

Geo-Rota - Sistema Web de Suporte a Decisão para o Problema de Roteirização de Veículos com Multi-Compartimentos

Geo-Rota - Web Decision Support System for the Multi-Compartment Vehicle Routing Problem

Carlos Leonardo Ramos Póvoa¹, Universidade Estadual do Norte Fluminense - Darcy Ribeiro

Frederico Galaxe Paes², Instituto Federal Fluminense - Campus Guarus

André Soares Velasco³, Instituto Federal Fluminense - Campus Guarus

RESUMO

Este trabalho objetiva descrever o desenvolvimento do sistema de roteirização de veículos Geo-Rota web. O referencial teórico aborda a integração de algoritmos de roteirização com dados geográficos, bem como as estratégias de implementação utilizadas. Foram feitos testes com instâncias encontradas na literatura para validar o algoritmo GRASP desenvolvido para resolver o problema de roteirização de veículos com multi-compartimentos. Para demonstrar a eficiência do sistema foi realizado um estudo de caso, onde se obteve melhora nas rotas geradas pela empresa, apresentando redução de veículos e de distância. A conclusão que se chegou foi que o sistema desenvolvido otimiza de forma eficiente os roteiros de entrega, contribuindo para aumentar a eficiência logística das empresas.

Palavras-chave: Logística. Roteirização de veículos. Suporte a decisão.

Editor Responsável: Prof.
Dr. Hermes Moretti Ribeiro da
Silva

ABSTRACT

This work aims to describe the development of the Geo-Rota webvehicle routing system. The theoretical framework addresses the integration of routing algorithms with geographic data, as well as the implementation strategies used. We have performed tests with examples s found in the literature to validate the GRASP algorithm developed to solve the vehicle routing with multi-compartments problem. In order to demonstrate the effectiveness of the system, a case study was carried out that resulted in improvements in the routes generated by the company, showing a reduction of vehicles and distance. The conclusion reached was that the developed system optimizes the delivery schedules efficiently, contributing to an increase in the logistical efficiency of the companies.

Keywords: Logistics. Vehicle routing problem. Decison support system.

1. Av. Alberto Lamego, 2000 - Parque California, Campos dos Goytacazes - RJ, 28013-602, clrp@uenf.br; 2. fpaes@iff.edu.br; 3. asvelasco@iff.edu.br

PÓVOA, C.L.R.; PAES, F.G.; VELASCO, A.S. Geo-Rota - Sistema Web de Suporte a Decisão para o Problema de Roteirização de Veículos com Multi-Compartimentos. **GEPROS. Gestão da Produção, Operações e Sistemas**, v. 15, n. 3, p. 133 - 154, 2020.

DOI: 10.15675/gepros.v15i3.2553

1. INTRODUÇÃO

A atividade de distribuição física de produtos de uma empresa compreende toda a movimentação de bens entre as fábricas, centros de distribuição e consumidores. A última etapa nessa movimentação, dos centros de distribuição para os consumidores, a qual pode ser definida como transporte local ou entrega, representa o elo mais caro da cadeia de distribuição. As atividades relacionadas ao transporte e a distribuição física buscam, cada vez mais, o aprimoramento da qualidade e da produtividade, de forma a garantir um melhor aproveitamento da frota e a redução do custo de transporte. O aumento do número de entregas e sua dispersão geográfica em decorrência da política de redução de estoque das empresas, que as leva a efetuar pedidos menores e com maior frequência, causam um impacto significativo nas operações e nos custos associados aos sistemas de distribuição. Concomitantemente, aumentam as exigências dos clientes em relação a uma entrega mais eficiente.

Uma das atividades mais importantes no transporte de mercadorias consiste na montagem dos roteiros de entrega dos veículos, ou seja, quais clientes cada veículo deve atender e qual a rota que cada um deve fazer de forma a percorrer todos os pontos de entrega com um custo mínimo de operação. Numa malha viária, ou rede de vias, pode-se obter o percurso mínimo para os veículos considerando: os custos de operação mínimos, os tempos de percurso mínimos, as distâncias mínimas ou o menor consumo de combustível.

Esse problema é conhecido como problema de roteirização de veículos, que segundo Laporte *et al.* (2000) consiste em definir roteiros que minimizem o custo total de atendimento, cada um dos quais iniciando e terminando no depósito ou base de veículos, assegurando que cada ponto seja visitado exatamente uma vez e a demanda em qualquer rota não exceda as restrições do veículo que a atenda.

Quando o número de entregas e a frota disponível crescem, montar os roteiros dos veículos sem o auxílio de um sistema de suporte a decisão, conhecidos como roteirizadores, é uma tarefa árdua que certamente levará ao aumento dos custos de transporte. Os roteirizadores comerciais existentes são restritos ao uso em grandes empresas, devido ao alto custo de aquisição e pela alta complexidade de utilização pelos operadores, tornando inviável a utilização por pequenas e médias empresas.

Dados cartográficos são primordiais para a implementação de softwares de roteirização de veículos (SANTOS *et al.*, 2008). Normalmente uma base de arruamentos (*centerlines*) com dados de nome da rua, numeração, sentido da via, velocidade média e tempo de deslocamento são essenciais para aplicações de transporte. Destaca-se, também, a importância de localizar os pontos de entrega (clientes) nessa base utilizando recursos de *Geocode*, algoritmos que transformam um endereço do tipo rua x, nº y em um par de coordenadas geográficas (longitude e latitude).

Recentemente, serviços de mapas *online* vêm sendo utilizados para desenvolver sistemas de roteirização devido à facilidade de compartilhar os dados geográficos pela internet e por disponibilizarem APIs para usuários desenvolverem suas próprias aplicações (SANTOS *et al.*, 2011).

A necessidade de desenvolver um sistema de roteirização de fácil uso, com acesso aos mais recentes mapas viários e que possa otimizar de forma eficiente os roteiros de entrega em empresas, que não tem condições de adquirir sistemas disponíveis no mercado, foram os motivadores para o desenvolvimento do presente estudo que tem como objetivo apresentar o desenvolvimento do sistema Geo-Rota web de apoio a decisão para o problema de roteirização de veículos.

O uso de sistemas de roteirização garante uma melhora significativa nas operações de transporte das empresas (FAULIN *et al.*, 2005). Sua utilização resulta em otimização de tempo, economia de combustível e satisfação do cliente, elemento essencial para que essas possam atuar de forma competitiva no mercado contemporâneo.

O presente artigo é estruturado em cinco seções, além da introdução. A segunda seção é dedicada a uma revisão teórica sobre as estratégias de integração de algoritmos de roteirização com dados cartográficos; a terceira descreve a arquitetura desenvolvida para o sistema; a quarta seção demonstra o algoritmo GRASP desenvolvido para resolver o problema de roteirização de veículos com multi-compartimentos (PRV-MC), bem como os experimentos computacionais com instâncias da literatura; na quinta seção é realizado um estudo de caso para comprovar a eficiência do sistema em uma empresa, por fim, na sexta seção são feitas as conclusões.

