

## S.D.R - Sistema de Dimensionamento de Rotas

Diego Bittencourt de Oliveira<sup>1</sup>, Daniela Scherer dos Santos<sup>1</sup>, Daniel Biasoli<sup>1</sup>,  
Márcio Daniel Puntel<sup>1</sup>, Rodrigo dos Santos Keller

<sup>1</sup>Sistemas de Informação – Universidade Luterana do Brasil (ULBRA)  
Caixa Postal s/nº – 96.503-000 – Cachoeira do Sul – RS – Brazil

dbo.oliveira@gmail.com

{daniela.santos37,daniel,marcio.puntel,rodrigo.keller}@ulbra.edu.br

**Abstract.** *This work aims to present a tool (SDR - Dimensioning System of Routes) developed to generate optimized routes for deliveries through the application of different routing algorithms. SDR comprises a desktop tool and a tool for mobile devices that helps the user to correctly follow the route generated by the algorithm used. Results obtained during validation of the system were satisfactory since the generated routes have showed very similar values if compared to those obtained after the effective execution of the route.*

**Resumo.** *Este trabalho tem por objetivo apresentar a ferramenta S.D.R. (Sistema de Dimensionamento de Rotas), desenvolvida para gerar rotas otimizadas de entregas por meio da aplicação de diferentes algoritmos de busca. O S.D.R. é composto por uma ferramenta desktop e outra para dispositivos móveis que auxilia o usuário a seguir a rota gerada pelo algoritmo utilizado. Resultados obtidos durante a validação mostraram-se satisfatórios uma vez que as rotas geradas resultaram valores bem similares se comparados aos obtidos após a execução efetiva do percurso.*

### 1. Introdução

A logística está se tornando um elemento fundamental para o crescimento de diversas empresas, sejam estas grandes indústrias ou pequenas lojas virtuais. Ambas necessitam entregar seus produtos, e o prazo de entrega é fundamental para a concretização da própria venda.

Segundo [Goldbarg et al. 2006] sistemas logísticos de roteamento podem ser considerados um conjunto de meios que possuem o objetivo de atender as demandas de entregas localizadas nos arcos ou vértices de redes de transportes. Onde os arcos e vértices representam basicamente o emaranhado de ruas que a cidade possui.

Atividades relacionadas ao transporte e distribuição de mercadorias são importantes para o setor logístico, pois os custos de transporte representam atualmente uma parcela significativa do preço praticado. Assim, a busca pela eficiência nesses serviços é um nicho a ser constantemente explorado. Esta eficiência pode ser obtida por meio de um adequado sistema de dimensionamento de rotas de entregas.

Desta forma, o objetivo desta pesquisa é apresentar o desenvolvimento da ferramenta S.D.R, um sistema de dimensionamento de rotas geradas a partir da aplicação dos algoritmos de busca A\*, Dijkstra e Prim.

O S.D.R. é composto por duas ferramentas: uma *desktop* e outra para dispositivos móveis. A ferramenta *desktop* fornece funcionalidades administrativas para controle da aplicação, como cadastros das entregas e relatórios para controlar o andamento das mesmas. A ferramenta para dispositivos móveis auxilia o usuário

entregador a seguir com exatidão a rota que foi gerada, além disso, coleta informações estatísticas que são utilizadas para avaliar o desempenho do algoritmo de roteamento utilizado na execução da rota.

O presente artigo encontra-se organizado da seguinte maneira: na Seção 2 apresentam-se os algoritmos de busca Dijkstra, Prim e A\*; na Seção 3 encontra-se descrita a metodologia utilizada para o desenvolvimento do trabalho; a Seção 4 descreve os resultados obtidos; por fim, na Seção 5, estão relatadas as considerações finais e trabalhos futuros.

## 2. Algoritmos de Busca

Para a construção das rotas de entregas utilizou-se os algoritmos A\*, Dijkstra e Prim por se tratarem de algoritmos amplamente usados em problemas de roteamento. Os algoritmos Dijkstra e A\* foram usados em [Lux e Furtado 2001] para determinação da melhor rota entre um centro de atendimento a situações de emergência e o local de sua ocorrência. Uma implementação modificada do algoritmo de Prim foi usada em [Kheirkhahzadeh e Barforoush 2009] para o problema de roteamento de veículos.

