



## A metaheurística ILS e a API do GoogleMap aplicados na resolução do problema de roteamento de veículos

Francisco Regis Abreu Gomes<sup>1</sup>, Flaviana de Lima Araújo<sup>2</sup>

<sup>1</sup>Professor do curso de Tecnologia em Agronegócio – IFCE Quixadá. e-mail: regisgomes@ifce.edu.br

<sup>2</sup>Tecnolanda do curso de Tecnologia em Agronegócio – IFCE Quixadá. Bolsista de iniciação científica. e-mail: flaviana.ar@hotmail.com

**Resumo:** O Problema de Roteamento de Veículos (PRV) é um dos problemas mais estudados pela Pesquisa Operacional devido a sua aplicação em empresas que lidam com transporte de produtos e pessoas. Os roteirizadores são sistemas computacionais desenvolvidos para solucionar o PRV usando algoritmos e uma base de dados geográfica a partir de um Sistema de Informação Geográfica (SIG). Na prática, um SIG pode custar muito caro para a maioria das empresas, inviabilizando a resolução racional do PRV. Por isso, esse trabalho apresenta a utilização da tecnologia API do GoogleMaps como forma de substituir a aquisição de um SIG e, assim, permitir que um maior número de empresas otimizem sua função de transporte. Foi proposto também um algoritmo baseado na metaheurística *Iterated Local Search* (ILS) para a resolução do PRV. Para ilustrar a utilização da API do GoogleMaps e do algoritmo, foi criado um exemplo de PRV com 1 depósito e 24 clientes. Usando essa API, foi construída a matriz de tempos e apresentada a visualização das rotas construídas. O algoritmo proposto resolveu o PRV em tempo hábil para aplicações reais e a referida API substituiu satisfatoriamente o SIG.

**Palavras-chave:** API do GoogleMaps, *Iterated Local Search*, problema de roteamento de veículos.

### 1. INTRODUÇÃO

As empresas estão em constante busca pela redução de custos por meio de maior eficiência nos processos de negócio. Nesse sentido, a logística das empresas tem sido objeto de esforços para redução de custos, sem esquecer da qualidade. Novas tecnologias são desenvolvidas com o objetivo de proporcionar melhor desempenho operacional às atividades logísticas das empresas, como forma de reduzir os custos logísticos.

O transporte de produtos é o componente mais importante nos custos logísticos de uma empresa. Um estudo realizado pelo Instituto de Logística e *Supply Chain* (ILOS) identificou que os custos logísticos das organizações brasileiras somaram 10,6% do PIB no ano de 2010, onde 54% correspondem a custos de transporte. Isso representa 4,6% da receita líquida das empresas, sendo que os custos de estoque e armazenagem representam 23% cada (VALOR ONLINE, 2012).

Além disso, os clientes estão mais exigentes no que diz respeito a prazos de entrega cada vez menores, gerando uma concorrência crescente entre as empresas de distribuição de produtos, sendo um dos fatores mais importantes na obtenção de vantagem competitiva e conquista de participação de mercado (MELO e FILHO, 2001). Essa exigência dos clientes pressiona os custos de transporte das empresas.

Um dos principais problemas de decisão presentes na gestão de transporte das empresas é o Problema de Roteamento de Veículos (PRV). Segundo Laporte *et al.* (2000), o PRV consiste em definir roteiros de veículos, ou seja, a sequência de pontos a serem visitados que minimizem o custo total de atendimento, cada um dos quais inicia e termina na base dos veículos. Tal medida assegura que cada ponto seja visitado exatamente uma vez e a demanda total em qualquer rota não exceda a capacidade do veículo que a atende.

A Pesquisa Operacional é uma área do conhecimento dedicada a estudar problemas de decisão e propor métodos para sua resolução. Problemas de decisão são aqueles que possuem um grande número de alternativas que satisfazem as restrições práticas, mas normalmente existe apenas uma solução que é a melhor entre todas. A Pesquisa Operacional possui várias abordagens diferentes para tratar o PRV. Uma dessas abordagens é a metaheurística eficaz e de fácil implementação, conhecida como *Iterated Local Search* (ILS), apresentada em Lourenço *et al.* (2002). Stützle (1998) aplicou o ILS ao problema



de sequenciamento permutacional *flowshop* com sucesso. Essa foi a técnica de resolução do PRV usada nesse trabalho.

