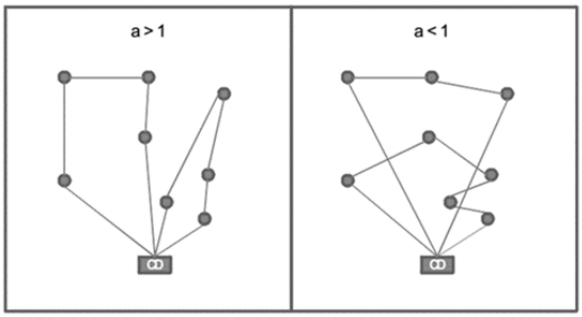


Figura 36 - Exemplo de impacto do índice alfa aplicado na fórmula de Clarke e Wright.

5 Protótipo

5.1 Introdução

Neste capítulo serão descritos os principais aspectos técnicos como; pacotes, classes e diagramas, considerados no desenvolvimento do software, construído em formato de *plugin*, denominado *LogisticTools*, que implementa os procedimentos apresentados no capítulo 4. Para a construção do *plugin* foi utilizado o ambiente de desenvolvimento Eclipse Standard/*Version: Kepler* com a linguagem de Java e o SIG OpenJUMP /*Version: 1.6.3*.

5.2 Adicionando Novas Funcionalidades ao OpenJUMP

SILVA (2008) descreve que é possível adicionar novas funcionalidades ao OpenJUMP através da extensão de 4 itens diferentes:

1. *Plugins* – Itens de menu
2. Ferramentas de cursor – Botões a serem incorporados na barra de ferramentas
3. Renderes – Através de desenhos.
4. DataSources – Carregar e salvar dados de várias formas

Ao inicializar o OpenJUMP, o *Workbench* carrega *extensions*, que adicionam funcionalidades ao Workbench.

Segue abaixo na Figura 37 os componentes da arquitetura OpenJUMP.

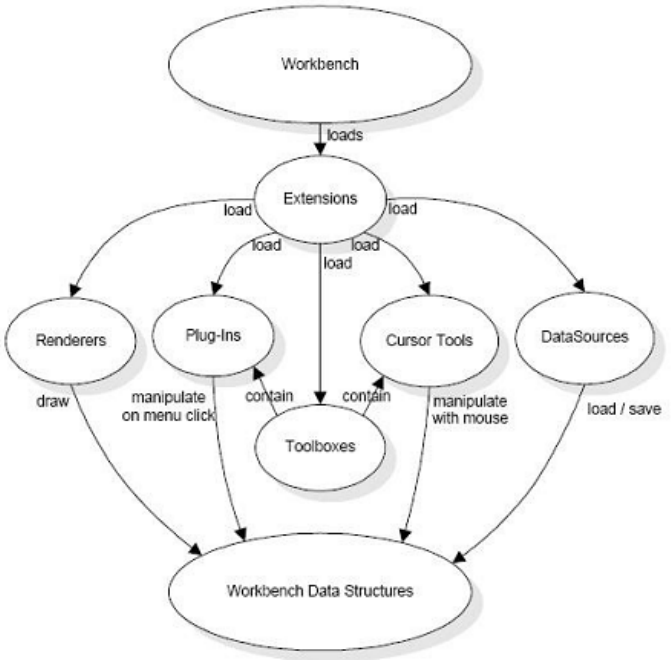


Figura 37 - Extensão de Funcionalidades do OPEMJUMP.

Fonte: SILVA, 2008.

Para maiores detalhes sobre o funcionamento do OpenJUMP, deve-se consultar o Guia do Usuário e do Desenvolvedor, ambos disponíveis em www.openjump.org e também o trabalho de SILVA (2008). Neste trabalho, o protótipo foi implementado através da extensão *Plugin* (vide esquema anterior Figura 37). As funcionalidades podem ser acessadas através do item de menu.

5.3 Levantamento de Requisitos

O mapeamento e a definição de requisitos que o protótipo deveria implementar, foram realizadas em conjunto com o orientador desse trabalho.

SOMMERVILLE (2010) em consonância com PRESSMAN (1987), os requisitos para um sistema são compostos pela descrição do que o sistema deve fazer e suas restrições de funcionamento. Os requisitos podem e devem, ser levantados por diferentes pontos de vista, por exemplo; requisitos de usuário, requisitos de sistema, entre outros.

Para o protótipo em questão, foram levantados os requisitos funcionais e os requisitos não-funcionais.

5.3.1 Requisitos Funcionais

SOMMERVILLE (2010) os requisitos funcionais são as declarações do que o sistema deveria fornecer como ele deve reagir para diferentes *inputs* e como deve ser o seu comportamento em cada situação.

Para o protótipo foram levantados os requisitos funcionais por procedimento conforme a Figura 38 abaixo:

Número Requisito	Procedimento	Tipo de Requisito	Descrição de Requisito
ReqFun1	Definição do Ponto Central	Para Usuário	Usuário poderá inserir os pontos manualmente ou selecionar uma Layer com os pontos via formulário.
ReqFun2		Para Usuário	Usuário poderá escolher qual será o método de definição do ponto central a ser aplicado via formulário. Ele poderá escolher entre os métodos de definição do ponto central: Método de Fibonacci, Método
ReqFun3		Para Usuário	O usuário poderá inserir a precisão almejada para calcular o ponto central pelo método de Weisfield via formulário.
ReqFun4		Para Usuário	O usuário poderá selecionar o tipo de precisão a ser utilizada (por quantidade de iterações ou por precisão) e inserir os parâmetros de acordo com o tipo via formulário.
ReqFun5		Para Sistema	O sistema poderá calcular o ponto central com base nas informações inseridas pelo usuário e pelo método escolhido e retornar as coordenadas dos pontos do ponto central, assim como plotar a
ReqFun6	Clark and Wright	Para Usuário	O usuário poderá selecionar a Layer onde estão os pontos de coleta ou entregas via formulário.
ReqFun7		Para Usuário	O usuário poderá carregar a Layer com as informações sobre os pontos de coleta ou entrega no OPENJUMP.
ReqFun8		Para Usuário	O usuário poderá selecionar o campo da Layer que possui a identificação única de cada ponto via formulário.
ReqFun9		Para Usuário	O usuário poderá carregar a Layer no OPENJUMP que deverá conter OBRIGATORIAMENTE os seguintes campos por ponto de coleta ou entrega: COORDENADAEIXOX, COORDENADAEIXOY, CARGA, TEMPO, TIPOPONTO,
ReqFun10		Para Usuário	O usuário poderá selecionar a referência depósito via formulário.
ReqFun11		Para Usuário	O usuário poderá escolher o cálculo do procedimento pela forma padrão ou com o parâmetro de forma, que exigirá que seja definido um
ReqFun12		Para Usuário	O usuário poderá inserir mais restrições como VELOCIDADE MÉDIA DO VEÍCULO, PESO MÁXIMO A SER TRANSPORTADO POR ROTA, TEMPO MÁXIMO POR ROTA OU VALOR MÁXIMO POR ROTA via formulário.
ReqFun13		Para Sistema	O sistema poderá executar o procedimento escolhido pelo usuário e plotar os resultados graficamente, incluindo a geração de uma camada com os resultados do procedimento.
ReqFun14	Formação de Cluster	Para Usuário	O usuário poderá carregar a Layer com as informações sobre os pontos no OPENJUMP.
ReqFun15		Para Usuário	O usuário poderá selecionar a camada com os pontos via formulário.
ReqFun16		Para Usuário	O usuário poderá selecionar o campo da Layer que possui a identificação única do ponto via formulário.
ReqFun17		Para Usuário	O usuário poderá definir o número de clusters a serem gerados via
ReqFun18		Para Sistema	O sistema poderá gerar os clusters de acordo com parâmetros inseridos pelo usuário com base no método de Kruskal e plotar os resultados
ReqFun19	Formação de Zonas	Para Usuário	O usuário poderá executar 3 procedimentos: Calcular área admissível, Calcular nível de serviço e Formar Zonas.
ReqFun20		Para Usuário	O usuário poderá preencher, para cada procedimento, todos os parâmetros solicitados via formulário.
ReqFun21		Para Usuário	O sistema deverá imprimir os resultados via formulário para os procedimentos: Calcular área admissível e Calcular nível de serviço .
ReqFun22		Para Sistema	O sistema deverá plotar graficamente as diferentes SubZonas geradas, com cores diferentes e com uma layer com os dados de processamento

Figura 38 - Lista de Requisitos Funcionais do Protótipo.

