

# Um Algoritmo Heurístico Para O Problema De Roteamento De Veículos

Fernanda Buriol<sup>1</sup>, Angela Mazzonetto<sup>2</sup>, Clicerres Mack Dal Bianco<sup>3</sup>

<sup>1</sup>Aluna do Curso de Ciência da Computação –Universidade Regional Integrada(URI)  
Caixa Postal 709 – 98.400-000 – Frederico Westphalen – RS – Brasil

<sup>2</sup>Aluna do Curso de Ciência da Computação –Universidade Regional Integrada(URI)  
Caixa Postal 709 – 98.400-000 – Frederico Westphalen – RS – Brasil

<sup>3</sup>Professora do Curso de Ciência da Computação –Universidade Regional Integrada(URI)  
Caixa Postal 709 – 98.400-000 – Frederico Westphalen – RS – Brasil.

inf18558@uri.edu.br, inf16753@uri.edu.br, clicerres@uri.edu.br

**Abstract.** *For companies working with a fleet of vehicles, reduce spending on unnecessary distances becomes very advantageous. Thus, this article shows the development of an algorithm that seeks a solution for the Vehicle Routing Problem, using technical Constructive heuristics.*

**Resumo.** *Para empresas que trabalham com uma frota de veículos, diminuir o gasto com distâncias desnecessárias torna-se muito vantajoso. Com isso, esse artigo mostra o desenvolvimento de um algoritmo que busca uma solução para o Problema de Roteamento de Veículos, através de técnica Heurística Construtiva.*

## 1. Introdução

É através de uma frota de veículos, que muitas empresas entregam e buscam produtos em clientes. Com isso muitas vezes elas se deparam com o problema do roteamento de veículos. Esse problema consiste em encontrar um percurso de distância mínima através da rede de rodovias a ser percorrida.

É possível encontrar uma solução exata através do uso de solução matemática, que procura através de todas as combinações de rotas possíveis, a que seja mais vantajosa. No entanto, dependendo da quantidade de possíveis rotas existentes, poderá haver um alto número de combinações, tornando o processo muitas vezes impossível. Por isso utiliza-se outra maneira de resolução para o problema, a utilização de algoritmos computacionais que usam as técnicas de heurísticas e meta-heurísticas.

Portanto, após estudos sobre o problema e suas diversas resoluções, foi desenvolvido um algoritmo empregando uma heurística Construtiva, objetivando alcançar posteriormente uma razoável solução para o Problema do Roteamento de Veículos, empregando outras heurísticas e Meta-heurísticas.

## 2. Heurística Construtiva

Segundo Liu e Shen (2005) Heurísticas Construtivas são métodos para construir uma solução passo a passo, onde em cada passo se adiciona componentes individuais: nós, arcos, variáveis, etc. As rotas começam vazias, e a cada iteração elas vão sendo

expandidas (construídas). A cada iteração, além da geração de novos elementos, é feita a eliminação dos elementos que não satisfaçam a um critério determinado para a permanência na lista, guardando-se o melhor conjunto de sequências (melhor rota) encontrado.

As principais heurísticas construtivas são os procedimentos de economia e inserção, destacando-se as heurísticas de Clark e Wright e de Mole e Jameson. A heurística das economias de Clark e Wright é o método utilizado pelo módulo ROUTER na resolução de problemas de roteamento. Ela baseia-se na noção de economias, que pode ser definida como o custo da combinação, ou união, de duas sub-rotas existentes (LIU & SHEN, 2005).

### 3. Solução utilizando da Heurística Construtiva

Para encontrar o menor caminho entre os lugares a serem percorridos o algoritmo desenvolvido baseia-se em uma heurística construtiva. Assim ele primeiramente irá ler os pontos a serem visitados e posteriormente fazer uma comparação entre cada um dos lugares a serem visitados. Ele procurará o ponto mais próximo de si para fazer a próxima visitação, sem selecionar maiores requisitos.

Primeiramente é preciso informar quantos locais serão visitados, para assim fazer-se o cálculo de quantas são as possibilidades existentes de caminhos, para poder ser feita posteriormente a leitura da distância de cada caminho.

As combinações crescem exponencialmente, tornando muitas vezes o cálculo bastante demorado. Por isso, que devem ser usadas outras técnicas futuramente para baixar o tempo computacional, pois uma solução meramente Construtiva apenas gerará uma solução inicial e não uma solução aceitável para o problema.

Na tabela 1, pode-se ver como é preciso dar a distância entre cada uns dos lugares com cada um dos outros lugares, para depois disso escolher o caminho que será seguido.