Uma etapa anterior à aplicação do ILS na resolução de PRV é a determinação do tempo de viagem ou as distâncias entre os pontos a serem visitados na rota. Cunha (2000) afirma que esse tema é pouco discutido em trabalhos acadêmicos mas de fundamental importância na resolução de problemas reais.

Uma forma de obter esses dados é por meio de Sistemas de Informação Geográfica (SIG). No entanto, segundo Bodin (1990), implementar e manter um SIG representa um custo alto para as empresas, principalmente em cidades maiores, devido às mudanças constantes de direção de vias e criação de novas ruas. Essas dificuldades poderiam inviabilizar a utilização de sistemas computacionais de resolução de PRV.

Melo e Filho (2001) confirmam a preocupação com os custos de um SIG e vão além: a prática mostrou que tais sistemas mal implantados e gerenciados constituem uma fonte permanente de problemas e prejuízos. Sugerem até que seja estudada a possibilidade de desenvolver um sistema próprio antes de adquirir tais sistemas.

Uma forma de superar os custos para desenvolvimento e manutenção de um SIG é utilizar ferramentas gratuitas. Uma dessas opções é a GoogleMaps JavaScript API v.3 (GOOGLE MAP API, 2012). Essa API permite que seja incorporado o GoogleMaps a páginas de web desenvolvidas para uso particular. A API fornece diversos utilitários para manipulação de mapas (como os disponíveis na página web <http://maps.google.com>) e adição de conteúdo aos mapas por meio de vários serviços, permitindo criar aplicativos de mapas robustos em um site. A API do GoogleMaps é um serviço gratuito, disponível para qualquer implementação em html com acesso a internet.

O PRV, mesmo sendo estudado há muito tempo, ainda é importante à proposição de novas abordagens para sua resolução, pois, nas palavras de Couto (2004) relatadas por Enomoto e Lima (2007), “as heurísticas existentes para resolver os problemas das rotas e entregas, nos *softwares* especializados no mercado, tendem a ser muito generalistas e não costumam gerar resultados satisfatórios”.

O objetivo desse trabalho foi mostrar o uso da API do GoogleMaps como um substituto a um SIG comercial, sendo assim uma forma de viabilizar a aplicação de métodos de Pesquisa Operacional – no caso, o ILS na resolução de PRV.

## 2. MATERIAL E MÉTODOS

### 2.1 Problema de Roteamento de Veículos

O PRV já foi definido na introdução deste trabalho. A seguir, foi descrito e apresentado o modelo matemático clássico do PRV de acordo com Arenales *et al.* (2007).

O PRV é representado por um grafo orientado completo  $G = (N, E)$ , em que  $N = C \cup \{0, n+1\}$ ,  $C = \{1, \dots, n\}$  é o conjunto de nós que representam os clientes, e 0,  $n+1$  são os nós que representam o depósito. O conjunto  $E = \{(i, j) : i, j \in N, i \neq j, i \neq n+1, j \neq 0\}$  corresponde aos arcos associados às conexões entre nós. Nenhum arco termina no nó 0 e nenhum arco começa no nó  $n+1$ . Todas as rotas começam em 0 e terminam em  $n+1$ . Um custo  $c_{ij}$  e um tempo de viagem  $t_{ij}$  estão associados a cada arco  $(i, j) \in E$ . O tempo de viagem  $t_{ij}$  inclui o tempo de serviço do cliente  $i$ . Cada cliente  $i$  tem uma demanda  $d_i$ . Um conjunto  $K$  de veículos idênticos, cada veículo  $k \in K$  com capacidade  $Q$ , está situado no depósito. O objetivo é minimizar o custo total de viagens, sujeito às seguintes restrições:

- Cada rota inicia e termina no depósito.
- Cada cliente pertence somente a uma rota.
- A demanda total de uma rota não pode exceder a capacidade  $Q$  do veículo.
- O tempo de viagem de uma rota não pode exceder o limite  $D$ .