5.3.2 Requisitos Não-Funcionais

SOMMERVILLE (2010) os requisitos não-funcionais, são os requisitos que não estão diretamente conectados aos serviços do que o sistema oferece, apesar dessa relação indireta com os serviços, podem ser tão ou mais críticos do que os

requisitos funcionais, como por exemplo: performance, segurança, tempo de resposta, entre outros.

Segue abaixo conforme a Figura 39, a lista de requisitos não-funcionais para o protótipo.

Número Requisito	Descrição de Requisito
ReqNFun1	Portabilidade: O sistema deverá ser executado em qualquer plataforma.
ReqNFun2	O sistema deverá possuir formulários com interfaceis simples e intuitivas.
ReqNFun3	O sistema deverá ser compatível com as versões anteriores do OPENJUMP 1.6.3 inclusive.
ReqNFun4	O sistema deverá ser desenvolvido na linguagem Java.
ReqNFun5	O sistema deverá ser construído com abordagem de orientação a objeto.

Figura 39 - Lista de Requisitos Não-Funcionais do Protótipo.

5.4 Especificação

Com o objetivo de detalhar e melhorar o entendimento do protótipo de modo a facilitar seu projeto e manutenção, adiante serão detalhados os diagramas de casos de uso que faz parte do padrão *Unified Modeling Language* (UML).

Para o desenvolvimento dos diagramas foi usado o software editor do padrão UML de nome *Astah Community*, que pode ser encontrado no site www.astah.net.

5.4.1 Diagramas de Casos de Uso

Introduzida por JACOBSON em 1993, o diagrama de casos de uso atualmente é um recurso fundamental dentro da UML. De uma forma muito simples, o diagrama consegue representar quais atores, que podem ser pessoas ou

sistemas, estão envolvidos em quais interações. O diagrama de casos de uso descreve interações de alto nível. Essas interações são detalhadas posteriormente dentro dos requisitos do sistema (SOMMERVILLE, 2011). Segue abaixo os diagramas de casos de uso para cada funcionalidade do *plugin*.

5.4.1.1 Diagrama de Caso de Uso – Definição do Ponto Central

Pode-se observar nesse caso de uso que para calcular o ponto central, por qualquer método, o usuário tem a opção de informar os dados através da seleção de uma *Layer* (camada) com pontos e pesos ou inserir manualmente os pontos e pesos. Segue abaixo o caso de uso (Figura 40).

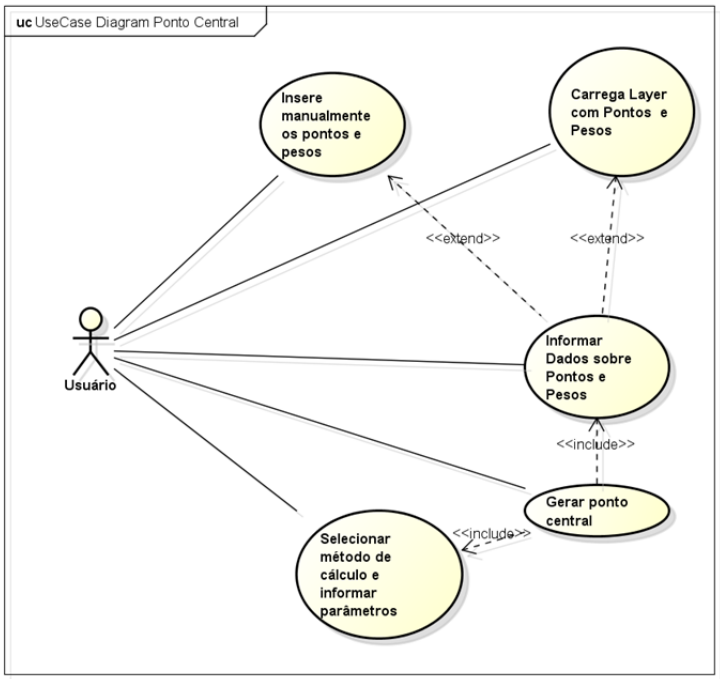


Figura 40 - Diagrama de Caso de Uso para definição do ponto central.

5.4.1.2 Diagrama de Caso de Uso – Formação de Clusters

No caso de uso abaixo, pode-se observar que é obrigatório que o usuário selecione uma *Layer* com os pontos e insira os parâmetros necessários para a geração dos clusters. Segue abaixo o caso de uso (Figura 41).

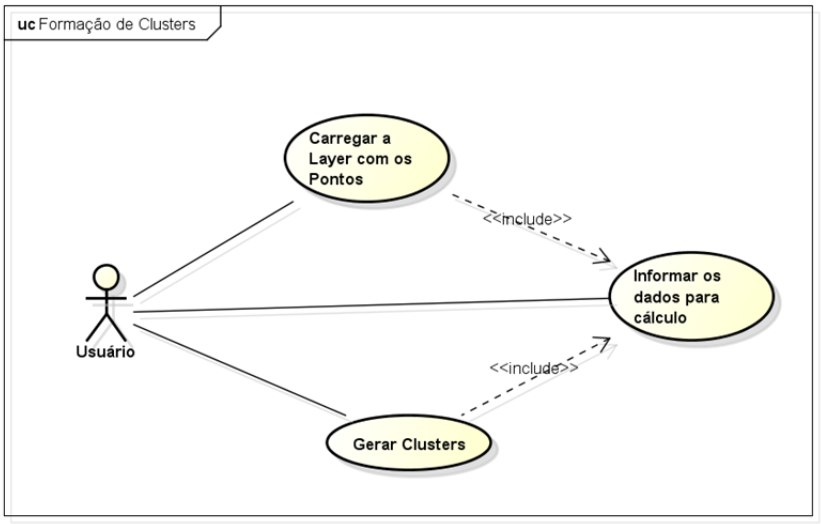


Figura 41 - Diagrama de Caso de Uso para formação de clusters.

5.4.1.3 Diagrama de Caso de Uso – Roteirização com Clarke e Wright

No caso uso de roteirização é obrigatório que o usuário selecione a *Layer* com os pontos de entrega ou coleta que possua os campos de atendimento devidamente preenchidos e posteriormente insira os parâmetros de roteirização, como por exemplo, as restrições de peso, tempo e valor de carga. Segue abaixo o caso de uso (Figura 42).

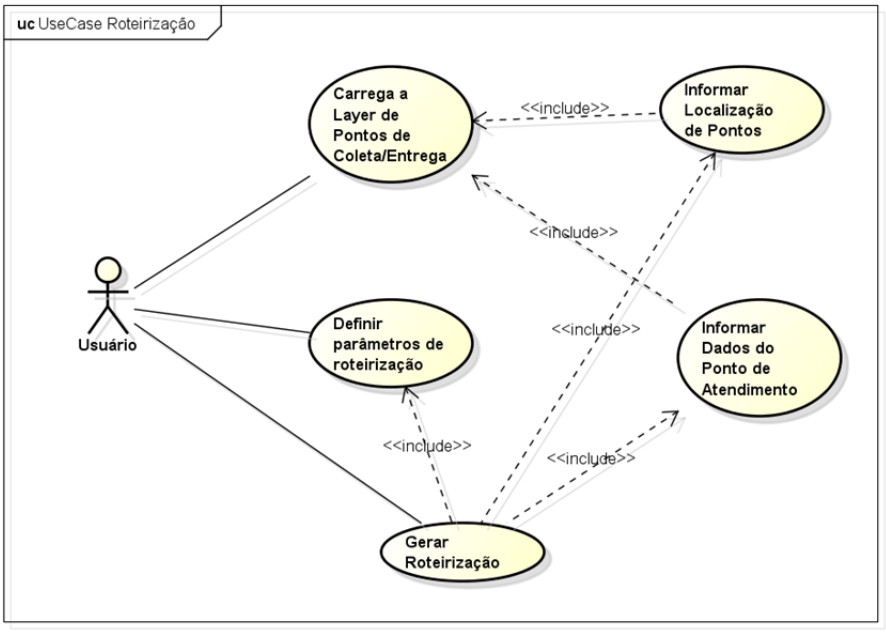


Figura 42 - Diagrama de Caso de Uso de Roteirização.