2. REFERENCIAL TEÓRICO

A solução de problemas de roteirização de veículos e suas variantes vêm sendo amplamente estudados na literatura, principalmente as soluções utilizando algoritmos heurísticos. Entretanto, apenas alguns estudos utilizam esses algoritmos aliados a sistemas de apoio a decisão baseados em dados geográficos (KEENAN, 1998). Os trabalhos de Angerhrn e Lüthi (1991) e Lapalme e Rosseau (1992) foram os pioneiros em utilizar mapas e algoritmos em rede (caminho mínimo) fora dos softwares tradicionais de sistemas de informação geográfica (GIS). Os sistemas desenvolvidos davam autonomia para o usuário montar e editar a rede, utilizando mapas como plano de fundo. O trabalho de Wunderlich *et al.* (1992) apresenta um sistema de apoio a decisão para a roteirização de leitores de gás para uma empresa localizada na Califórnia.

Basnet *et al.* (1996) apresenta uma aplicação para resolver o problema de coleta de leite de fazendeiros na Nova Zelândia. Esse problema de roteirização é multi-depósito e foi resolvido utilizando o algoritmo da varredura (*sweep*). O trabalho de Weigel e Cao (1999) mostra a integração de um algoritmo de roteirização com um sistema de informação geográfica comercial.

O sistema desenvolvido por Nussbaum *et al.* (1997) foi implementado para resolver o problema de distribuição de combustível de uma empresa Chilena. As rotas são limitadas pela capacidade do tanque, duração da viagem e janela de tempo. Em vez de utilizar técnicas de otimização, o mesmo foi resolvido através do histórico de conhecimento das rotas geradas (*historical expert knowledge*).

Faulin *et al.* (2005) desenvolveram o software *LogDis*, um sistema de apoio a decisão projetado para resolver o problema de roteirização de veículos numa empresa de produtos congelados na Espanha. Gayialis e Tatsiopoulos (2004) usaram apenas softwares comerciais para desenvolver o sistema, ao invés de desenvolver módulos específicos de roteirização. A mesma estratégia foi utilizada no trabalho de Mendoza *et al.* (2009) que desenvolveram um sistema para roteirizar a medição de hidrômetros em uma companhia de água e esgoto da Colômbia. Tarantilis e Kiranoudis (2002) usaram uma estratégia similar com o sistema de informação geográfica *Arcview* da ESRI®.

Santos *et al.* (2008) desenvolveram um sistema de apoio a decisão para otimizar a coleta de lixo (C-CARP) utilizando a linguagem orientada a objetos *Avenue*, incluída no GIS

Arcview. Já em Santos *et al.* (2011) os autores desenvolveram o mesmo sistema de apoio a decisão para a plataforma web utilizando a base de dados do *Google Maps*.

O sistema de apoio a decisão desenvolvido por Pamučar *et al.* (2016), modifica o algoritmo de Dijkstra para gerar rotas com menor impacto ambiental (*green transport*) num sistema de informação geográfica (GIS), que contém toda a informação cartográfica necessária.

O trabalho de Kramer *et al.* (2016) utiliza a API do *Google Maps* para calcular a matriz de distância para a solução do problema assimétrico de roteirização de veículos com frota heterogênea e limitada. Um estudo de caso foi feito numa indústria de bebidas localizada no Brasil. Já o trabalho de Li *et al.* (2016) utiliza a mesma API para desenvolver um sistema de roteirização dinâmica de veículos.

Amal *et al.* (2018) implementaram um algoritmo genético para resolver o problema de coleta de resíduos sólidos urbano e integraram o algoritmo com o sistema de informação geográfica *ArcGis*.

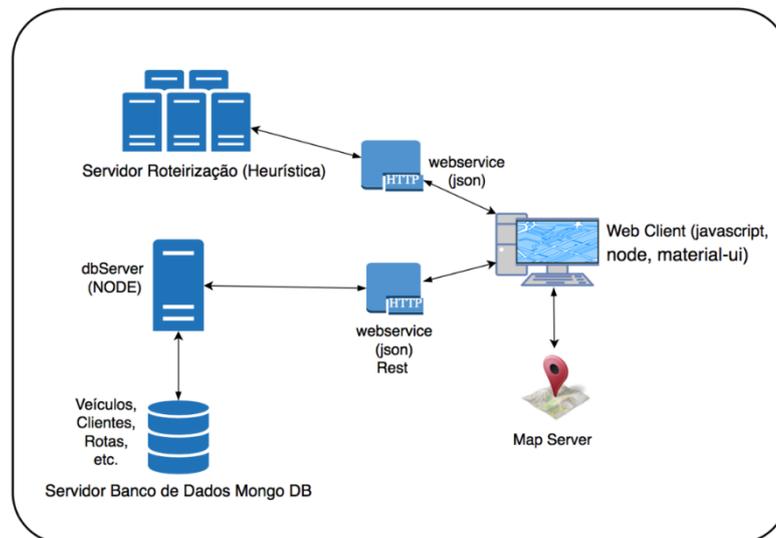
3. PROCEDIMENTOS METODOLÓGICOS

3.1 Arquitetura do sistema

O sistema de apoio a decisão desenvolvido, chamado de Geo-Rota web, utiliza uma arquitetura distribuída conforme a Figura 1. A carga de processamento do mesmo foi dividida em três servidores (roteirização, banco de dados e *dbServer*). O servidor de roteirização é responsável pela solução do problema de roteirização de veículos e o *webclient* é a interface onde o usuário interage com o sistema. O sistema pode ser acessado através do seguinte endereço: <http://www.loggeo.net/luca> (login:clrp / senha:123456).

O servidor de banco de dados é responsável pelo armazenamento da base de dados utilizada pelo sistema e o mesmo utiliza um gerenciador orientado a objetos (*MongoDB*[®]). Em um banco de dados orientado a objetos as tabelas são denominadas de coleções e sua estrutura de armazenamento pode ser modificada a qualquer instante, sem perdas de desempenho e de compatibilidade.