As variáveis de decisão do problema são:



$$x_{ijk} \begin{cases} 1, & \text{se o veículo } k \text{ percorre a arco } (i, j), \forall k \in K, \forall (i, j) \in E. \\ 0, & \text{caso contrário.} \end{cases}$$

e o modelo matemático é o seguinte:

$$\text{Min} \sum_{k \in K} \sum_{(i,j) \in E} c_{ij} x_{ijk} \quad (1)$$

$$\sum_{k \in K} \sum_{j \in N} x_{ijk} = 1, \quad \forall i \in C \quad (2)$$

$$\sum_{i \in C} \sum_{j \in N} d_i x_{ijk} \leq Q, \quad \forall k \in K \quad (3)$$

$$\sum_{i \in N} \sum_{j \in N} t_{ij} x_{ijk} \leq D, \quad \forall k \in K \quad (4)$$

$$\sum_{j \in N} x_{0jk} = 1, \quad \forall k \in K \quad (5)$$

$$\sum_{i \in N} x_{ihk} - \sum_{j \in N} x_{hjk} = 0, \quad \forall h \in C, \quad \forall k \in K \quad (6)$$

$$\sum_{i \in N} x_{i,n+1,k} = 1, \quad \forall k \in K \quad (7)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijk} \leq |S| - 1, \quad S \subset C, 2 \leq |S| \leq \left\lfloor \frac{n}{2} \right\rfloor, \quad \forall k \in K \quad (8)$$

$$x \in B^{k|E|} \quad (9)$$

A função objetivo (1) representa a minimização do custo total das rotas. O conjunto de restrições (2) assegura que cada cliente  $i$  é designado a um único veículo. O conjunto de restrições (3) impõe que a demanda total de cada rota do veículo  $k$  não exceda a capacidade  $Q$  do veículo. O conjunto de restrições (4) garante que a duração de cada rota do veículo  $k$  não exceda o limite  $D$ . Os conjuntos de restrições (5), (6) e (7) representam restrições de fluxo em redes que exigem que cada veículo  $k$  parta do depósito (nó 0) somente uma vez, deixe o nó  $h$ , se e somente se, entrar nesse nó, e retorne ao depósito (nó  $n + 1$ ) somente uma vez. O conjunto de restrições (7) é redundante mas é mantido no modelo para enfatizar a estrutura de redes. O conjunto de restrições (8) garante a não-existência de sub-rotas e, finalmente, o conjunto de restrições (9) indica o tipo da variável.

Se o número de veículos no modelo matemático for um parâmetro fixo ou um limitante superior igual a  $K$ , o arco  $(0, n + 1)$  é removido. Caso contrário, se o número de veículos é uma variável, é atribuído um custo  $c_v$  a cada veículo usado. Isso é feito impondo-se  $c_{0,n+1} = -c_v$ . Se  $c_v$  é grande, o modelo matemático primeiramente minimiza o número de veículos e, em segundo lugar, minimiza o custo das viagens.



O PRV pertence à classe de problemas conhecida como *NP*-difícil (do inglês *NP-hard*). Isso significa que possui ordem de complexidade exponencial, ou seja, o esforço computacional para sua resolução ótima cresce exponencialmente em relação ao tamanho do problema (número de clientes a serem atendidos). Na prática, isso significa que não é possível resolver até a otimalidade um PRV de tamanho real. A opção é usar heurísticas que são capazes de encontrar boas soluções.

Muitas empresas no Brasil abordam o PRV por meio de métodos empíricos ou técnicas simples, sem considerar modelos matemáticos. Segundo Enomoto e Lima (2007), a maioria das empresas no Brasil emprega profissionais especialmente treinados para a execução das atividades de roteirização. Quando os pedidos dos clientes chegam à empresa, os funcionários os separam e agrupam por algum critério, que pode ser espacial ou temporal. Esse método, mesmo que simplifique o trabalho, acarreta em soluções distantes do ótimo.