5.4.1.4 Diagrama de Caso de Uso – Dimensionamento e Formação de Zonas

No caso de uso abaixo, o usuário pode executar três tipos de funcionalidade; calcular nível de serviço, calcular área admissível e gerar zonas (formação de subzonas). Para a execução de todas é necessário informar todos os parâmetros de cálculos solicitados via formulário, sendo que para a funcionalidade "gerar zonas" ainda é necessário que o usuário selecione uma *Layer* de distritos. Segue abaixo o caso de uso (Figura 43).

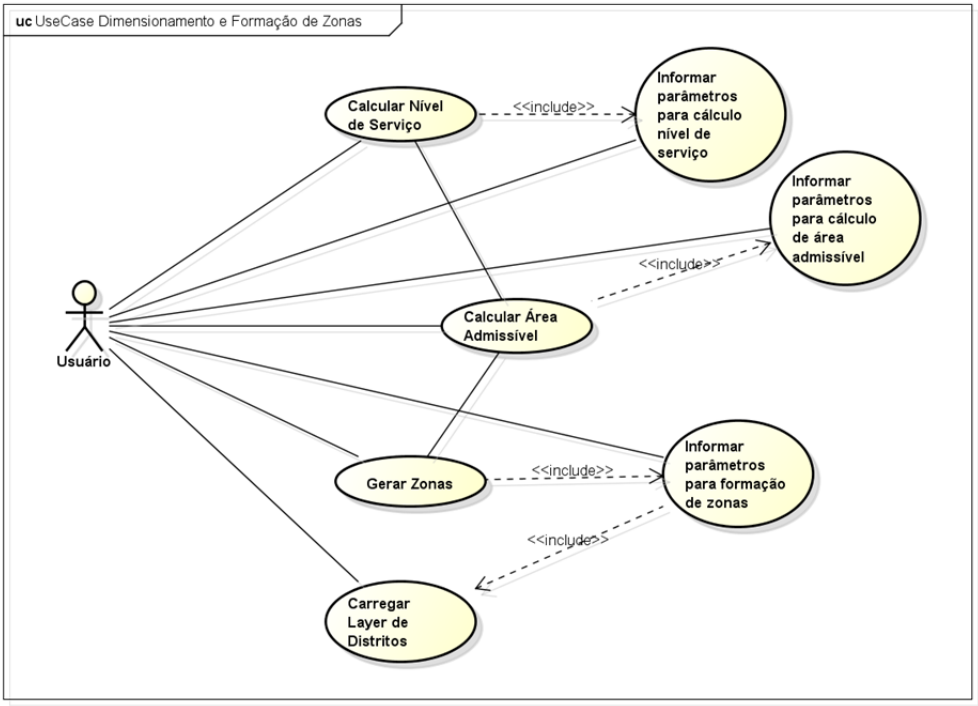


Figura 43 - Diagrama de Caso de Uso de dimensionamento e formação de Zonas.

5.5 Implementação

Nesta seção será descrito os aspectos referentes ao desenvolvimento do código do protótipo destacando sua organização, telas, manual de uso, classes e os principais métodos usados.

5.5.1 Tela e manual do *LogisticTools*

Após carregamento do sistema OpenJUMP, o menu do *plugin* é incorporado a barra de itens de menu padrão do sistema e através desse menu é possível acessar todas as suas funcionalidades (Figura 44).

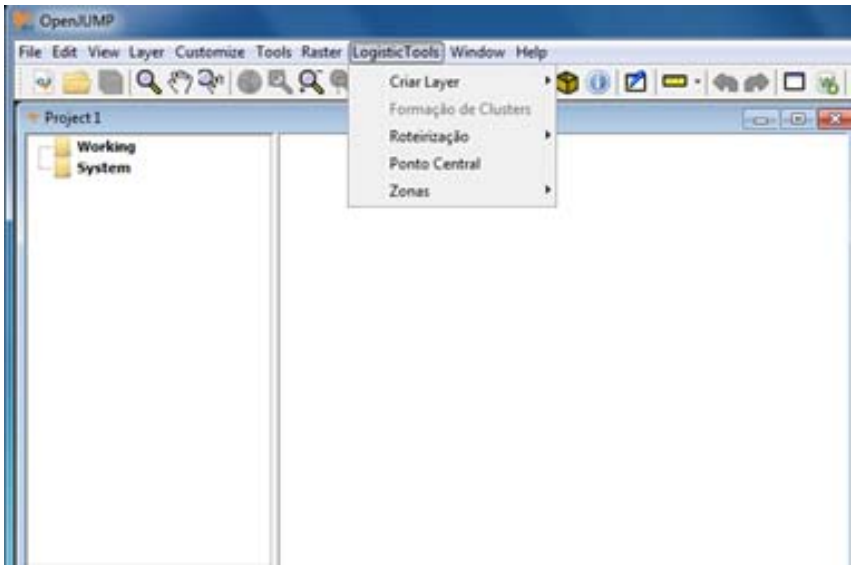
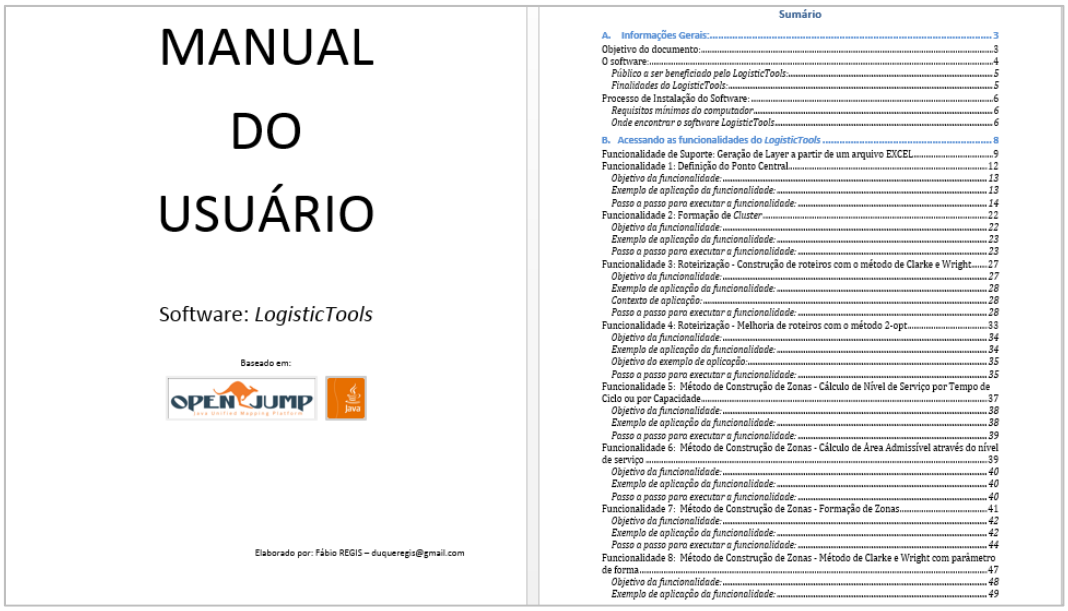


Figura 44 - Menu do protótipo *LogisticTools*

Para facilitar o uso do *plugin* foi desenvolvido um "manual do usuário" (Figura 45). O manual aborda questões como; instalação do *plugin*, como executar cada funcionalidade, exemplo de aplicação para cada funcionalidade, entre outros temas.


MANUAL
DO
USUÁRIO

Software: *LogisticTools*

Baseado em:

Elaborado por: Fábio REGIS – duqueregis@gmail.com

Sumário

Figura 45 - Capa do documento Manual do Usuário

5.5.2 Organização do código

5.5.2.1 Diagrama de classes

Segue abaixo (Figura 46) o diagrama simplificado de classes.