Figura 1 - Arquitetura Geo-Rota web



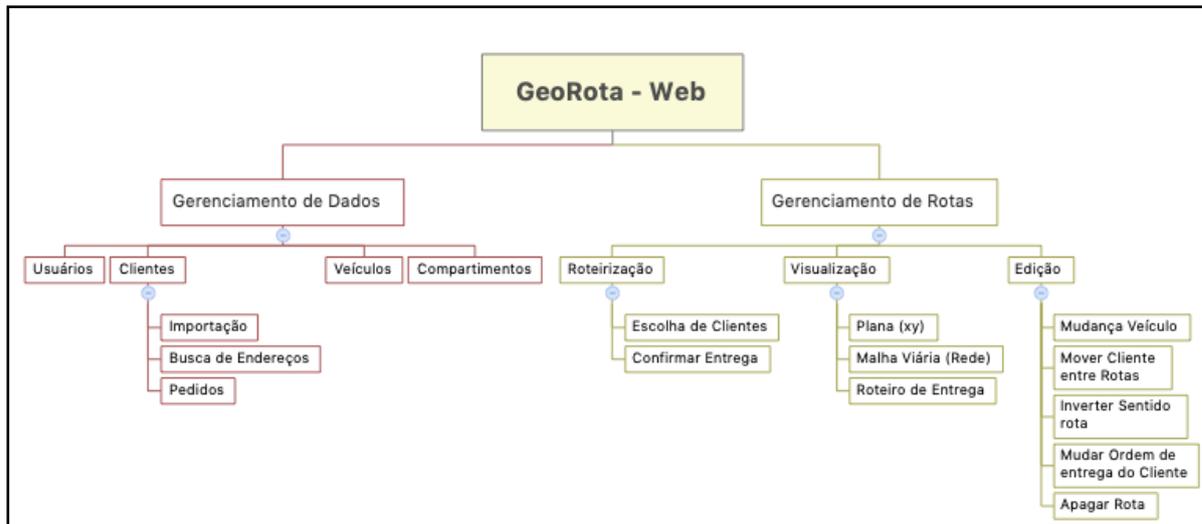
Fonte: Elaborado pelos autores (2018).

O servidor *dbserver* implementa o serviço de arquitetura REST para acesso ao servidor de banco de dados. A arquitetura REST (*Representational State Transfer*) foi definida no trabalho de Fielding (2000) e basicamente utiliza o protocolo HTTP (*get, put, delete e post*) para implementar um *webservice* para troca de dados.

O servidor de roteirização é responsável por resolver o problema de roteirização de veículos com multi-compartimentos, com frota heterogênea e limitada. No problema de roteirização de veículos com multi-compartimentos os veículos tem seu conjunto de divisórias e as demandas dos clientes devem ser alocadas no seu compartimento específico. Para exemplificar, considere uma empresa que vende produtos congelados e secos e que possui uma frota de veículos dividida de acordo com a temperatura do produto. Ou uma transportadora de combustíveis que possui uma frota de veículos com divisórias para cada tipo de combustível. O problema foi resolvido através de uma heurística GRASP. Detalhes do algoritmo desenvolvido são demonstrados na Seção 4.

A interface final (*webclient*) foi construída utilizando a biblioteca *React (javascript)* e a Figura 2 mostra as funções desenvolvidas. A base de dados cartográfica utilizada foi a do *Google Maps*.

Figura 2 - Funções do sistema



Fonte: Elaborado pelos autores (2018).

A interação do usuário é feita utilizando um menu (painel) que dá acesso as funções de gerenciamento de dados. Um campo de busca de endereços, para localizar novos clientes, e um para buscas de clientes cadastrados encontra-se na barra superior.

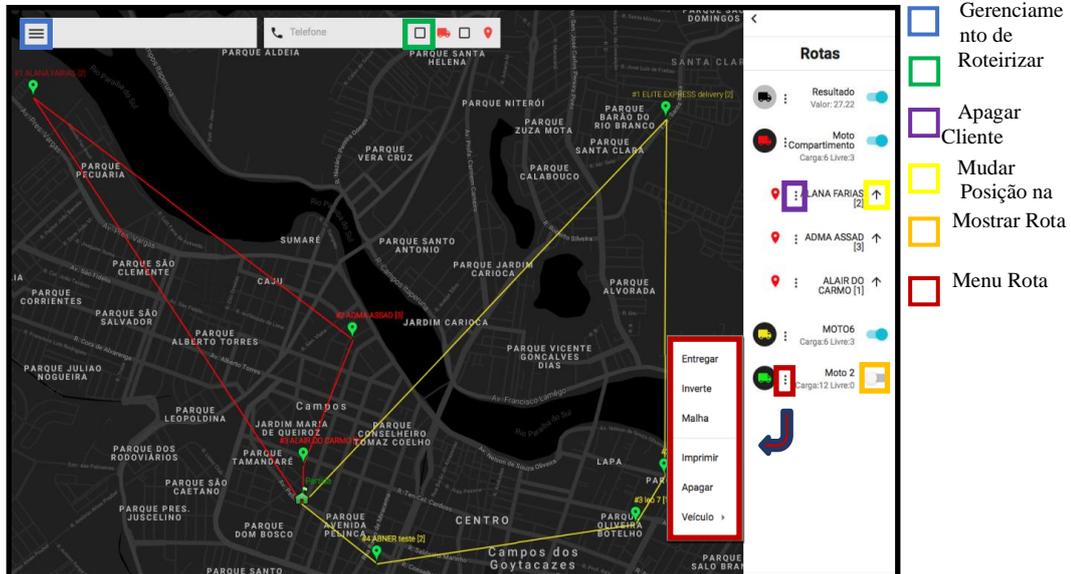
Ao ser digitado um endereço, sua localização é mostrada no mapa e automaticamente a tela de cadastro de um novo cliente é aberta. Toda a interface foi projetada para facilitar a utilização do sistema, bem como ser discreta em relação ao mapa. A Figura 3 mostra as funções de edição de rotas.

O menu Rota fica disponível para cada rota gerada e tem as seguintes opções: *Entregar*, *Inverte*, *Malha*, *Imprimir*, *Apagar* e *Veículo*. A opção *Entregar* grava a rota no servidor de banco de dados e retira os clientes da rota da lista de pedidos. Já a *Inverte*, como o próprio nome sugere, inverte a ordem de entrega dos clientes na rota.

A opção *Malha* gera a rota na rede de vias e considera todas as restrições de trânsito. O sentido da rota é mostrado através de setas ao longo do caminho gerado. Existe uma limitação na API do *Google Maps* de gerar no máximo um roteiro com 25 pontos de entrega. Essa limitação foi contornada gerando um particionamento dos clientes da rota e unindo os roteiros gerados.

O roteiro de entrega pode ser impresso utilizando a opção *Imprimir*. A rota pode ser eliminada por completo utilizando a opção *Apagar*. Essa opção apenas apaga a rota gerada e os pedidos voltam para a lista de clientes com pedidos a serem entregues.

Figura 3 - Rotas Geradas – Edição



Fonte: Elaborado pelos autores (2018).