Por isso, foram desenvolvidos sistemas computacionais usando-se heurísticas para resolução do PRV conhecidos como Sistemas de Roteirização e Programação de Veículos ou, simplesmente, Roteirizadores. Segundo Melo e Filho (2001), no Brasil, atualmente são comercializados diversos sistemas de roteirização, sendo a maioria deles ainda desenvolvidos no Exterior com heurísticas de resolução que, geralmente, não são disponibilizadas pelos seus desenvolvedores.

Os roteirizadores são usados por empresas especializadas em transporte de cargas, como transportadoras ou operadores logísticos, ou empresas que têm na função de transporte um importante fator competitivo, como empresas de envasamento de bebidas, fábricas de alimentos e distribuidores.

Mesmo algumas empresas que adquiriram roteirizadores não usam toda sua potencialidade. Visto o que dizem Enomoto e Lima (2007), um grande atacadista localizado no sul de Minas Gerais, apesar de utilizar diversos conceitos e ferramentas de apoio em seu processo de distribuição física, especialmente um *software* robusto de roteirização, ainda é refém da falta de informações em processos operacionais tecnicamente simples, como as distâncias viárias exatas entre os clientes e/ou depósitos. Por isso, a importância de encontrar alternativas mais eficientes e econômicas que o SIG, como proposto neste trabalho.

## 2.2 Iterad Local Search

*Iterated Local Search* (ILS) é uma metaheurística introduzida por Baxter (1981) e pode ser descrita resumidamente em quatro procedimentos. O procedimento *GerarSoluçãoInicial* retorna uma solução inicial  $s_0$ . O procedimento *BuscaLocal* que, a partir de uma solução inicial que pode ser  $s_0$ , obtém uma solução ótima local  $s^*$ . O procedimento *Perturbação* gera uma nova solução  $s'$  para o procedimento *BuscaLocal*. O procedimento *BuscaLocal* é repetido encontrando soluções  $s'^*$  enquanto um critério de término não é atingido. O procedimento *CritérioAceitação* decide de qual solução  $s^*$  a busca é continuada. A ILS depende também do histórico da busca em relação aos procedimentos *Perturbação* e *CritérioAceitação*, indicado por *histórico* (LOURENÇO *et al.*, 2002). O algoritmo da ILS é apresentado na Figura 1 a seguir.

```
procedimento ILS
   $s_0 = \text{GerarSoluçãoInicial}$ 
   $s^* = \text{BuscaLocal}(s_0)$ 
  repete
     $s' = \text{Perturbação}(s^*, \text{histórico})$ 
     $s'^* = \text{BuscaLocal}(s')$ 
     $s^* = \text{CritérioAceitação}(s^*, s'^*, \text{histórico})$ 
  até condição de encerramento ser atingido
fim
```

Figura 1- Pseudocódigo da ILS.

Analisando a estrutura da ILS, nota-se que a efetividade dessa metaheurística depende dos seus quatro componentes. O procedimento *GerarSoluçãoInicial* depende do tipo de problema que está sendo resolvido. Normalmente é utilizada uma heurística de conhecida eficiência para o problema em



questão, pois reduz o tempo de execução computacional até encontrar boas soluções. Por exemplo, para o problema *flowshop*, pode-se usar a heurística NEH. Outra opção é usar uma solução inicial completamente aleatória, implementação mais rápida, porém menos eficiente. O procedimento *BuscaLocal* é o componente mais importante da ILS porque influencia fortemente a qualidade da solução final e o resultado do tempo computacional e consiste em realizar uma busca em toda uma vizinhança específica da solução de partida. O procedimento *Perturbação* tem o objetivo de fazer a ILS escapar de ótimos locais e, ao mesmo tempo, evitar a desvantagem de *restart* aleatórios e consiste em realizar  $k$  movimentos na solução atual aleatoriamente, seguindo uma distribuição de probabilidade. Os procedimentos *CritérioAceitação* e *Perturbação* influenciam fortemente o caminho da busca no espaço de soluções do problema e são usados para controlar o balanço entre intensificação e diversificação. Há vários critérios de aceitação, sendo um deles sempre aceitar a solução localmente ótima.