### 2.1. Algoritmo de Dijkstra

Segundo [Goldbarg et al. 2006] o algoritmo de Dijkstra, detalhado na Figura 1, é o mais famoso dos algoritmos para cálculo de caminho de custo mínimo entre vértices de um grafo. A Figura 1 apresenta o procedimento para um grafo direcionado  $G=(V, A)$ , onde  $V=\{1, 2, \dots, n\}$  e o vértice 1 é a origem,  $S$  inicialmente armazena somente a origem, posteriormente vai possuindo todos os nós do grafo a medida que forem encontrados os menores caminhos do vértice de origem aos demais nós. A matriz de custos é identificada por  $C$ , onde  $C[i,j]$  é o custo de ir do vértice  $i$  ao vértice  $j$  no arco  $i \rightarrow j$ . Se o arco  $i \rightarrow j$  não existir então,  $C[i,j]$  assumirá  $\infty$ . A cada iteração  $D[i]$  recebe o valor do caminho mais curto da origem até o vértice  $i$ .

Uma vez definido o vértice de partida, o algoritmo de Dijkstra possui a capacidade de calcular o caminho de menor custo entre esse e os demais vértices do grafo [Goldbarg et al. 2006].

```
Figura 1: Algoritmo de Dijkstra
1 S ← {1};
2 for i ← 2 to n do
3   D[i] ← ∞; // D é inicializado
4 end
5 for i ← 1 to n - 1 do
6   Escolha um vértice w em V - S tal que D[w] seja mínimo;
7   Acrescente w a S;
8   foreach vértice v em V - S do
9     D[v] ← min(D[v], D[w] + C[w,v]);
10  end
11 end
```

Figura 1. Algoritmo de Dijkstra

### 2.2. Algoritmo Prim

O algoritmo de Prim, segundo [Goldbarg et al. 2006], é um algoritmo de árvore geradora mínima, ou seja, determina uma rota entre um ponto de partida e todos os demais vértices do grafo. A rota é construída de forma que a soma dos valores das arestas seja a menor possível.

A Figura 2 detalha o algoritmo de Prim, sendo  $G=(V, A)$  um grafo direcionado,  $V=\{1, 2, \dots, n\}$  e o vértice origem é escolhido aleatoriamente e armazenado em  $T$ . A matriz de custos é identificada por  $D$ , onde  $D_{ij}$  é o custo de ir do vértice  $i$  ao vértice  $j$  no arco  $i @ j$ . Se o arco  $i @ j$  não existir então,  $D_{ij}$  assumirá  $\infty$  (um valor extremamente grande). A cada iteração  $S$  recebe o valor do caminho mais curto da origem até o vértice  $T$ .

```

Figura 2: Algoritmo de Prim
1 Considerando os conjuntos  $S, T \in V$ , onde  $S \subseteq A, T \subseteq N$  e  $V \subseteq N$ ;
2 Ler  $G = (N, A)$  e  $D = |D_{ij}|$ ;
3 Escolher aleatoriamente um vértice  $i \in N$ ;
4  $T \leftarrow \{i\}$ ;
5  $V \leftarrow N \setminus T$ ;
6 while  $T \neq N$  do
7   foreach  $j \in T$  do
8     Encontrar a menor aresta  $(j, k) \in A$  tal que  $j \in T, k \in V$ ;
9      $T \leftarrow T \cup \{k\}$ ;
10     $V \leftarrow V \setminus \{k\}$ ;
11     $S \leftarrow S \cup \{j, k\}$ ;
12  end
13 end
14 Escrever  $S$ ; // arestas do menor geradora mínima
    
```