**Tabela 1. Leitura dos locais a serem visitados**

1	<code>cout &lt;&lt; "Informe distância de cada ponto de visita" &lt;&lt; endl;</code>
2	<code>for (int j=i+2; j&lt;=pot; j++) {</code>
3	<code>  c++; e=1+i;</code>
4	<code>  cout &lt;&lt; "Distância entre locais " &lt;&lt; i+1 ;</code>
5	<code>  cout &lt;&lt; " e local " &lt;&lt; j &lt;&lt; " ";</code>
6	<code>  cin &gt;&gt; dist[c]; // insere o valor da distância entre os pontos na variável dist</code>
7	<code>  pnt1[c]=e; pnt2[c]=j; // guarda os pontos de cada distancia entre pontos</code>
8	<code>  if (j==a){ i=i+1; j=i+1;} if (c==pot-1) {j=pot+1;} e=0; }</code>

Cada ponto se ligará com todos os outros pontos a serem visitados o que fará uma teia. Quanto mais lugares a serem visitados mais caminhos haverá. Assim, se há três lugares, serão três possibilidades de caminho, se forem quatro serão seis e assim por diante.

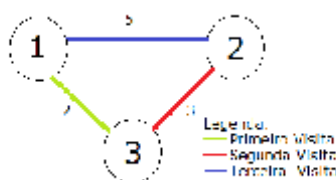
Após os locais a serem visitados forem lidos, o algoritmo calcula qual o caminho mais curto a ser seguido, seguindo o princípio do vizinho mais próximo a ele. O modo como essa escolha é feita pode ser acompanhada através do código da tabela 2.

**Tabela 2. Calculando percurso mais curto**

1	for (int j=0; j<pot; j++){
2	for (int i=0; i<pot; i++){
3	if ((pnt1[i]==pnt)    (pnt2[i]==pnt)) {
4	if (dist[i] !=0) { // o percurso mais curto entre local 1 e outro local
5	if (dist[i]<menorinicial){
6	menorinicial= dist[i]; d=i; //h=4; l=1 } } } }
7	if (pnt1[d]==pnt) pnt=pnt2[d];
8	else pnt =pnt1[d];
9	ordemvisitas[z]= menorinicial;
10	z++; menorinicial= 99999999;
11	dist[d]=0; //zera os lugares que já foram visitados

Assim, partindo de um ponto inicial que será o primeiro valor a ser lido, o algoritmo calculará qual o próximo local mais próximo a ele. O algoritmo passará para esse lugar e fará o mesmo processo procurando o próximo lugar com menor distância, até que todos os pontos forem visitados e possa retornar ao local inicial. Desta maneira, ao fim dos cálculos feitos pelo algoritmo, ele retornará como resposta a ordem a ser visitada para ser percorrido o caminho mais curto.

Como exemplo, foram informados três locais para visita, no caso utilizado, distância entre local 1 e local 2 igual a 5, distância entre local 1 e 3 igual a 2 e distância entre local 2 e 3 igual a 3, posteriormente a distância entre cada local. Ao fim é mostrada a ordem que deve ser visitada para percorrer o menor caminho, que no caso são as distâncias 2, 3 e 5, que se referem a ir do local 1 para o local 3 e depois do local 3 para o 2. Pode-se acompanhar na figura 2 a ordem que foi seguida, devido a distância de cada ponto a cada ponto.



**Figura 1. Grafo com ordem de visitação.**

E é também dessa forma que acontece com quantidades maiores de visitas, no entanto, quanto mais locais, mais demorada será a resposta do programa, chegando, dependendo das proporções, a não dar resposta em tempo hábil.

Nas próximas etapas mais técnicas heurísticas e meta-heurísticas serão testadas, buscando diminuir o tempo de resposta e melhorar o programa desenvolvido em todos

os quesitos possíveis. Será, a partir desse esboço inicial utilizado para refinar a resposta inicial, a Meta-heurística Busca Tabu, que segundo Glover e Laguna (1997) é uma meta-heurística que consiste em explorar o espaço de soluções de um problema evitando-se executar movimentos desnecessários. Para tal, são definidos movimentos proibidos (ou tabus).

#### **4. Conclusão**

A busca pela resolução de um problema é sempre válida e melhorar uma solução já existente também. O problema do Roteamento de veículos possui algumas soluções, mas como ele é um problema de difícil resolução devido a quantidade de combinações de caminho que podem existir, ainda não se chegou em uma resposta perfeita para o problema.

Para a solução desenvolvida utilizou-se uma heurística Construtiva, que consiste em uma solução simples, passo a passo, adicionando locais e formulando um caminho através do lugar mais próximo do último lugar visitado. Posteriormente esta solução será melhorada através do uso de outras Meta-heurísticas e Heurísticas.

#### **Referências**

- GLOVER, F.; LAGUNA, M. Tabu Search. Massachusetts: Kluwer Academic Publishers, 1997.
- LIU, FUH-HWA & SHEN, SHENG-YUAN. A Method for Vehicle Routing Problem with Multiple Vehicle Types and Time Windows. Department of Industrial Engineering and Management National Chiao. BUENO, M. Informática Fácil para Concursos. (2005)