O problema de implementar uma ILS é escolher que heurística será utilizada no procedimento *GerarSoluçãoInicial*, qual tipo de estrutura de vizinhança será utilizada no procedimento *BuscaLocal*, o tipo e o número de alterações na solução atual no procedimento *Perturbação* e qual será a regra utilizada no procedimento *CritérioAceitação*. Esses componentes devem ser escolhidos de forma a obter as melhores soluções com o menor custo computacional. Por causa da relação entre esses quatro componentes ser não-linear, encontrar essa melhor combinação é uma tarefa difícil.

### 3. RESULTADOS E DISCUSSÃO

#### 3.1 Construção da Matriz de Tempos

O PRV abordado nesse trabalho foi do tipo assimétrico, ou seja, os tempos de deslocamento entre dois pontos são diferentes dependendo do sentido, isso reflete melhor o mundo real, pois ruas de mão única geram caminhos diferentes de ida e volta para os mesmos pontos.

A matriz de tempo, que é um dos fatores mais importantes para o sucesso dos roteirizadores, foi construída usando a API do GoogleMaps. Pode-se trabalhar em uma página html usando-se a API do GoogleMaps gratuitamente com até 25 pontos simultaneamente. É possível trabalhar com mais pontos executando os procedimentos de forma sequencial. Foi usada a função *google.maps.DistanceMatrixService()*, da API do GoogleMaps, para calcular as distâncias de um ponto para um conjunto de outros 24 pontos, totalizando 25 pontos trabalhados ao mesmo tempo. Assim, para construir uma matriz de tempos com 25 pontos, é necessário executar esse procedimento 25 vezes.

#### 3.2 ILS para o PRV

Nesse trabalho, uma solução para o PRV foi representada por um vetor, onde cada elemento do vetor representa um cliente que precisa ser visitado. Na Figura 2 a seguir, foi representada uma solução para  $n = 9$  (número de clientes). Um segundo vetor representa os últimos pontos visitados em cada rota, como mostra a Figura 3.

1	5	6	7	4	9	8	3	2
---	---	---	---	---	---	---	---	---

Figura 2 - Representação de uma solução do PRV usando a ILS.

6	8	2
---	---	---

Figura 3 - Representação dos últimos pontos visitados de cada rota de uma solução.

De acordo com as Figuras 2 e 3, para esse exemplo, existem três rotas que são: rota 1: 0-5-6-0, rota 2: 0-7-4-9-8-0 e rota 3: 0-3-2-0.



Uma solução do PRV é avaliada pela função  $f$ , de acordo com a Equação (1) que calcula a distância total percorrida ou tempo total gasto por todas as rotas, respeitando as restrições impostas pelo modelo matemático apresentado na seção 2.1.

O procedimento *Solução Inicial* consistiu em construir um vetor de números inteiros de 1 até  $n$ , aleatoriamente, por meio de uma distribuição de probabilidades uniforme, pois é de mais rápida implementação mas que não comprometeu a qualidade das soluções.

O tipo de movimento utilizado neste trabalho foi o *shift*, que consiste em selecionar o cliente da posição  $p_1$  e o inserir na posição do cliente  $p_2$ , fazendo com que os clientes entre  $p_1$  até  $p_2$  (inclusive) sejam deslocados no sentido de  $p_1$ , com  $1 \leq p_1 \leq n$ ,  $1 \leq p_2 \leq n$ ,  $p_1 \neq p_2$ ,  $p_1 - 1 \neq p_2$ . O movimento  $v$  é definido pelo seguinte par de clientes:  $v = (p_1, p_2)$ . A vizinhança é composta pelo conjunto de movimentos  $Z = \{ (p_1, p_2) \mid p_1, p_2 \in \{1, 2, \dots, n\}, p_2 \notin \{p_1, p_1-1\} \}$  de cardinalidade  $(n - 1)^2$ , onde a condição  $p_2 \notin \{p_1, p_1-1\}$  evita a redundância de movimentos.

Para o procedimento *Perturbação*, foi utilizado o movimento *shift*. Por meio de alguns testes, foi verificado que  $k = 3$  obtinha as melhores soluções para o problema avaliado.