Figura 2. Algoritmo Prim

### 2.3. Algoritmo A\*

Segundo [Russell e Norvig 2004], o A\* é um algoritmo de definição de rotas de alta eficiência, muito aplicado no desenvolvimento de jogos de computador. O algoritmo A\* (Figura 3) implementa a estimativa do custo do nó atual ao nó destino, sendo vértice  $I$  a origem e a matriz de custos é identificada por  $C$ , onde  $C[w, v]$  é o custo de ir do vértice  $w$  ao vértice  $v$  no arco  $w @ v$ . Observando que a cada vértice  $w$  conectado na origem  $S$  será selecionado o vértice mais próximo do destino  $d$ , o armazenando em  $P$ . Se o vértice  $w$  tiver um custo maior que o custo de outro vértice visitado, então os vértices conectados a este não serão visitados. A cada iteração  $S$  recebe o valor do caminho mais curto da origem  $i$  até que  $S$  contenha o vértice  $d$ .

```

Figura 3: Algoritmo A*
1  $S \leftarrow \{i\}$ ;
2 for  $j \in V$  to  $d$  do
3    $D[j] \leftarrow \infty$ ;
4 end
5 while  $d$  não estiver em  $S$  do
6   foreach vértice  $w$  em  $S$  do
7     Escolher em  $V - S$  o nó conectado a  $w$  tal que  $D[w, w] + h^*$  seja
      mínimo;
8      $P \leftarrow w$ ;
9      $D[w] \leftarrow D[w] + C[w, v]$ ;
10  end
11  Escolher um vértice  $v$  em  $P$  tal que  $D[v] + h^*$  seja mínimo;
12   $S \leftarrow S \cup \{v\}$ ;
13   $P \leftarrow P \setminus \{v\}$ ;
14 end
    
```

Figura 3. Algoritmo A\*

## 3. Metodologia

O S.D.R. foi desenvolvido usando-se o conceito de programação em três camadas, tornando simplificada a alteração e a manutenção do código, pois, segundo [Freeman et al. 2009], um *software* que possui a característica de ser facilmente alterado ou receber manutenção de forma simplificada, terá uma vida longa no mercado. Assim, o acesso ao

banco de dados, as regras de negócio e a interface com o usuário encontram-se implementados em pacotes individuais utilizando-se a Linguagem C#<sup>3</sup>.

Para o desenvolvimento do aplicativo móvel, foi necessária a implementação de *webservices* em linguagem C# com base nas práticas que estão sendo adotadas na área de desenvolvimento em plataformas móveis, onde o sistema operacional *Android*<sup>4</sup> não provê acesso direto a um banco de dados via TCP/IP.

Para a geração de uma rota de entregas pelos algoritmos A\*, Prim e Dijkstra, primeiramente é necessário a definição dos pontos inicial e final de entrega e a seguir é preciso criar o grafo com informações sobre as entregas. Estas tarefas encontram-se detalhadas a seguir.

### 3.1. Definição dos Pontos de Entrega Inicial e Final

As coordenadas geográficas de cada entrega e da empresa responsável pelas entregas são obtidas utilizando-se a API do *Google Maps*<sup>5</sup>, através do endereço cadastrado previamente no sistema.

Para a definição dos pontos inicial e final de entrega é necessário calcular a distância entre cada ponto de entrega e a empresa de entregas e escolher dois pontos que possuam a menor distância em relação a empresa de entregas e um número de CEP diferentes, ou seja, estejam situados em ruas diferentes. Entre os dois pontos escolhidos, o ponto inicial será aquele que possuir a menor distância com relação a empresa.

### 3.2. Criação do Grafo com as Entregas das Rotas

Os pontos de entregas são armazenados em uma estrutura de dados do tipo grafo. A Figura 4 ilustra um exemplo do grafo necessário para que a rota seja gerada, onde os vértices representam os pontos de entrega e suas arestas representam as distâncias entre cada um deles. O vértice *empresa* representa o ponto de saída, o qual somente necessita possuir as distâncias entre os pontos inicial e final (*P1* e *P6* respectivamente). *P1* e *P6* também não possuem a necessidade de possuir ligações diretas entre ambos.

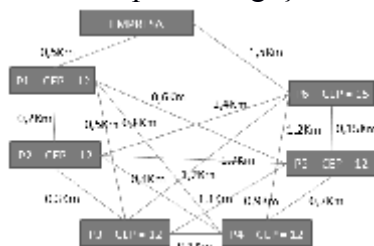


Figura 4. Ligações necessárias para o roteamento

Para a aquisição das distâncias que cada ligação possui, são necessários acessos ao *webservices* do *Google Maps* que possui um limite de 2500 consultas diárias, o qual é rapidamente excedido. Para que isso seja evitado, utiliza-se a Equação 1, onde *D* é a distância entre as coordenadas, *LatI* e *LonI* representam latitude e longitude iniciais, e *LatF* e *LonF* representam latitude e longitude finais  $Z = \cos((90 - LatI) * p/180)$ ,  $Y = \cos((90 - LatF) * p/180)$ ,  $W = \sin((90 - LatI) * p/180)$ ,  $X = \sin((90 - LatF) * p/180)$ ,  $V = (360 + LonI) * p/180$  e  $U = (360 + LonF) * p/180$ .