O veículo que foi alocado para a rota pode ser alterado utilizando a opção *Veículo* que contém a listagem de veículos disponíveis. Quando o veículo é modificado, o sistema acessa o servidor de roteirização para testar se o mesmo atende todas as restrições do novo veículo. Se for viável o veículo é trocado, caso contrário, uma mensagem de alerta é mostrada e o veículo original é mantido na rota. Os clientes podem ser transferidos de uma rota para outra, apenas sendo arrastados para a rota de destino. O servidor de roteirização é acessado para verificar a viabilidade dessa mudança e se nenhuma restrição for violada, as rotas são imediatamente atualizadas no mapa.

4. RESULTADOS E DISCUSSÕES

4.1 Roteirização de veículos com multi-compartimentos

O problema de roteirização de veículos com multi-compartimentos (PRV-MC) aparece pela primeira vez no trabalho de Brown e Graves (1981) e Avella *et al.* (2004), onde é descrito uma aplicação para o problema de distribuição de combustíveis. Nessa aplicação os compartimentos são fixos e a alocação dos produtos pode ser feita em qualquer compartimento. Chajakis e Guignard (2003) estudaram o problema aplicado à distribuição em

lojas de conveniência com veículos com divisórias de tamanhos variáveis e desenvolveram uma solução utilizando relaxação lagrangeana.

O trabalho de El Fallahi *et al.* (2008) descreve o problema num contexto de distribuição de ração para animais em fazendas, onde os tamanhos dos compartimentos são fixos e cada produto só pode ser alocado em um tipo de compartimento. Além disso, os mesmos não podem ser misturados com nenhum produto similar. Foi proposta uma solução para o problema usando algoritmos meméticos e busca tabu.

Muyldermans e Pang (2010) consideraram um problema similar e desenvolveram um algoritmo de busca local guiada para a solução do mesmo. Cornillier *et al.* (2008) estende o problema para múltiplos períodos de entrega. Lahyani *et al.* (2015) apresentam um estudo de caso de distribuição de azeite de oliva, que são categorizados em diferentes compartimentos em função de sua qualidade. Henke *et al.* (2015) apresenta uma versão do problema adaptado a coleta de material reciclado, onde os tamanhos dos compartimentos são flexíveis. Um algoritmo de busca em vizinhança variável (VNS) foi desenvolvido para solução do problema.

O problema de roteirização de veículos com multi-compartimentos pode ser formulado da seguinte forma: dado um grafo não direcionado $G = (P, E)$, o conjunto de vértices $P = \{0, 1, \dots, n\}$ representa os pontos nos quais os veículos deverão visitar, sendo o nó 0 a base de operações onde os veículos irão partir. Os clientes serão visitados por um conjunto de veículos $V = \{1, 2, \dots, v\}$ não idênticos com diferentes custos e capacidades. Cada veículo possui seu conjunto de compartimentos $C = \{1, \dots, m\}$. O conjunto de arestas $E = \{(i, j) \in P, i < j\}$ representa as ligações entre os vértices e tem um custo de deslocamento não negativo $c_{ij} = c_{ji}$. Todos esses custos satisfazem a desigualdade triangular. O objetivo é minimizar a distância total dos veículos.

O problema de roteirização de veículos com multi-compartimentos é caracterizado como *NP-hard*, uma vez que é considerado uma generalização do problema clássico de roteirização. Apenas instâncias pequenas podem ser solucionadas de forma exata. Essa é a justificativa do uso de heurísticas para a solução de problemas reais (EL FALLAHI *et al.*, 2008). Na próxima seção é apresentado o algoritmo desenvolvido para a solução do problema.

4.2 Algoritmo GRASP

O algoritmo GRASP (*Greedy Randomized Adaptive Search Procedures*) é um método iterativo, proposto por Feo e Resende (1995), que consiste de duas fases: uma fase de construção, na qual uma solução viável é gerada a cada iteração, elemento a elemento, e uma fase de busca local na qual a solução é melhorada. O resultado final é a melhor solução obtida nas diferentes iterações do procedimento.

O algoritmo desenvolvido (Algoritmo 1) é baseado no trabalho de Chaovalitwongse *et al.* (2003) e Kontovarids e Bard (1995). A construção de soluções viáveis é iniciada pelo procedimento seleção de sementes (Algoritmo 2), que tem como objetivo inicializar cada nova rota com um consumidor. Esta seleção é feita de modo que os consumidores mais dispersos sejam alocados primeiro e o número de rotas iniciais seja igual ao número mínimo de veículos (I) necessários para que todos os clientes sejam atendidos.

Algoritmo 1. GRASP

```

 $P = P - \{0\}$ 
 $\rho = \text{SeleçãoSementes}(P, V)$ 
 $f^* \leftarrow \infty$ 
for  $k = 1$  to  $iter$  do
 $x = \text{ConstruirSolução}(\rho, \delta_1, \delta_2, \delta_3, \lambda)$ 
 $\dot{x} = \text{BuscaLocal}(x)$ 
if  $(f(\dot{x}) < f^*)$  then
 $s^* = \dot{x}$ 
 $f^* = f(\dot{x})$ 
end if
end for
return  $s^*$ 

```

Algoritmo 2. SeleçãoSementes (P, V)

```

 $S = \{ \}$ 
 $s = \text{Ache o consumidor } c \in P \text{ mais distante do depósito}$ 
while  $|S| < I$  do
 $S = S \cup s$ 
Criar nova rota  $r$  com o cliente  $s$  e associar um veículo viável  $v$  para a rota.
 $\rho = \rho \cup r$ 
 $V = V - v$ 
 $P = P - s$ 
 $s = \text{Ache o consumidor } c \in P \text{ que maximize a soma das distâncias para todos os elementos de } S.$ 
end while
return  $\rho$ 

```

Após o procedimento de seleção de sementes, o algoritmo de construção solução (Algoritmo 3) é chamado. O custo de inserção do consumidor k , na rota r , utilizando o veículo v , entre os consumidores i e j é designado por $C_{i,j,k,r,v}$.

Na formulação da função custo, Solomon (1987) introduziu várias maneiras de calcular o custo de inserção de clientes em rotas, mas experiências computacionais mostram que essas funções são boas para aplicações onde os consumidores estão agrupados e a capacidade do veículo não é considerada. Em razão dessa limitação iremos introduzir a (1) seguinte função de custo, baseada no trabalho de Chaovaitwongseet al. (2003):

$$C_{i,j,k,r,v} = \delta_1 c_{i,j,k,r,v}^1 + \delta_2 c_{i,j,k,r,v}^2 + \delta_3 c_{i,j,k,r,v}^3$$

onde $\delta_1, \delta_2, \delta_3$ são pesos não negativos que irão ponderar os valores de cada componente da função custo e $\delta_1 + \delta_2 + \delta_3 = 1$. A primeira componente $c_{i,j,k,r,v}^1 = VFE_v - \sum_{c=1}^m q_k^c$, onde VFE_v é a capacidade livre do veículo v e q_k^c é a demanda de cada compartimento do cliente k . Essa função designa o custo associado à capacidade livre do veículo e mapeia grandes demandas em pequenos custos, consumidores com altas demandas devem ser associados aos veículos primeiros. Essa lógica é a mesma utilizada para problemas da mochila (MARTELLO e TOTH, 1990). A segunda componente é dada por $c_{i,j,k,r,v}^2 = (d_{ik} + d_{jk} - d_{ij})\alpha_v$, onde α_v é o custo por distância do veículo v . Esta componente mede o custo do aumento da distância com a inserção do consumidor k . A terceira componente insere o custo fixo do veículo v e é dada por $c_{i,j,k,r,v}^3 = F_v$.