O procedimento *Busca Local* consistiu em executar os movimentos  $v$  de uma vizinhança  $Z$  e, no momento que uma solução melhor que a atual é encontrada, reiniciar a busca do ponto inicial. Se toda a vizinhança é pesquisada e nenhuma solução melhor que a atual é encontrada, significa que a solução atual é um ótimo local para a vizinhança  $Z$ , então a execução da *Busca Local* é encerrada.

O procedimento *Critério Aceitação* adotado pelo algoritmo ILS nesse trabalho foi aceitar a solução que seja um ótimo local.

O algoritmo ILS foi implementado em linguagem Java 6. Para avaliar o método proposto, foram escolhidos 25 pontos localizados no município de Fortaleza, Estado do Ceará, com suas respectivas latitudes e longitudes. A localização do depósito foi -3.793709,-38.56901 e as localizações dos 24 clientes a serem visitados foram apresentados na Figura 4. Na Figura 5, são mostrados usando a API do GoogleMaps, com destaque para o depósito marcado com o número 1.

-3.735899,-38.541867	-3.811223,-38.510388	-3.751465,-38.496782	-3.807883,-38.46655
-3.721146,-38.546038	-3.838307,-38.521696	-3.818738,-38.563349	-3.732109,-38.528799
-3.727248,-38.582271	-3.821928,-38.619479	-3.820793,-38.630337	-3.725407,-38.530387
-3.764012,-38.590704	-3.723747,-38.516652	-3.755769,-38.583623	-3.739475,-38.540794
-3.712152,-38.546416	-3.729935,-38.5075	-3.82439,-38.481463	-3.748682,-38.549098
-3.813921,-38.529035	-3.731295,-38.496782	-3.82681,-38.466722	-3.783796,-38.546823

Figura 4 - Latitudes e longitudes dos 24 pontos a serem visitados.

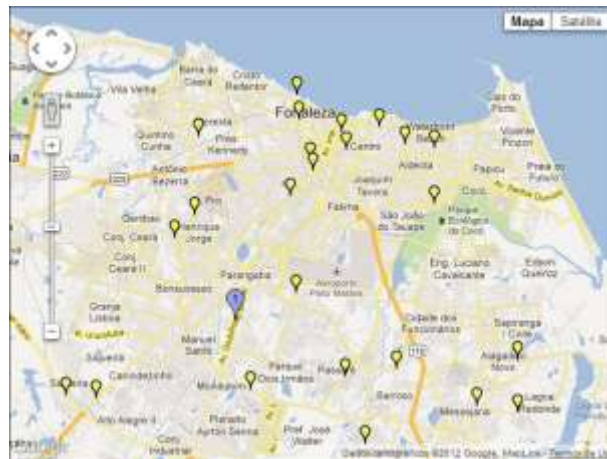


Figura 5 - Representação no mapa usando a API do GoogleMaps da localização dos 25 pontos da matriz de tempos.



Outro parâmetro para o PRV é a demanda dos clientes. Neste trabalho, a demanda foi representada por um percentual da capacidade do veículo utilizado. As demandas foram geradas aleatoriamente e apresentadas na Figura 6 a seguir.

0,11	0,19	0,14	0,20	0,17	0,11
0,17	0,13	0,13	0,12	0,18	0,14
0,16	0,18	0,10	0,15	0,15	0,19
0,20	0,15	0,19	0,12	0,13	0,13

Figura 6 - Matriz de demandas para os 24 clientes a serem visitados.

A matriz de tempos não foi inserida no trabalho porque ocuparia muito espaço.

A condição de encerramento foi a execução de 1.000 procedimentos *BuscaLocal*.

O tempo total de todas as rotas da solução inicial foi de 681 minutos e, ao final da execução da ILS, o tempo total de todas as rotas foi de 337 minutos, ou seja, uma melhoria de 50,51%. O tempo de execução computacional consumido foi de 2,57 segundos, um tempo reduzido, o que viabiliza a utilização desse método em aplicações reais.