$$D = (\arccos(Z * Y + W * X * \cos(\text{abs}(V - U))) * 6371.004) * 1000 \quad (1)$$

3 <http://msdn.microsoft.com/pt-br/vstudio/hh341490.aspx>

4 <http://www.android.com/>

5 <https://developers.google.com/maps/documentation/javascript/reference?hl=pt-br>

A Equação 1 é baseada na lei dos cossenos da trigonometria esférica e a distância segue a linha geodésica, que é a curva de menor comprimento unindo dois pontos. Como o sistema de coordenadas terrestre é dividido em hemisférios, com os ângulos entre 0 e 90 graus e 0 e 180 graus, faz-se necessário ajustes das coordenadas inicial e final para estes ângulos, os quais são calculados de acordo com  $Z$ ,  $Y$ ,  $W$ ,  $X$ ,  $V$  e  $U$ . A Equação 1 calcula Arco Cosseno utilizando estes ângulos ajustados e multiplicando os resultados pelo raio médio da terra que é igual a 6371,004 km, por fim o resultado é multiplicado por 1000 para que este seja expresso em metros.

Para que a diminuição das consultas ao *Google Maps* ocorra, efetua-se a eliminação de algumas arestas. Desta forma, entre vértices (entregas) que possuem o mesmo CEP são mantidas apenas três arestas e as demais são eliminadas. A condição para que uma aresta seja eliminada considera se existem outras 3 arestas com distâncias menores entre os vértices de mesmo CEP analisados pela rotina de eliminação de arestas.

Após a realização dos procedimentos descritos acima, são buscadas as distâncias reais das arestas remanescentes via *Google Maps*, com isso o grafo se encontra apto a receber a execução do algoritmo de roteamento escolhido para calcular a rota em questão.

## 4. Resultados

A presente Seção descreve a ferramenta desenvolvida bem como os resultados obtidos com relação a sua validação em dois cenários distintos: um ambiente de simulação de entregas e um ambiente de entregas reais.

### 4.1. Ferramenta S.D.R.

O objetivo principal da ferramenta S.D.R. é apresentar ao usuário uma rota de entrega otimizada, gerada a partir dos algoritmos A\*, Prim e Dijkstra. Além disso, a ferramenta pode ser útil para comparar os resultados fornecidos por esses algoritmos.

O S.D.R. é composto por duas ferramentas de *software*, uma para a plataforma *desktop* e outra para dispositivos móveis.

A ferramenta *Desktop* concentra todas as funcionalidades de controle de acessos ao sistema. Isso se deve ao fato de que empresas de entregas muitas vezes possuem um número grande de colaboradores que desempenham diversos processos para que as entregas sejam executadas. Assim, o controle de acesso limita o uso da ferramentas apenas por colaboradores que possuam um cadastro no sistema. Além disso, pode-se personalizar o menu de opções de cada usuário a fim de permitir ou não o acesso a uma determinada funcionalidade.

O endereço de uma entrega é uma informação de extrema importância e pode comprometer a correta geração de uma rota de entregas. Por isso, o S.D.R. possui um aprimorado conjunto de cadastros que visam disponibilizar informações sobre os endereços das entregas. Desta forma, são concentradas informações pré cadastradas de todos os endereços de CEP, Bairros, Cidades e Estados conforme a área em que a empresa de entregas atua. Com base nestes pré cadastros o endereço de uma entrega somente é validado e salvo se o mesmo estiver pré cadastrado na base de dados da ferramenta. Na ocorrência de um endereço que não conste nos cadastros do S.D.R. o mesmo deve ser verificado e cadastrado.

Os cadastros fundamentais da ferramenta *Desktop* são: cadastro de entregas, onde são cadastradas as entregas com seus respectivos endereços e cadastro de rotas, onde as entregas cadastradas são selecionadas manualmente para compor a rota. Além disso, a ferramenta *Desktop* possui um relatório que fornece informações estatísticas das rotas calculadas e também após a sua execução.