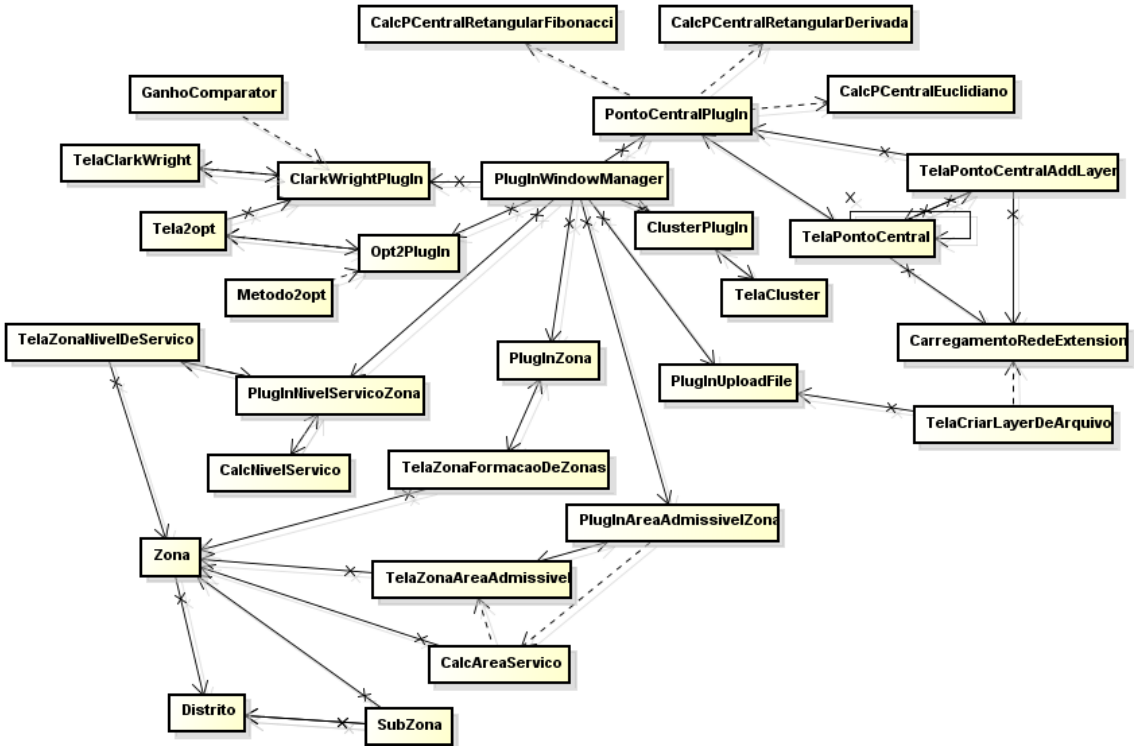


Figura 46 - Diagrama de classes *LogisticTools*

5.5.2.2 Pacotes, classes e métodos

Segue abaixo na Figura 47 a organização de pacotes implementada no *plugin*.

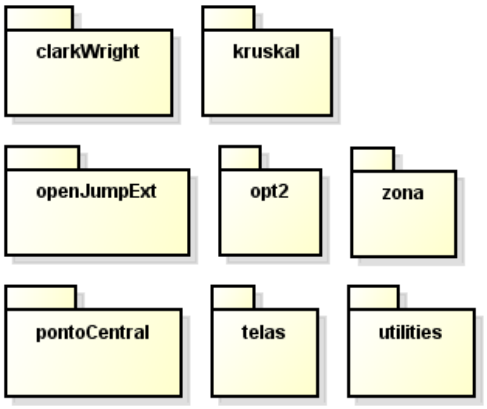


Figura 47 - Organização dos pacotes de código do protótipo

5.5.2.2.1 Pacote pontoCentral

Classes que fazem parte do pacote (Figura 48):

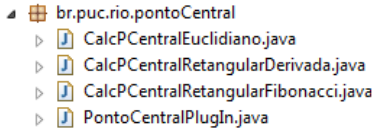


Figura 48 - Classes do pacote analiseEspacial.

Classe CalcPCentralEuclidiano

A classe CalcPCentralEuclidiano (Figura 49) é especializada para realizar o cálculo do ponto central através do método iterativo proposto por Weisfield, abordado anteriormente. O método *calcule* recebe a lista de pontos inseridos pelo usuário, a precisão requerida e executa o método de Weisfield para encontrar o ponto central. Ao encontrar o ponto central, o método também atribui ao valor das variáveis *xfinal* e *yfinal* os as posições x e y do ponto central.. O método *isPrecisaoFuncaoOk* verifica se a diferença entre os valores de distância total da solução atual e a iteração anterior, são maiores ou menores do que a precisão requerida. Se maior retorna falso, se igual ou menor retorna verdadeiro. O método *calculeDistanciaEuclidiana* recebe os eixos de dois pontos quaisquer e retorna a distância euclidiana entre eles.

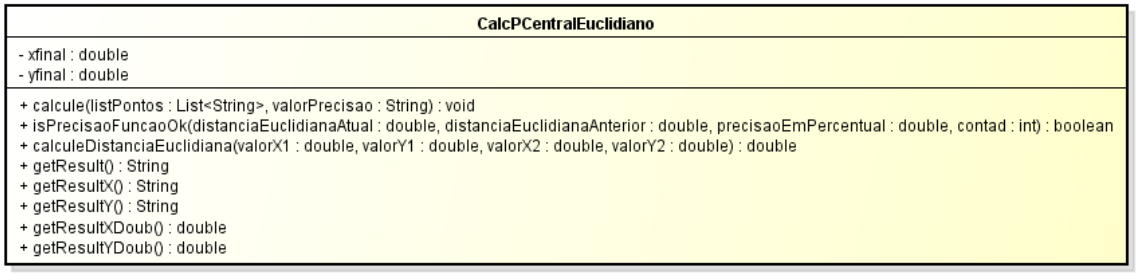


Figura 49 - Classe CalcPCentralEuclidiano

Classe CalcPCentralRetangularDerivada

A classe CalcPCentralRetangularDerivada (Figura 50) é especializada para realizar o cálculo do ponto central pelo método da derivada. O método *calcule* executa o método da derivada e atribui os resultados as variáveis *xfinal* e *yfinal*.

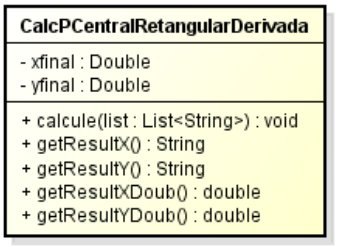


Figura 50 - Classe CalcPCentralRetangularDerivada

Classe CalcPCentralRetangularFibonacci

A classe CalcPCentralRetangularFibonacci (Figura 51) é especializada para realizar o cálculo do ponto central pelo método iterativo proposto por Fibonacci. O método *calcule* executa o método de Fibonacci de acordo com o tipo de critério de parada definido que é atribuído na variável *tipoCalculo*. Há dois tipos; por quantidade de iterações e por precisão. A variável valorDoTipo define qual é a quantidade de iterações caso seja o tipo escolhido ou qual é a precisão requerida no caso da escolha do tipo por precisão. O método também atribui os eixos do ponto central calculado as variáveis *xfinal* e *yfinal*.

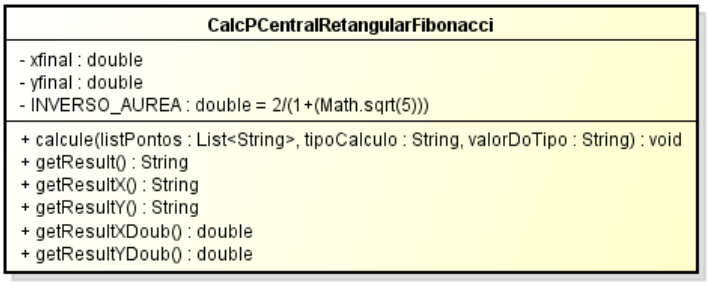


Figura 51 - Classe CalcPCentralRetangularFibonacci

Classe PontoCentralPlugIn

A classe PontoCentralPlugIn (Figura 52) é uma classe que instancia a tela formulário. O método *execute* é responsável por criar e carregar a tela do procedimento para a entrada de dados. A classe estende a classe AbstractPlugIn, que pertence ao padrão do OpenJUMP com o objetivo de criar *plugins* .

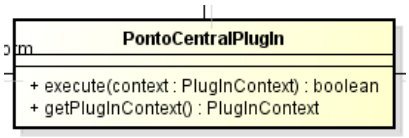


Figura 52 - Classe PontoCentralPlugIn

5.5.2.2.2 Pacote kruskal

Classes que fazem parte do pacote (Figura 53):

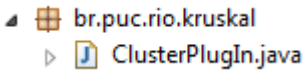


Figura 53 - Classe do pacote kruskal

Classe ClusterPlugIn

A classe ClusterPlugIn (Figura 54) é única do pacote kruskal. Ela implementa todos o procedimento kruskal adaptado para a formação de clusters. O método *MultiEnableCheck* é nativo da classe AbstractPlugin e define que para

carregar esse *plugin* é necessário ter ao menos um *Layer* carregada no OpenJUMP. Esse método é implementado em quase todas as classes de *plugin*.

O método *createKruskal* é o principal método da classe. Ele recebe a lista de pontos, lista do tipo matriz ordenada por distância e a quantidade de cluster requerida pelo usuário. O método executa o procedimento de kruskal, criando uma árvore geradora mínima e depois retira a quantidade de arcos com os maiores pesos até atingir a quantidade de cluster requerida.