A solução final do exemplo apresentou cinco rotas: rota 1: 0- 24- 10- 11- 12- 13- 23- 0; rota 2: 0- 21- 5- 2- 3- 16- 4- 0; rota 3: 0- 9- 15- 14- 8- 6- 0; rota 4: 0- 7- 17- 18- 19- 22- 0; e rota 5: 0- 20- 1- 0.

Para ilustrar como as rotas podem ser apresentadas pela API do GoogleMaps, a rota 1 foi desenhada usando a função *directionsService.route()*. Na Figura 7, é apresentada essa rota, onde *H* representa o depósito.



Figura 7 - Representação gráfica da rota 1 da resolução do exemplo usando a API do GoogleMaps.

#### 4. CONCLUSÕES

Os estudos práticos sobre o PRV sempre enfocam a importância das matrizes de distância ou tempo e a dificuldade operacional ou custo financeiro em obtê-las.

Este trabalho conseguiu mostrar que é possível construir um roteirizador de qualidade sem incorrer em grandes investimentos na aquisição de um SIG. Isso foi possível utilizando-se a tecnologia da API do GoogleMaps na sua versão gratuita. Essa API tanto executa a tarefa de escolher os trechos das ruas que resultam na menor distância entre dois pontos, como também é excelente para a visualização gráfica dos roteiros.



A resolução do PRV, usando a metaheurística ILS para um problema com 24 clientes, consumiu aproximadamente 2,57 segundos e demonstrou ser completamente viável esse método para aplicações práticas.

Um trabalho futuro será incorporar a metaheurística ILS diretamente na página html por meio da tecnologia JSF (Java Server Faces), que é uma tecnologia que permite que procedimentos sejam executados em um servidor e apresentados para os usuários por meio de uma página html.

## AGRADECIMENTOS

Os autores agradecem ao apoio financeiro do IFCE Campus Quixadá em forma de bolsa de iniciação científica.

## REFERÊNCIAS

ARENALES, M.; ARMENTANO, V.; MORABITO, R.; YANASSE, H. **Pesquisa operacional**. 1. ed. São Paulo: Elsevier, 2007.

BAXTER, J. Local optima avoidance in depot location, *The Journal of the Operational Research Society*, v. 32, n. 9, p. 815–819, 1981.

BODIN, L.D. Twenty years of routing and scheduling. *Operations Research*, v. 38, n. 4, p. 571-579, 1990.

CUNHA, C.B. Aspectos práticos da aplicação de modelos de roteirização de veículos a problemas reais. *Revista Transportes da ANPET – Associação Nacional de Pesquisa e Ensino em Transportes*, v.8, n.2, p.51-74, 2000.

ENOMOTO, L. M.; LIMA, R. S. Análise da distribuição física e roteirização em um atacadista. *Produção*, v. 17, n. 1, p. 094-108, 2007.

GOOGLE MAPS API. **The Google Maps Javascript API V3**. Disponível em: <<https://developers.google.com/maps/documentation/javascript/?hl=pt-BR>>. Acesso em: 28 mai. 2012.

LAPORTE, G.; M. GENDREAU; J.Y. POTVIN e F. SEMET (2000) Classical and modern heuristics for the vehicle routing problem. *International Transactions in Operational Research*, v.7, n 4/5, p. 285-300, 2000.

LOURENÇO, H. R.; MARTIN, O. & STÜTZLE, T. Iterated Local Search. In F. Glover and G. Kochenberger, editors, **Handbook of Metaheuristics**, International Series in Operations Research & Management Science, v.57, p. 321-353, Kluwer Academic Publishers, Norwell, MA, 2002.

MELO, A. C. da S.; FILHO, V. J. M. F. Sistemas de Roteirização e Programação de Veículos. *Pesquisa Operacional*, v. 21, n. 2, p. 223-232, 2001.

STÜTZLE, T. *Applying iterated local search to the permutation flow shop problem*. Technical report, AIDA-98-04, TU Darmstadt, FG Intellektik, 1998.

VALOR ONLINE. **Brasil gasta 10,6% do PIB com logística, mostra estudo**. Disponível em: <<http://g1.globo.com/economia/noticia/2011/09/brasil-gasta-106-do-pib-com-logistica-mostra-estudo.html>>. Acesso em: 27 mai. 2012.