A ferramenta móvel do S.D.R. possui basicamente três telas de interação com o usuário. A primeira apresenta o *login* onde o usuário acessa o sistema. Na segunda tela são listadas todas as rotas geradas pelos algoritmos A\*, Prim e Dijkstra que estão cadastradas para o usuário que acessou a ferramenta móvel. Nessa tela, o usuário deve selecionar a rota que vai ser executada. A terceira tela auxilia o usuário a realizar as entregas conforme a rota escolhida. Para isso, o mesmo possui uma visualização gráfica da rota (Figura 5) e uma visualização das entregas em forma de lista. Estas funcionalidades auxiliam o usuário para que o mesmo possa analisar o andamento das suas entregas (quantas foram realizadas e quantas ainda restam).

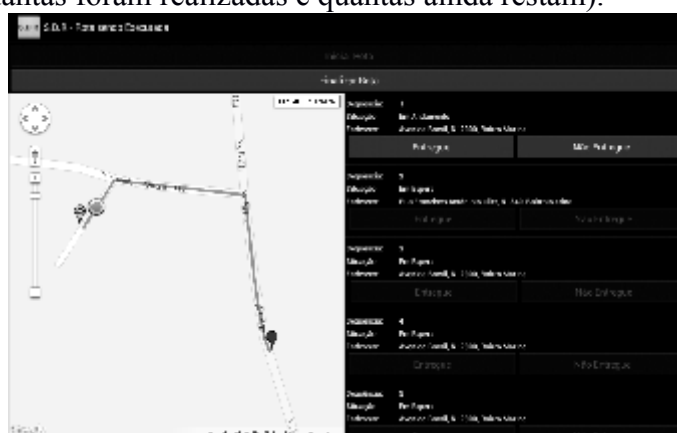


Figura 5. Tela de execução de rotas

Na visualização gráfica, o usuário também pode visualizar possíveis desvios existentes para utilizar em uma ocorrência de uma rua bloqueada, por exemplo.

## 4.2. Validação do S.D.R.

Para verificar o correto funcionamento da ferramenta desenvolvida e dos algoritmos implementados para a geração de rotas utilizou-se dois cenários distintos: um cenário real de entregas e outro gerado por simulação, ambos descritos a seguir.

### 4.2.1. Testes em Ambiente de Simulação de Entregas

Testes simulados foram realizados com duas rotas fictícias, uma com 50 e a outra com 100 pontos de entregas. Os pontos foram escolhidos aleatoriamente seguindo-se as seguintes restrições: deveriam pertencer à área urbana da cidade de Cachoeira do Sul - RS e o limite máximo de pontos com um mesmo CEP deveria ser 4.

As Tabelas 1 e 2 apresentam os valores de distâncias e tempo calculados pelos algoritmos de roteamento em uma rota fictícia com 50 e 100 pontos de entregas, respectivamente. Pode-se notar que o algoritmo A\* apresentou um desempenho melhor que os demais algoritmos. Para calcular o tempo médio das entregas considerou-se que o percurso seria realizado caminhando a uma velocidade de 2,5 quilômetros por hora. Esta velocidade é apenas uma estimativa, baseada na distância média que uma pessoa normal caminha em uma hora de percurso.

Tabela 1. Rota fictícia com 50 pontos de entrega

Algoritmos	Distância Total	Distância Média	Tempo Total	Tempo Médio
Dijkstra	11,629 Km	0,232 Km	04:39:05	00:05:34
Prim	11,929 Km	0,236 Km	04:43:53	00:05:39
A*	10,237 Km	0,204 Km	04:05:41	00:04:53

**Tabela 2. Rota fictícia com 100 pontos de entrega**

Algoritmos	Distância Total	Distância Média	Tempo Total	Tempo Médio
Dijkstra	16,947 Km	0,169 Km	06:46:43	00:04:03
Prim	17,234 Km	0,172 Km	06:56:36	00:04:07
A*	15,995 Km	0,160 Km	06:23:52	00:03:50

Visando um teste com maior eficácia das rotas geradas na Tabela 1, as rotas foram efetivamente percorridas. Para simular o tempo levado pelo entregador para efetuar a entrega no respectivo endereço, aguardou-se 15 segundos em cada ponto de entrega. A Tabela 3 ilustra os resultados obtidos após a execução de todo o percurso.

**Tabela 3. Rota fictícia com 50 pontos de entrega (percorrida)**

Algoritmos	Distância Total	Distância Média	Tempo Total	Tempo Médio
Dijkstra	12,242 Km	0,244 Km	05:06:06	00:06:06
Prim	12,523 Km	0,246 Km	05:08:04	00:06:09
A*	10,869 Km	0,217 Km	04:44:33	00:05:40

Os testes realizados em um cenário de simulação mostram que a rota gerada resulta em valores similares se comparados aos obtidos após a execução efetiva do percurso.