No algoritmo 3 apresenta-se a fase de construção da GRASP e sua execução considera inicialmente achar o menor custo de inserção viável em cada rota $r \in \rho$, para cada consumidor $c \in P$ não associado, utilizando o veículo $v \in V$.

Algoritmo 3. ConstruirSolução ($\rho, \delta_1, \delta_2, \delta_3, \lambda$)

```

while  $P \neq \emptyset$  do
  for  $c = 1$  to  $|P|$  do
    for  $r = 1$  to  $|\rho|$  do
      for  $v = 1$  to  $|V|$  do
        if (viável) then Ache o custo de inserção do cliente  $c$  entre os clientes  $i$  e  $j$ ,  $C_{i,j,c,r,v} = \min\{C_{i,j,c} \ \forall i,j \in r\}$ 
          end for
        for  $c = 1$  to  $|P|$  do
          Ache a rota  $r^*$  de menor custo de inserção  $C_{c,r^*,v} = \min\{C_{c,r,v} \ \forall r \in \rho, \forall v \in V\}$ 
          end for
        for  $c = 1$  to  $|P|$  do
          Ache o custo de penalização  $P_c = \sum (C_{c,r,v} - C_{c,r^*,v}) \ \forall r \in \rho, \forall v \in V$ 

```

GEPROS. Gestão da Produção, Operações e Sistemas, v. 15, nº 3, p. 133 - 154, 2020.

```
end for
LCR = Construa uma lista com os  $\lambda$  maiores custos de penalização ( $P_c$ ).
 $\hat{c}$  = Selecione um cliente aleatoriamente da LCR para ser roteirizado.
 $P = P - \hat{c}$ 
end while
return  $x$ 
```

Então é calculado uma penalidade P_c , que mede o custo que se deve pagar posteriormente, se o consumidor não for associado a sua rota de menor custo. Consumidores com altos valores de penalização devem ser associados a uma rota primeiro, enquanto os de menores valores podem esperar. A lista de candidatos restritos (LCR) é construída com os λ maiores custos de penalização, e o cliente a ser roteirizado é escolhido de forma aleatória.

4.3 Busca Local

São utilizados dois tipos de movimentos para tentar melhorar a solução. O primeiro (*Move*), tenta realocar os consumidores para uma rota de menor custo. O procedimento é inicializado com as rotas que possuem menos consumidores. Se alguma rota apresentar no final uma quantidade nula de consumidores, a mesma é eliminada da solução, liberando a utilização do veículo. O segundo movimento é o clássico 2-opt (LIN, 1965) que consiste em mudar dois arcos não adjacentes $(i, i+1)$ e $(j, j+1)$ por (i, j) e $(i+1, j+1)$ numa mesma rota. Foi implementado uma variação para o caso de duas rotas distintas (2-opt *inter-route*). Dois arcos $(i, i+1)$ e $(j, j+1)$ de duas rotas r e r' são removidos, e as rotas são reconectadas pela inserção dos arcos $(i, j+1)$ e $(j, i+1)$. Todas as restrições devem ser verificadas antes da mudança de rota. Se algum veículo for incompatível, tenta-se trocar os veículos das rotas ou utilizar algum outro veículo disponível. Todos os movimentos tem uma complexidade de $O(|P|^2)$.

4.4 Resultados Computacionais

O algoritmo foi codificado em *object pascal* utilizando técnicas de orientação a objetos. As seguintes classes foram implementadas para facilitar o aproveitamento do código em futuras implementações: *TClient*, *TVehicle*, *TDemand*, *TCompartment*, *TRoute*, *TRouting*, *TGrasp*, *TRoutes*, *TVehicles*, *TClients* e *TLocalSearch*. As instâncias utilizadas para avaliação da heurística são as utilizadas no trabalho de El Fallahi *et. al.* (2008) que utilizam veículos com dois compartimentos. Inicialmente o autor dividiu a demanda de cada cliente aleatoriamente em duas partes. A demanda do primeiro compartimento seguiu a equação $q_{il} =$

GEPROS. Gestão da Produção, Operações e Sistemas, v. 15, nº 3, p. 133 - 154, 2020.

q_i / k , sendo q_i a demanda obtida no problema clássico de roteirização (VRP) e k um número inteiro entre 3 e 5. A demanda do segundo compartimento é a diferença entre a demanda total e a do primeiro compartimento. Para o cálculo da capacidade dos compartimentos (veículos), utilizou-se a demanda média. Sendo c o índice de cada compartimento, emprega-se a seguinte fórmula: $Q_c = (Q \times \bar{Q}_c) / (\bar{Q}_1 + \bar{Q}_2)$.

A técnica adotada possui uma particularidade, cada cliente deve ser atendido somente por um veículo, ou seja, não é permitida a divisão de entrega por mais de um veículo. As instâncias 6 até 10, 13 e 14 incluem restrição de tamanho da rota. Essa restrição foi incorporada ao algoritmo. Após alguns experimentos iniciais, os parâmetros utilizados pelo algoritmo foram fixados em: $iter = 300$; $\lambda = 6$; $\delta_1 = 0.1$; $\delta_2 = 0.8$; $\delta_3 = 0.1$. Os testes foram feitos em um *PC Intel Core i5-2400 at 3.1 Ghz*.

Os resultados obtidos foram comparados com os trabalhos de El Fallahi *et al.* (2008), e Mendoza *et al.* (2010). El Fallahi *et al.* (2008) utilizou Busca Tabu (TS) e Algoritmo Memético (MA) para resolver o problema, sendo esta última a mesma escolhida por Mendoza *et al.* (2010). Na Tabela 1, os resultados da comparação são expostos, onde Instância, n e Ganho (%) indicam, respectivamente, o nome da instância testada, o número total de clientes e a diferença relativa entre a solução obtida pela GRASP e a melhor solução já encontrada.

Tabela 1- Comparação de Resultados.