O método *geraLayerArcosKruskal* cria a *Layer* e plota os arcos dos clusters gerados com os seguintes atributos: comprimento do arco, nome do cluster, nome do arco, coordenadas *x* e *y*. O método *geraLayerPontoskruskal* é similar ao anterior. Porém ele cria a *Layer* e plota os pontos que fazem parte dos clusters gerados com os seguintes atributos: nome do cluster e nome do ponto.

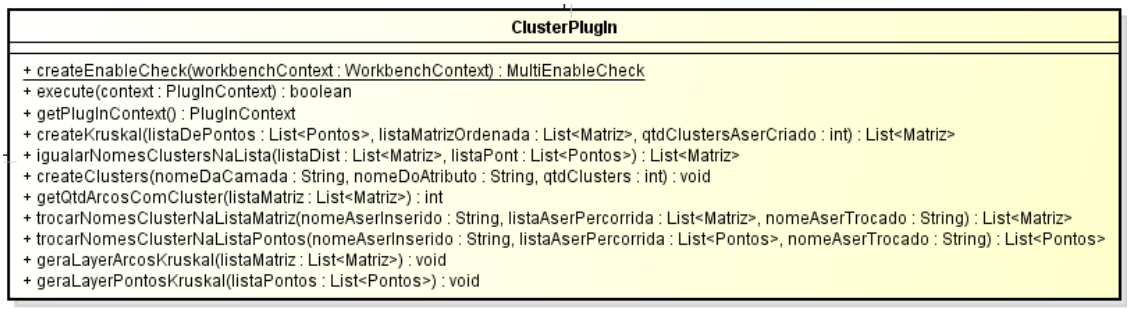


Figura 54 - Classe ClusterPlugIn.

5.5.2.2.3 Pacote clarkWright

Classes que fazem parte do pacote (Figura 55).

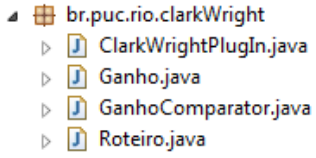


Figura 55 - Classes do pacote clarkWright

Classe ClarkWrightPlugIn

A classe ClarkWrightPlugIn (Figura 56) implementa o método de Clarke e Wright usado para resolver problemas de roteirização e também em formação de zonas.

O *calcClarkWright* é o método principal da classe, pois ele a lista de todos os possíveis ganhos já ordenada e todos os parâmetros de roteirização definidos pelo usuário. Ele tem como saída uma lista com todos os roteiros formados pelo método.

Os métodos que começam com *geraLayer* geram diferentes visões dos resultados obtidos com o método de Clarke e Wright em *Layers* separadas, com diferentes informações sobre os roteiros criados.

O método *createListaGanhoOrdenada* recebe a lista de pontos e o ponto-depósito e cria uma lista ordenada de ganhos de acordo com a fórmula de ganho proposto por Clarke e Wright. Caso o usuário selecione o método que adiciona o parâmetro de forma à fórmula, ele também é considerado para elaborar a lista de ganhos.

O método *calcGanho* realiza o cálculo de ganho para dois pontos quaisquer.

Os métodos *isRestricaoPesoOK*, *isRestricaoValorOK* e *isRestricaoTempoOK* verifica se um determinado roteiro, durante a sua fase de construção, obedece os parâmetros de restrição definidos em relação a limite de carga no roteiro, limite de valor de mercadoria no roteiro e limite de tempo para percorrer o roteiro.

O método *configRoteiroAPartirDeDois* faz a união de dois roteiros de acordo com cada ponto de extremidade definido em cada roteiro que depois será verificado se a junção é viável por outro método.

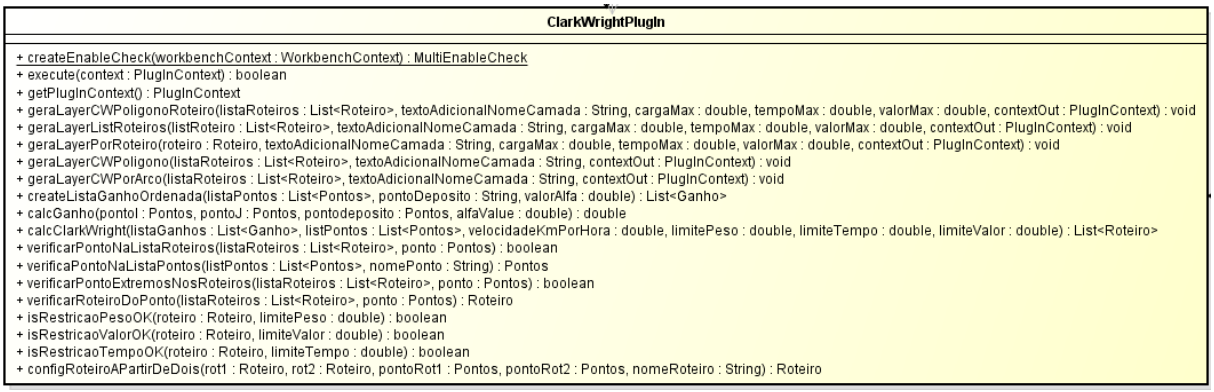


Figura 56 - Classe ClarkWrightPlugIn

Classe Ganho

A classe Ganho (Figura 57) é usada no procedimento de Clarke e Wright e contém as informações sobre cada ganho calculado pelo método como: pontos considerados no ganho, valor do ganho e ponto de depósito.

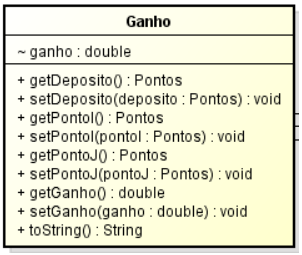


Figura 57 - Classe Ganho

Classe Roteiro

A classe Roteiro (Figura 58) representa um roteiro construído pelo método Clarke e Wright. Ela possui métodos de informações sobre o roteiro como: peso total, tempo total, etc, e métodos relacionados à construção e edição do roteiro.

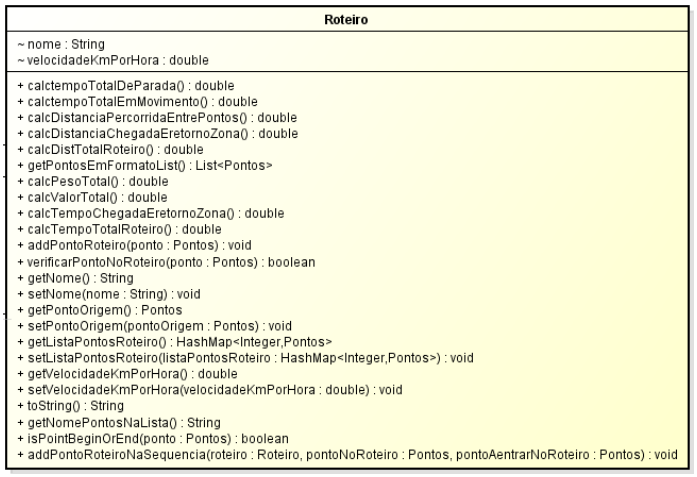


Figura 58 - Classe Roteiro.

5.5.2.2.4 Pacote openJumpExt

Classes que fazem parte do pacote (Figura 59).

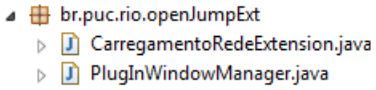


Figura 59 - Classes do Pacote openJumpExt

Classe CarregamentoRedeExtension

A classe CarregamentoRedeExtension (Figura 60) é filha da classe Extension, a qual realiza a interligação entre o *plugin* e o OpenJUMP. O método configure será executado quando o OpenJUMP for aberto. Esse método irá adicionar os itens de menu a barra de menu do OpenJUMP. A partir dessa barra é possível acessar o menu do *plugin* e escolher qual funcionalidade executar.

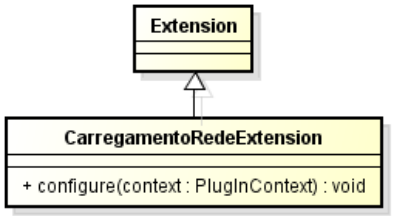


Figura 60 - Classe CarregamentoRedeExtension

Classe PlugInWindowManager

A classe PlugInWindowManager (Figura 61) contém os itens de menu (cada item é uma funcionalidade) que serão carregados quando o OpenJUMP iniciar e irá acionar a execução da funcionalidade que o usuário escolher.