#### 4.2.2. Testes em Ambiente Real de Entregas

Os testes em ambiente real foram realizados em uma empresa do ramo de supermercados localizada na região central da cidade de Cachoeira do Sul. A empresa possui 14 funcionários, cerca de 5,4 mil clientes e uma receita média mensal de 430 mil reais. Realiza em média 530 entregas por mês, sendo que 80% é feita gratuitamente.

Foram efetuadas 16 entregas em duas rotas (uma com 6 e outra com 10 entregas). Por possuir poucos pontos de entregas, os resultados obtidos para a rota de 6 pontos foram idênticos para os três algoritmos: Distância Total = 7,451 Km, Distância Média = 1,241 Km, Tempo Total = 00:11:10 e Tempo Médio = 00:01:51.

A Tabela 4 apresenta os resultados obtidos para a rota com 10 pontos, onde o algoritmo A\* apresentou uma rota mais otimizada, a qual foi escolhida para ser executada. Após a execução da rota obteve-se uma Distância Total de 8,022 Km, variando apenas 154 metros com relação ao valor calculado pelo algoritmo, e um Tempo Total de 00:46:34.

**Tabela 4. Rota real com 10 pontos de entrega**

Algoritmos	Distância Total	Distância Média	Tempo Total	Tempo Médio
Dijkstra	8,214 Km	0,821 Km	00:12:19	00:01:13
Prim	8,214 Km	0,821 Km	00:12:19	00:01:13
A*	7,868 Km	0,786 Km	00:11:48	00:01:09

### 5. Considerações Finais

O presente artigo apresentou a ferramenta S.D.R (Sistema de Dimensionamento de Rotas), um sistema cuja principal função é a geração de rotas otimizadas de entregas para empresas da área logística. O S.D.R. é composto por duas interfaces: uma *desktop* que possui funcionalidades administrativas para controle da aplicação, cadastros de entregas e relatórios, e uma interface para dispositivos móveis cujo objetivo é auxiliar o usuário entregador a seguir a rota gerada pela ferramenta. O S.D.R. emprega os

algoritmos A\*, Dijkstra e Prim para gerar as rotas a partir de um conjunto de entregas. Uma vez geradas, o usuário escolhe a rota para ser efetivamente executada pela empresa (por exemplo, aquela cuja distância a ser percorrida é menor).

A validação do S.D.R. aconteceu em dois cenários distintos: um ambiente simulado de entregas e também um ambiente empresarial real. Nos dois ambientes a ferramenta apresentou resultados satisfatórios uma vez que as rotas geradas resultaram valores bem similares se comparados aos obtidos após a execução efetiva do percurso.

A ferramenta S.D.R., embora não tenha esse objetivo, se mostrou útil como meio de estudo de desempenho de algoritmos de busca, uma vez que, através dela, pode-se analisar as rotas geradas como resultado da aplicação desses algoritmos.

Como trabalhos futuros pretende-se aprimorar a ferramenta incluindo a funcionalidade de trabalhar com rotas dinâmicas, ou seja, permitir que a rota seja alterada durante a sua execução mediante o surgimento de novas informações, por exemplo devido a chegada de uma nova solicitação de um cliente ou ao cancelamento de pedidos. Aplicações para este tipo de problema podem ser encontrados em serviços do tipo coleta e entrega.

## Referências

- Freeman, E., Robson, E., Bates, B., e Sierra, K. (2009). *Head First Design Patterns*. O'Reilly Media, 2ª edition.
- Goldberg, M. C., Pacca, H., e Luna, L. (2006). *Otimização Combinatória e Programação Linear*. Campus, 2ª edition.
- Kheirhahzadeh, M. e Barforoush, A. A. (2009). A hybrid algorithm for the vehicle routing problem. *In Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, pages 1791-1798. IEEE.
- Lux, B. e Furtado, J. C. (2001). Sistema de otimização de rotas com suporte de software gerenciador de informações geográfica. *Anais do Encontro Nacional de Engenharia da Produção, 2001*.
- Russell, S. e Norvig, P. (2004). *Inteligência Artificial*. Elsevier, 1ª edition.