Instância	n	El Fallahi et al. (2008)		Mendoza et al. (2010)	Artigo	Tempo (s)	Melhor
		MA - MC-VRP-WS	TS - MC-VRP-WS	MA/SCS	GRASP	GRASP	Ganho (%)
vrpnc1	50	558.8	556.1	550.2	552.1	3	0.3
vrpnc2	75	888.6	863.6	884.3	900.1	10	4.2
vrpnc3	100	878.4	837.6	852.7	872.9	20	4.2
vrpnc4	150	1089.1	1070.7	1080	1169.8	56	9.3
vrpnc5	199	1408.5	1361.4	1403.5	1512.8	108	11.1
vrpnc6	50	569.4	563.4	562.7	562.5	4	-0.0
vrpnc7	75	955.1	949	946.3	916.1	13	-3.2
vrpnc8	100	958.9	916.2	874.7	904.5	24	3.4
vrpnc9	150	1262.7	1290.8	1240.5	1190.1	69	-4.1

GEPROS. Gestão da Produção, Operações e Sistemas, v. 15, nº 3, p. 133 - 154, 2020.

vrpnc10	199	1509.1	1490.2	1515.6	1536.3	133	3.1
vrpnc11	120	1122.9	1201.6	1162.4	1180.4	30	1.5
vrpnc12	100	926.5	934.1	916.6	986.8	19	7.7
vrpnc13	120	1542.4	1582.3	1607.4	1296.1	37	-16.0
vrpnc14	100	966.5	1141.6	937.6	994.2	24	6.0
Média		1045.5	1054.2	1038.2	1041.1	39.3	0.3

Fonte: Autores.

Os resultados encontrados confirmam o bom desempenho do algoritmo desenvolvido. O custo médio obtido nas 14 instâncias testadas mostra-se inferior aos trabalhos de El Fallahi *et al.* (2008). Também se assemelha ao menor custo médio encontrado na literatura, por Mendoza *et al.* (2010), excedendo em apenas 0,3% . Cabe salientar que em 4 (vrpnc6, vrpnc7, vrpnc9 e vrpnc13) das 14 instâncias o custo encontra-se abaixo dos valores obtidos pelos trabalhos citados.

É interessante informar que a estratégia de implementação do algoritmo pressupõe a utilização de uma frota heterogênea e limitada, devido às limitações e condições encontradas na maioria das empresas. Tal fato evidencia uma importante diferença na forma com a qual o algoritmo busca resolver as instâncias, voltadas prioritariamente para uma frota homogênea e ilimitada, pois é assim que El Fallahi *et al.* (2008) buscam roteirizar através de seu algoritmo. Tal particularidade na frota também é utilizada por Mendoza *et al.* (2010).

5. ESTUDO DE CASO

Com a finalidade de testar o desempenho do sistema numa empresa, foi realizado um estudo de caso na concessionária Águas do Paraíba S/A, integrante do Grupo Águas do Brasil. A mesma é responsável pelos serviços de saneamento no município de Campos dos Goytacazes.

O sistema de tratamento e distribuição de água no município possui 25 estações de tratamento de água pela cidade. Atualmente, 99% da população de Campos dos Goytacazes possui acesso à água potável e ainda 90% do efluente de esgoto produzido é coletado e totalmente tratado. Com a inauguração da Estação de Esgoto Esplanada, em 2018, Campos dos Goytacazes passa a ter sete estações e tratamento de esgoto em operação, 24 horas por dia, em ambas as margens do rio Paraíba do Sul.

5.1. Setor Estudado

O setor escolhido para estudo dentro da empresa foi a Gestão de Serviços Operacionais (GSO). Trabalham no GSO onze funcionários, onde dois desses são programadores de serviços. Os programadores de serviços são responsáveis por gerenciar todas as ordens de serviços (OS's) geradas no sistema para o setor, programando e roteirizando cada atividade para a equipe responsável. Semanalmente o setor recebe em média 250 OS's, com demanda variando de acordo com o dia da semana. Feriados e fins de semana tendem a receber menos solicitações de serviços.

Atualmente o GSO possui oito equipes que executam tarefas relacionadas à água e três equipes para as tarefas de esgoto. No entanto, a rotina de trabalho é através de escala e sempre terão equipes de folga a cada dia, variando a disponibilidade total das mesmas. A capacidade diária de cada equipe é de nove horas de trabalho.

Dentre todos os serviços executados pela empresa, o GSO se concentra apenas no segmento de operações. O tempo padrão é o tempo médio de execução de cada serviço. É através deste que os programadores de serviços distribuem as OS's para as equipes, sem que ultrapassem o tempo limite de disponibilidade de cada uma. O tempo padrão também serviu de parâmetro para a realização da programação pelo *software* utilizado nesse estudo, onde a capacidade de cada equipe diária foi de 540 minutos e a demanda de cada serviço de acordo com seu respectivo tempo padrão.

5.2. Descrição do Problema

Segundo último censo do IBGE (2018), o município onde a empresa atua tem ao todo 503.424 habitantes, distribuídos em 4.026.696 km² de extensão territorial. O município ainda apresenta um grande número de distritos e localidades espalhados nos mais diversos pontos do território. Essa característica demanda à empresa uma extensa malha de redes para o transporte da água potável aos moradores. Muitas dessas redes ainda são antigas e sofrem degradação devido a desgastes naturais causado pelo tempo, necessitando de reparos constantes por parte da equipe operacional.

A solicitação de serviços em redes torna-se, portanto, a maior demanda do GSO. Ainda, outros setores da empresa possuem suas próprias solicitações de serviços e passam pelas mesmas dificuldades de atendimento devido a tal aspecto territorial. Atender as OS's no

prazo, com um roteiro de custo mínimo e sem afetar a qualidade do serviço prestado ao cliente torna-se uma tarefa primordial nas operações da empresa.

Outro fator que agrava consideravelmente o gerenciamento interno das operações é a ausência de um sistema de controle das OS's, que permita a distribuição dos serviços entre as equipes, com um itinerário que reduza a distância total percorrida, o custo de atendimento e o tempo de locomoção. Esta tarefa ainda é feita manualmente pelos programadores de serviço que, apesar de muita experiência e conhecimento das características do município, não possuem a velocidade e eficiência de um algoritmo para a resolução do problema.

Fica evidente que com a forma na qual são coordenadas as atividades na empresa, alguns problemas surjam no dia a dia operacional. Sobrecarga e ociosidade de equipes, além de serviços executados fora do prazo devido à falta de tempo para locomoção são alguns dos exemplos que acontecem corriqueiramente.

5.3. Dados

O primeiro passo do estudo foi obter junto ao GSO a programação semanal de serviços das equipes de água. A programação fornecida foi a semana do dia 21/09/2018 ao dia 27/09/2018. Em cada dia eram descritas as atividades a serem executadas e para qual equipe o serviço era designado. Posteriormente, cada OS's foi inserida no sistema, de acordo com sua localização e com demanda prevista pelo tempo padrão de serviço. Ao todo, 240 solicitações foram introduzidas.