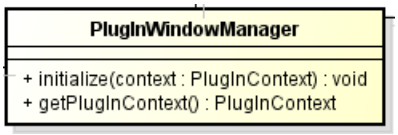


Figura 61 - Classe PlugInWindowManager

5.5.2.2.5 Pacote telas

Classes que fazem parte do pacote (Figura 62).

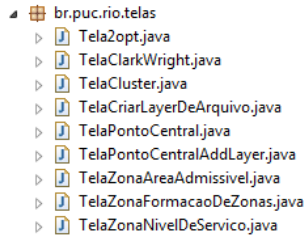


Figura 62 - Classes do Pacote telas.

Todas as classes dentro desse pacote são telas para a entrada e saída de dados. Para cada *plugin* criado existe uma tela respectiva, em formato de formulário, para receber os parâmetros de entrada necessários para a execução daquele procedimento implementado pelo *plugin*.

5.5.2.2.6 Pacote zona

Classes que fazem parte do pacote (Figura 63).

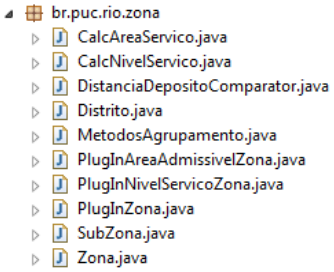


Figura 63 - Classes do pacote zona.

Classe CalcAreaServico

A classe CalcAreaServico (Figura 64) é especializada para calcular a área admissível por tempo ou por capacidade física (carga) conforme apresentado no procedimento de formação de zonas por nível de serviço. O método construtor da classe *CalcAreaServico* no momento da criação do objeto, ele recebe todos os parâmetros e já calcula a área, bastando apenas resgatar os resultados através dos *getters*.

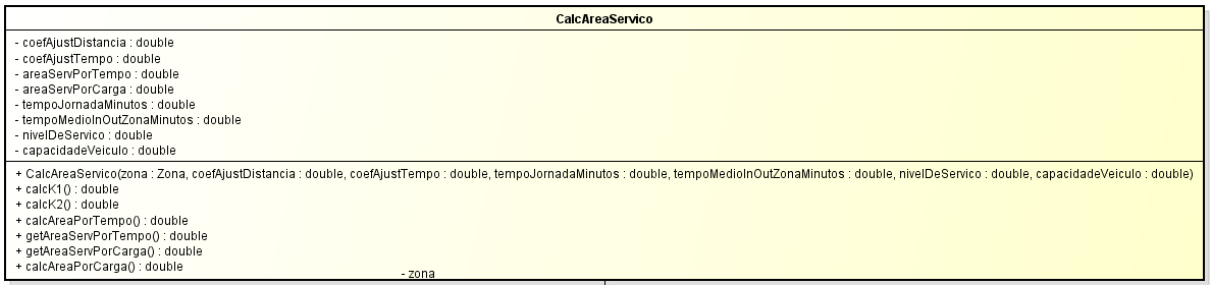


Figura 64 - Classe CalcAreaServico.

Classe CalcNivelServico

A classe CalcNivelServico (Figura 65) é especializada para calcular o nível de serviço a partir dos parâmetros de zona definidos. Ela calcula o nível de serviço por capacidade ou por tempo, sendo tempo de jornada normal ou de jornada máximo (com horas extras).

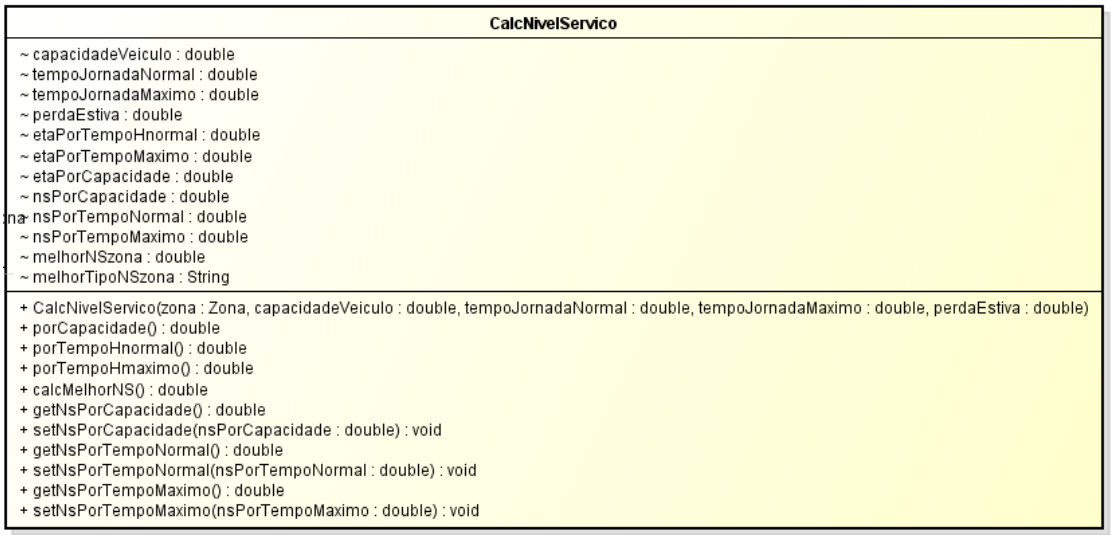


Figura 65 - Classe CalcNivelServico

Classe Distrito

A classe Distrito (Figura 66) representa um distrito que faz parte de uma zona qualquer. Ela tem métodos sobre informações do distrito, como: velocidade média, geometria do distrito, área do distrito, etc.

Figura 66 - Classe Distrito.

Classe MetodosAgrupamento

A classe MetodosAgrupamento (Figura 67) é responsável por implementar os métodos necessários para realizar o procedimento de formação de subzonas.

Há dois métodos de formação de zona implementados: agruparVarreduraSobreEixo e agruparVarreduraSobreEixoSetorizado.

Os métodos *agruparPorTrianguloGiratorio* e *agruparPorTrianguloGiratorio-Rotacionado* executam o procedimento de formação de zonas pelo método da varredura sobre eixo e pelo método varredura sobre eixo setorizado e depois retornam a lista de subzonas criadas.

Os métodos *criarListDeTriangulosIsoscelesSetorizado* e *criarListDeTriangulos-IsoscelesSetorizado* criam todos os triângulos que serão usados nos métodos de formação de zonas e retornam uma lista de polígonos.

O método *gerarSubZonasDentroDoTriangulo* forma subzonas agrupando distritos que possuem interseção com o triângulo projetado.

O método *gerarSubZonasDeDistritosComAreaMaiorIgualQueAreaAdmissivel* é executado antes dos métodos de formação de subzonas. Ele identifica os distritos que possuem tamanho igual ou maior a área admissível definida e cria uma subzona dedicada.

O método *getDistritosTocamNoTriangulo* cria uma lista de distritos que tem interseção com o triângulo.

O método *getDistritoMaisDistante* identifica qual é o distrito mais distante de um determinado distrito através da distância entre os centroides de cada distrito.

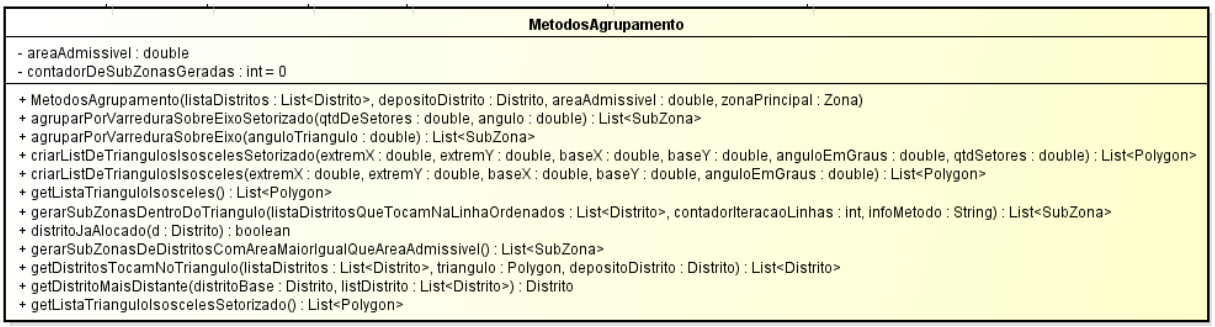


Figura 67 - Classe MetodosAgrupamento.