Em seguida, as oito equipes de água disponíveis para atendimento das solicitações foram inseridas. A capacidade de cada uma das equipes foi baseada na disponibilidade de horário diária dos funcionários. Para início da roteirização, as solicitações de cada dia eram selecionadas e então a rota do dia era gerada.

5.4. Análise dos Resultados

Inicialmente foi realizada uma análise qualitativa da disposição de cada rota no mapa. Essa análise é necessária para uma melhor visualização de como o roteirizador está se comportando e se satisfaz a condição de uma boa distribuição das ordens de serviço. Um roteiro somente era aceito se essa condição se cumprisse.

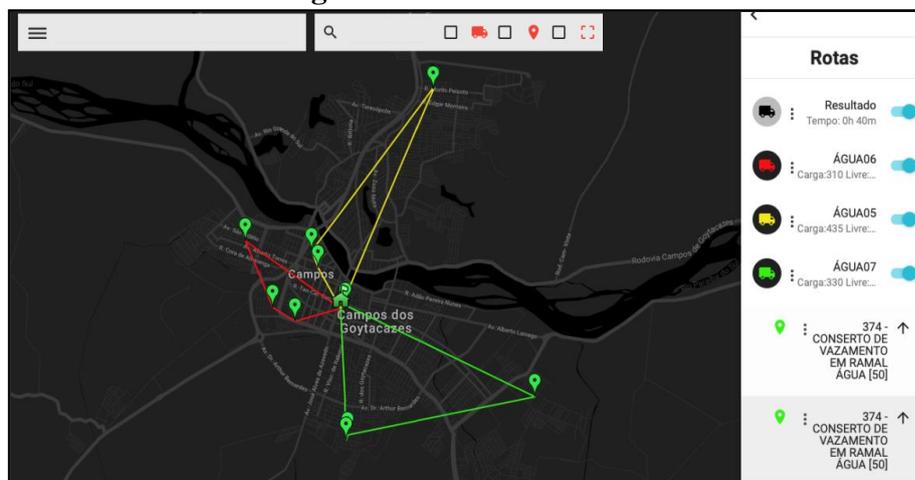
Na Figura 4, que representa o roteiro do dia 23/09/2018, percebe-se claramente uma boa distribuição das rotas entre as equipes, onde cada uma delas possui sua zona de operação e seus caminhos não se cruzam em nenhum momento.

Após a aprovação de cada roteiro pelo operador, eram então registrados os dados de cada um deles. Um roteiro eficaz deverá possuir um número de equipes igual ou inferior ao utilizado pela empresa no dia, trazendo uma redução de gastos com pessoal e equipamentos. Os dados foram registrados e estão apresentados na Tabela 2.

Ao analisarmos os resultados, fica evidente o ganho que a utilização do sistema proporciona. Destaca-se que em nenhum dos dias estudados o número de equipes previstas ultrapassou o de utilizadas realmente em campo.

Além disso, em 3 dos 7 dias com a utilização do roteirizador, a empresa conseguiria reduzir o número de equipes a serem utilizadas para executarem os serviços demandados, reduzindo consideravelmente o custo operacional para a concessionária.

Figura 4 – Roteiro: 23/09/2018



Fonte: Autores.

O sistema ainda proporciona a funcionalidade de cálculo de custo de acordo com a distância percorrida. O cálculo do custo é baseado na distância utilizando coordenadas planas xy de cada serviço localizado no mapa. Para tanto, utilizou-se o resultado do dia 27/09/2018, dia de maior número de serviços requisitados, e foi feito um comparativo com o custo que a rota de campo gerou para a empresa e o custo que a empresa teria se utilizasse as rotas do roteirizador.

Tabela 2 - Comparação de equipes utilizadas e previstas.

Data	21/9	22/9	23/9	24/9	25/9	26/9	27/9
Equipes Utilizadas no campo	5	3	3	5	5	6	8
Equipes Previstas pelo sistema	5	3	3	5	4	3	5

Fonte: Autores.

A análise foi feita inserindo no sistema rota a rota efetuada pela empresa em campo e registrando o custo (distância) de cada uma delas. Logo em seguida, foi feito o roteiro no sistema com o devido registro das distâncias geradas.

Pela Tabela 3 pode-se observar uma redução de 35% no total da distância percorrida pelas equipes quando utilizado o *software* de roteirização. Este resultado é bem compreensível, isto porque conforme o número de solicitações de serviços aumenta, a gestão manual de roteiros tende a apresentar mais falhas e ociosidade de equipes.

A tendência de crescimento das solicitações de serviços evidencia a necessidade de incorporação de um sistema de gerenciamento mais eficaz dos roteiros das equipes na empresa.

Tabela 3 - Comparação entre os custos - distância de rota

Equipes	Custo Campo	Custo Roteirizador
1	3693,7	3111,3
2	2103,6	2551,3
3	10266,5	9549,8
4	14474,3	8530,3
5	11388,9	15938,9
6	12296,2	-
7	3078,3	-
8	2446,8	-
Total	60748,3	39681,6

Fonte: Autores.

6. CONCLUSÕES

O presente estudo teve como objetivo descrever o desenvolvimento do sistema de roteirização Geo-Rota web. A arquitetura distribuída de processamento utilizada permite acesso ao sistema a partir de qualquer dispositivo conectado a internet: celulares, *tablets* e computadores com baixa capacidade de processamento. A API do *Google Maps* permitiu o desenvolvimento do sistema de forma eficiente e rápida. As rotinas de edição de rotas foram eficientes para modificar as rotas geradas pelo algoritmo de roteirização.

O estudo de caso realizado permitiu testar o desempenho do sistema em um ambiente empresarial real. A utilização por parte dos operadores da empresa mostrou-se de fácil aprendizado e uso. Durante a semana estudada, em todos os dias a utilização do sistema trouxe ganhos operacionais para a empresa e redução dos veículos utilizados. A heurística GRASP desenvolvida, utilizada no servidor de roteirização, teve bom desempenho com tempo de solução compatível para acesso via *web*.

O servidor de roteirização utiliza distância euclidiana para o cálculo de custos, uma melhoria a ser desenvolvida seria a geração de uma matriz de distâncias na própria rede de vias. Adicionalmente, fazem-se necessários mais estudos para que o sistema possa tratar casos mais condizentes com a realidade, como por exemplo, roteirização dinâmica de veículos, inclusão de restrição de janela de tempo, coleta e entrega, múltiplos depósitos de partida, dentre outros.

Referências

AMAL, L., SON, L.H., CHABCHOUB H. Spatial GIS-based genetic algorithm for route optimization of municipal solid waste collection. **Environmental Science and Pollution Research**, v. 25, p. 27569–27582, 2018.

ANGEHRN A.A., LUTHI H.-J. Intelligent support systems: a visual interactive approach. **Interfaces**, v. 20, n. 6, p. 17–28, 1991.

AVELLA, P., BOCCIA, M., SFORZA, A. Solving a fuel delivery problem by heuristic and exact approaches. **European Journal of Operational Research**, v. 152, n. 1, p. 170–179, 2004.

BASNET C., FOULDS L., IGBARIA M. Fleet manager: a microcomputer-based decision support system for vehicle routing. **Decision Support Systems**, v. 16, n. 3, p. 195–207, 1996.

BROWN, G., GRAVES, G. Real-time dispatch of petroleum tank trucks. **Management Science**, v. 27, p. 19–32, 1981.

CHAJAKIS, E.D.; GUIGNARD, M. Scheduling deliveries in vehicles with multiple compartments. **Journal of Global Optimization**, v. 26, p. 43-78, 2003.

CHAOVALITWONGSE, W.; KIM, D.; PARDALOS, P. M. GRASP with a new local search scheme for vehicle routing problems with time windows. **Journal of Combinatorial Optimization**, v. 7, n. 2, p. 179-207, 2003.

CORNILLIER, F.; BOCTOR, F. F.; LAPORTE, G.; RENAUD, J., A heuristic for the multi-period petrol station replenishment problem. **European Journal of Operational Research** v. 191, p. 295-305, 2008.

EL FALLAHI, A.; PRINS, C.; CALVO, R. W. A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. **Computers & Operations Research**, v. 35, n. 5, p. 1725-1741, 2008.

FEO, T. A., RESENDE, M.G.C. Greedy Randomized Adaptive Search Procedures. **Journal of Global Optimization**, v. 6, p. 109-133, 1995.

FIELDING, R. T. **Architectural Styles and the Design of Network-based Software Architectures**. Doctoral dissertation, University of California, Irvine, 2000.

FAULIN J., SAROBE P., SIMAL J. The DSS LOGDIS optimizes delivery routes for FRILAC's frozen products. **Interfaces**, v. 35, n. 3, 202–214, 2005.

GAYIALIS S.P., TATSIPOULOS I.P. Design of an IT-driven decision support system for vehicle routing and scheduling. **European Journal of Operational Research**, v. 152, n. 2, 382–398, 2004.

HENKE, T., SPERANZA, M. G. WÄSCHER, G. The multi-compartment vehicle routing problem with flexible compartment sizes. **European Journal of Operational Research**, v. 246, n. 3, p. 730–743, 2015.

KEENAN, P. B. Spatial decision support systems for vehicle routing. **Decision Support Systems**, v. 22, n. 1, p. 65–71, 1998.

KONTORAVDIS, G.; BARD, JONATHAN F. A GRASP for the vehicle routing problem with time windows. **ORSA journal on Computing**, v. 7, n. 1, p. 10-23, 1995.

KRAMER, R. H. F. R., SUBRAMANIAN, A., PENNA, P. H. V. (2016). Asymmetric vehicle routing problem with heterogeneous limited feet: A case study in a beverage industry. **Gestão e Produção**, v. 23, n. 1, p. 165–176, 2016.

- LAHYANI, R., LAPORTE, G., COELHO, L. C., KHEMAKHEM, M. A Multi-Compartment Vehicle Routing Problem Arising in the Collection of Olive Oil in Tunisia. **Omega**, v. 51, p. 1–10, 2015.
- LAPALME G., ROSSEAU J.-M. Georoute: a geographic information system for transportation applications. **Commun ACM**, v. 35, n. 1, p. 81–88, 1992.
- LAPORTE, G. M. GENDRAU, J.Y. POTVIN, F. SEMET. Classical and modern heuristics for the vehicle routing problem. **International Transaction in Operational Research**, v.7, p. 285-300, 2000.
- LIN, S. Computer solutions of the traveling salesman problem. **Bell System Technical Journal**, v. 44, p. 2245–2269, 1965.
- LI, R., QI, M., CHENG, C., LAI, W. Design of Dynamic Vehicle Routing System Based on Online Map Service. **The 13th International Conference on Service Systems and Service Management**, p. 1–5, 2016.
- MARTELLO S., TOTH P. Lower Bounds and Reduction Procedures for the Bin Packing Problem. **Discrete Applied Mathematics** v. 28, p. 59-70, 1990.
- MENDOZA, J. E., CASTANIER, B., GUÉRET, C., MEDAGLIA, A. L., VELASCO, N. A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands. **Computers and Operations Research**, v. 37, n. 11, p. 1886–1898, 2010.
- MENDOZA, J. E., MEDAGLIA, A. L., VELASCO, N. An evolutionary-based decision support system for vehicle routing: The case of a public utility. **Decision Support Systems**, v. 46, n. 3, p. 730–742, 2009.
- MUYLDERMANS, L., PANG, G. On the benefits of co-collection: Experiments with a multi-compartment vehicle routing algorithm. **European Journal of Operational Research**, v. 206, n. 1, p. 93–103, 2010.
- NUSSBAUM M., SEPULVEDA M., COBIAN A., GAETE J., CRUZ J. A fuel distribution knowledge-based decision support system. **International Journal of Management Science**, v. 25, n. 2, p. 225–234, 1997.
- PAMUČAR, D., GIGOVIĆ, L., ČIROVIĆ, G., REGODIĆ, M. Transport spatial model for the definition of green routes for city logistics centers. **Environmental Impact Assessment Review**, v. 56, p. 72–87, 2016.
- SOLOMON, MARIUS M. Algorithms for the vehicle routing and scheduling problems with time window constraints. **Operations research**, v. 35, n. 2, p. 254-265, 1987.
- SANTOS, L., COUTINHO-RODRIGUES, J., CURRENT, J. R. Implementing a multi-vehicle multi- route spatial decision support system for efficient trash collection in Portugal. **Transportation Research Part A: Policy and Practice**, v. 42, n. 6, p. 922–934, 2008.

SANTOS, L., COUTINHO-RODRIGUES, J., ANTUNES, C. H. A web spatial decision support system for vehicle routing using Google Maps. **Decision Support Systems**, v. 51, n. 1, p. 1–9, 2011.

TARANTILIS, C., KIRANOUDIS, C. Using a spatial decision support system for solving the vehicle routing problem, **Information and Management**, v. 5, n. 39, p. 359–375, 2002.

WEIGEL D., CAO B. Applying GIS and OR techniques to solve Sears technician dispatching and home delivery problems. **Interfaces**, v. 29, n. 1, p. 112–130, 1999.

WUNDERLICH J., COLLETE M., LEVY L., BODIN L. Scheduling meter readers for southern California gas company. **Interfaces**, v. 22, n. 3, p. 22–30, 1992.