Classe PlugInZona

A classe PlugInZona (Figura 68) é responsável por gerenciar os métodos que implementam os procedimentos de formação de zonas como mostrado anteriormente.

O método *createListDistrito* cria uma lista de distritos a partir de uma *Layer* selecionada.

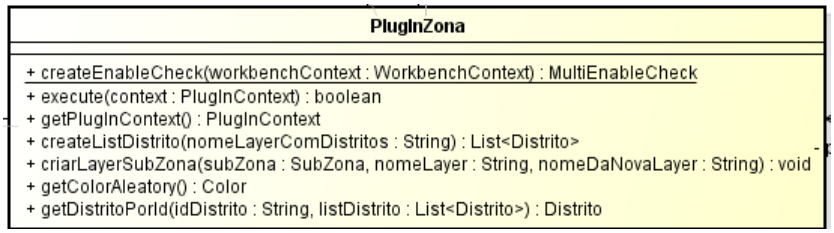


Figura 68 - Classe PlugInZona.

Classe SubZona

A classe SubZona (Figura 69) representa uma subzona que pode ser criada por algum método de formação de subzonas. Cada subzona é formada a partir de uma definição de parâmetros de zona. Ela tem métodos sobre informações da subzona, como: distritos dentro da zona, descrição do método que construiu a zona, etc.

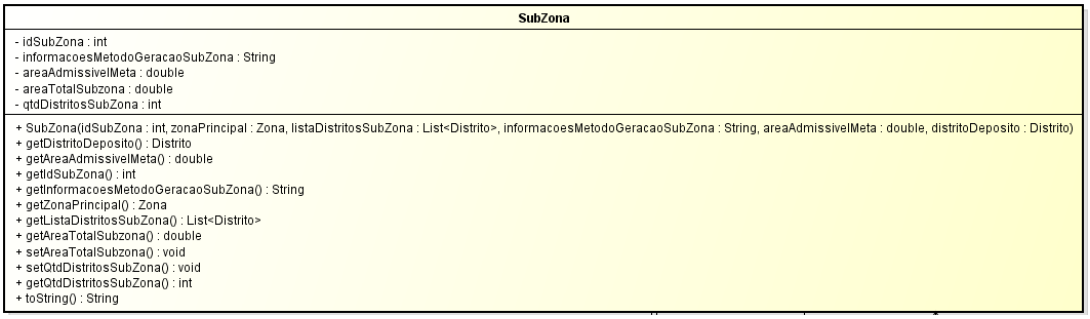


Figura 69 - Classe SubZona.

Classe Zona

A classe Zona (Figura 70) representa uma zona e tem métodos sobre informações como: tempo de ciclo médio na zona, velocidade média na zona, densidade de pontos de atendimento por unidade de área, etc. Ela é usada como objeto base para os cálculos de nível de serviço, área admissível e formação de subzonas.

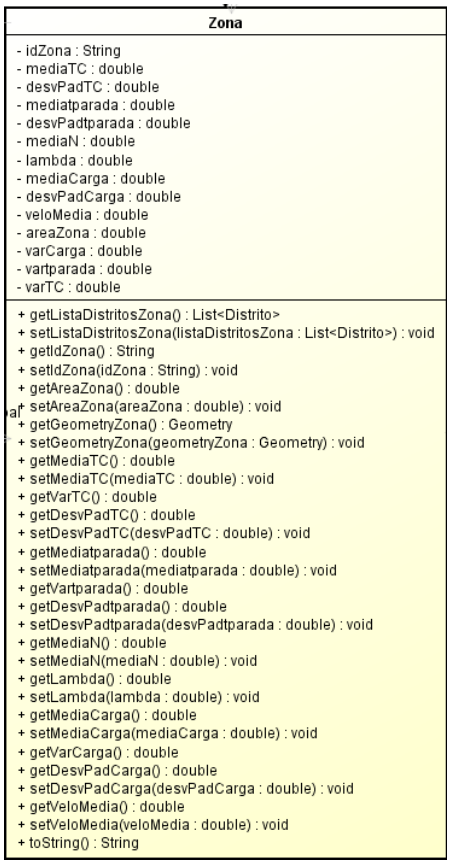


Figura 70 - Classe Zona.

5.5.2.2.7 Pacote utilities

Classes do pacote utilities (Figura **71**):

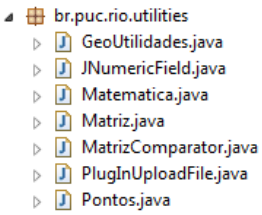


Figura 71 - Classes do pacote utilities.

O pacote *utilities* possuem classes que possui métodos específicos e que são usadas frequentemente por outras classes, justificando a centralização dessas classes num único pacote de modo a promover o reuso e a facilidade de manutenção no código.

Classe Matematica

A classe Matematica (Figura **72**) possui métodos matemáticos padronizados que são usados por métodos de outras classes.

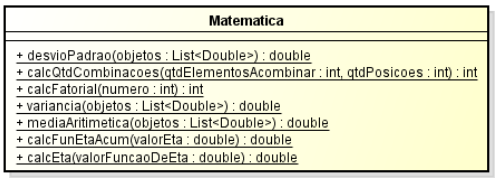


Figura 72 - Classe Matematica.

Classe Pontos

A classe Pontos (Figura 73) Essa classe tem o objetivo de criar um ponto, que pode representar clientes, depósitos, fornecedores, etc. Ela possui atributos que geralmente são comuns aos exemplos citados anteriormente como: tempo de atendimento, quantidade de carga para coleta ou entrega, coordenadas de localização, etc.

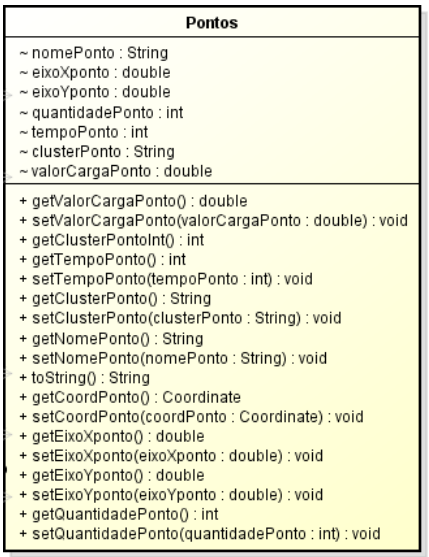


Figura 73 - Classe Pontos.

5.5.2.2.8 Pacote opt2

O possui somente uma classe (Figura 74) que implementa o procedimento usado no protótipo para a melhoria de roteiros, 2-opt. O parâmetro de parada definido para o método é a quantidade máxima de melhorias no roteiro.

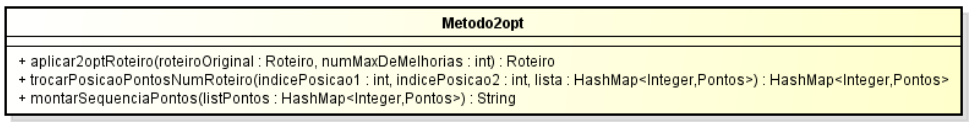


Figura 74 - Classe Metodo2opt

6 Conclusões e Recomendações

Nesta dissertação foram apresentados e desenvolvidos procedimentos para solucionar quatro tipos de problemas comuns na logística, são eles: problemas de localização, problemas de roteirização, problemas de formação de zonas e problemas de identificação de clusters. Todos os procedimentos foram implementados como ferramentas computacionais em forma de *plugins* sobre um software do tipo SIG utilizando a linguagem de programação Java.

Entre os problemas de localização abordados, foram tratados os problemas de definição de ponto central para as métricas retangular e euclidiana. Em relação à métrica retangular foram aplicados os métodos de Fibonacci e da Derivada. Para a métrica euclidiana foi aplicado o método de Weisfield.

Nos problemas de formação de clusters, foi abordado o problema de identificar uma quantidade *N* de clusters. Para o tratamento do problema foi aplicada uma versão modificada do algoritmo de Kruskal.

O problema de roteirização foi dividido em dois subproblemas: construção de roteiros e melhoria de roteiros. Em relação à construção de roteiros foi aplicado o tradicional método de economias de Clarke e Wright. Para a melhoria dos roteiros gerados foi aplicado o método 2-opt.

Por último, foi abordado o problema de formação de zonas. Para o tratamento desse problema foram aplicados dois métodos com abordagens bem diferentes. No primeiro caso foi aplicada uma versão modificada do método de Clarke e Wright com a adição de um parâmetro de forma. O segundo tratamento do problema foi feito a partir de um método de formação de zonas por nível de serviço com duas formas distintas de formar zonas que é semelhante ao método de varredura, porém, com o uso de polígonos.

Todo o trabalho foi baseado fortemente nos problemas abordados durante as aulas no mestrado profissional em logística na PUC-RJ e principalmente nos tópicos lecionados pelo professor José Eugênio Leal.

De certa forma, este trabalho também é uma continuação dos trabalhos de outros dois autores. SILVA (2008) que aplicou procedimentos para o problema de roteirização de veículos no mesmo software SIG aqui usado e BITENCOURT (2005) que implementou diversos procedimentos para solução de problemas logísticos baseado na literatura do Prof. Antônio Galvão Novaes.

Ao final deste trabalho foram apresentados os principais componentes usados no planejamento e no desenvolvimento do software tornando possível conhecer seus principais mecanismos de funcionamento e também para adição de futuras funcionalidades por outras pessoas.

Recomenda-se a contínua melhoria e incremento do *software* elaborado, revisando itens como aparência de formulário, informações geradas pelos procedimentos, apresentações gráficas mais customizadas, entre outros de modo a facilitar e tornar mais intuitivo o contato do estudando em logística com o software. Recomenda-se também a implementação de outros procedimentos para tratar os mesmos problemas ou até problemas diferentes, que apesar de estarem consolidados na literatura, ainda carecem de um apoio ferramental computacional sem restrições de uso, para se tornarem mais conhecidos e consequentemente mais aplicados aos desafios logísticos.

7 Referências Bibliográficas

ARKADER, Rebecca. **As Mudanças No Ambiente Empresarial E O Encino Da Logística No Brasil**. Academia (Consejo Latinoamericano de Escuelas de Administración) Volume 22, 1998.

BALLOU, Ronald H. **Gerenciamento da cadeia de suprimentos/logística empresarial.** Tradução Raul Rubenich. – 5.ed. – Porto Alegre : Bookman. ISBN: 978-85-363-0591-2, 2006.

BITENCOURT, Mario Antonio Pinheiro. **Componentes de um sistema computacional para análise de sistemas logísticos**. – Rio de Janeiro: Dissertação (Mestrado). PUC-Rio, Departamento de Engenharia Industrial, 2005.

BOWERSOX, Donald J. **Logística Empresarial: O Processo De Integração Da Cadeia De Suprimento.** Tradução Equipe do Centro de Estudos em Logística. – São Paulo: Atlas, 2001.

CARDOSO, Igor de Moraes; LIMA, Renato da Silva. **Métodos ativos de aprendizagem: o uso do aprendizado baseado em problemas no ensino de Logística e Transportes**. TRANSPORTES, Rio de Janeiro, RJ, v. 20, n. 3, p. 79-88,. ISSN 2237-1346. Disponível em: <http://www.revistatransportes.org.br/anpet/article/view/561>. Acesso em: 28 Nov. 2013. doi:10.4237/transportes.v20i3.561, 2012.

CHOPRA, Sunil. **Gerenciamento da cadeia de suprimentos**. Tradução Cláudia Freire; revisão técnica Paulo Roberto Leite. – São Paulo: Prentice Hall, 2003.

CLARKE, G. e WRIGHT, W. J. **Scheduling of Vehicle from a Central Depot to a Number of Delivery Points.Operations Research**, Vol. 12, p.568-581, 1964.

COLIN, E. C. **Pesquisa Operacional: 170 aplicações em Estratégia, Finanças, Logística, Produção, Marketing e Vendas**. – Rio de Janeiro : LTC, 2007. ISBN: 978-85-216-1559-0, 2007.

JAIN, AK and Murty, MN and Flynn, PJ .**Data Clustering: A Review**. In: ACM Computing Surveys, 31 (3). pp. 264-323.1999.

LEAL, J.E. **Apostila de Distribuição Física do Mestrado Profissionalizante em Logística** – PUC/RJ, 2012.

L. HUANG et al. **K-means Initial Clustering Center Optimal Algorithm Based on Kruskal**. Journal of Information & Computational Science 9: 9 , 2012.

LIN,S. E KERNIGHAN,B.W. "**An Effective Heuristic Algorithm for the Traveling Salesman Problem".** Operations Research, vol. 21, p.498-516, 1973.

LIU, Fuh-Hwa e SHEN, Sheng-Yuan**, A Method for Vehicle Routing Problem with Multiple Vehicle Types ad Time Windows**, Department of Industrial Engineering and Management National Chiao Tung University, Hsinchu, Taiwan, Proc. Natl. Sci. Counc., vol 23, n. 4, p. 526-536 , 1999.

MACHLINE, Claude. **Cinco décadas de logística empresarial e administração da cadeia de suprimentos no Brasil**. Revista de administração de empresas 51, no. 3, 2011.

MORENO, Eduardo e Ramírez, Héctor. **Fundamentos Y Algoritimo**s. Chile : Editorial ebooks Patagonia - J.C. Sáez Editor. ISBN 9789563060768, 2011.

NOVAES, A. G. N. et al. **Solving continuous location-districting problems with Voronoi diagrams**. Computers and Operations Research, n. 36, p. 40-59, 2009.

NOVAES, Antônio Galvão. **Panorama Profissional Evolução da logística no Brasil.** TRANSPORTES, Rio de Janeiro, RJ, v. 9, n. 1 . ISSN 2237-1346. Disponível em: <http://www.revistatransportes.org.br/anpet/article/view/561>. 2001.

NOVAES, Antonio Galvão. **Logística e Gerenciamento da Cadeia de Distribuição**. Rio de Janeiro : Elsevier. ISBN 978-85-352-2415-3, 2007.

NOVAES, Antonio Galvão - **Sistemas logísticos: transporte, armazenagem e distribuição física de produtos**. São Paulo: Edgard Blücher, 1989.

PRESSMAN, Roger S. **Software engineering : a practitioner's approach** /. 2nd. ed. – New York : McGraw-Hill, Inc. 567p. : ISBN 0071002324, 1987.RAINER, R. Kelly. **Introdução a sistemas de informação.** Tradução Multinet Produtos. – 3.ed. – Rio de Janeiro : Elsevier, 2011.

ROSE, Adriana. **Uma avaliação comparativa de alguns sistemas de informação geográfica aplicados aos transportes**.. Dissertação (Mestrado em Transportes) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2001. Disponível em: http://www.teses.usp.br/teses/disponiveis/18/18137/tde-17032002-102609/, 2001.

SILVA, Liliane Sena da,. **Nível de Serviço Logístico: Estudo de Caso em uma Empresa de Bebidas da Paraíba**. Enegep. Disponível em: www.producao.ufrgs.br/.../495_p20090334_enegep_nivel_de_servico_logistico.pdf. 2008.

SILVA Júnior, Orivalde Soares. **Roteirização de veículos de carga com múltiplos depósitos em sistema de informação geográfica livre**. –Rio de Janeiro: Dissertação (Mestrado). Instituto Militar de Engenharia - IME, 2008.

SILVA Júnior, Orivalde Soares. **Algoritmos para os Problemas de Roteirização Estáticae Dinâmica de Veículos com Janelas de Tempo**. Rio de Janeiro: Tese (Doutorado). PUC-Rio, Departamento de Engenharia Industrial, 2013.

SOMMERVILLE, Ian**. Software engineering**. 9. ed. Boston: Addison-Wesley. xv, 773 p. ISBN 9780137035151, 2010.

TURBAN, Efraim. **Tecnologia da Informação para Gestão.** Edição 3 – Porto Alegre : Bookman. ISBN 978-85-363-0341-3, 2004.

WU, Bang Ye and CHAO, Kun-Mao. **Spanning Trees and Optimization Problems**. Boca Raton,Florida: Chapman& Hall/CRC, 2005. ISBN: 1-58488-436-3. 2005.

KOBAYASHI, Shun`ichi. **Renovação da Logística: como definir estratégias de distribuição física global.** Tradução Valéria Custódio dos Santos. – São Paulo: Atlas. ISBN: 85-224-2557-4. 2000.

YELLOW, P.C. **A Computational Modification to the Savings Method of Vehicle Scheduling.** Operational Research Quarterly (1970-1977) Vol. 21, No. 2, pp. 281-283. Published by: Palgrave Macmillan Journals on behalf of the Operational Research Society. 1970.