UM ESTUDO DE CASO PARA O PROBLEMA DE ROTAS VIA MÉTODOS EXATOS E HEURÍSTICOS.

Wesley Vagner Ines Shirabayashi (Universidade Estadual de Maringá)
E-mail: wvishirabayashi@uem.br

Resumo: Neste trabalho lidamos com o problema de rotas de uma empresa de estofados localizada no interior do Paraná. Para solucionar tal problema, primeiramente fizemos uma revisão bibliográfica sobre o Problema do Caixeiro Viajante (PCV) visto que o problema de rotas é uma variante de tal problema. Além disso, realizamos um estudo de métodos heurísticos, neste caso, Algoritmo Genético (AG) e Otimização por Colônia de Formigas (ACO - Ant Colony Optimization) que são comumente utilizados na resolução do PCV e suas variantes. Ambos os métodos foram implementados em Matlab e testes com dados reais da empresa foram realizados a fim de propormos uma melhoria na logística de entrega dos produtos. Os resultados obtidos através do AG e do ACO foram comparados com as rotas realizadas pela empresa e foram satisfatórios. Além dos métodos heurísticos, utilizou-se também o solver do Libre Office que resolve o problema via métodos exatos, mais especificamente, o método Simplex combinado com o *Branch and Bound*.

Palavras-chave: Problema de rotas, Branch and Bound, Algoritmo Genético, Otimização por Colônia de formigas.

Abstract: In this work we deal with the problem of routes of a upholstery company located in the interior of Paraná. To solve this problem, we first made a bibliographical review on the Traveling Salesman Problem (TSP) since the problem of routes is a variant of such problem. In addition, we performed a study of heuristic methods, in this case, Genetic Algorithm (AG) and Ant Colony Optimization (ACO), which are commonly used in the resolution of PCV and its variants. Both methods were implemented in Matlab and tests with actual data of the company were carried out in order to propose an improvement in the logistics of delivery of the products. The results obtained through the GA and the ACO were compared with the routes performed by the company and were satisfactory. In addition to the heuristic methods, it was also used the Libre Office solver that solves the problem via exact methods, more specifically, the Simplex method combined with Branch and Bound.

Keywords: Routes problem, Branch and Bound, Genetic Algorithm, Ant Colony Optimization.

1. Introdução

A pesquisa por técnicas que resultem em implementações computacionais eficientes é bastante ativa e possibilita a resolução de diversos problemas, desde os mais simples até os maiores e mais complexos (Edgar e Himmelblau, 1989). O interesse por técnicas de otimização é de grande importância no mundo atual, por exemplo, empresas sempre querem maximizar seus lucros ou diminuir seus custos, e nesse sentido a pesquisa se faz necessária para que sempre tenhamos métodos mais eficientes para resolver tais problemas.

Ademais, as rápidas transformações ocorridas nos últimos tempos em diversos setores, sobretudo no âmbito empresarial trás consigo a necessidade constante de inovação e estruturação em todos os setores de uma empresa. Nesse cenário, a logística, que representa um dos principais fatores de lucro/custo para as empresas merece atenção especial, bem como a necessidade em se propor soluções inovadoras e otimizadas (Arenales et al., 2006). Dentro das atividades logísticas, a distribuição de produtos é uma das mais importantes e que mais influenciam nas despesas e, consequentemente, economia para uma empresa.

Um modelo eficiente de distribuição deve ser constantemente buscado pelas empresas, fato que faz com que mais e mais o problema de rotas, roteamento de veículos, caminho mínimo dentre outros seja estudado. Há vários trabalhos na literatura que lidam com os problemas citados anteriormente e a busca por soluções para os mesmos são ancoradas em diferentes metodologias: métodos heurísticos, métodos de programação inteira e mista, métodos clássicos de programação linear, dentre outros (Lawler et al., 1985), (Dorigo & Gambardella, 1997), (Malaquias, 2006), (Santos & Munari, 2017), (Rodrigues, 2004), (Simon, 2013).

Diante disso, neste trabalho tratamos do problema de rotas de uma empresa de estofados localizada na cidade de Pirapó - PR que entrega seus produtos no Paraná e em outros estados. O estudo foi feito através dos conhecidos métodos heurísticos AG e ACO e do método exato *Branch and Bound* e o problema estudado é o Problema do Caixeiro Viajante (PCV). Em termos gerais, o PCV é um problema de otimização combinatória que consiste em encontrar a sequência de cidades a serem visitadas por um caixeiro viajante que minimize a distância total percorrida e garanta que cada cidade seja visitada somente uma vez (Simon, 2013).

Para resolver o PCV é necessário conhecer as distâncias entre todas as cidades. Quando essa distância não depende do sentido da viagem, dizemos que o problema é simétrico. Em problemas práticos, como o problema aqui apresentado, a distância entre as cidades é obtida via sistemas de informações geográficas, neste caso utilizamos o *Google Maps* através de uma rotina feita na linguagem *Python* para obter tais distâncias e, além disso, construir o grafo real de cada caso estudado.

2. Modelagem do Problema

Em geral, problemas de otimização de rotas e muitos outros problemas são modelados através da teoria de grafos. Seja $G=(N,A)$ um grafo onde N representa o conjunto de nós e A representa o conjunto de arcos. No problema aqui estudado, os nós representam as cidades e os arcos representam as ligações (rodovias) entre elas. A modelagem do PCV está descrita em (1)-(5) (Carvalho, 2007):

$$min \sum_{j=1}^{n} \sum_{i=1}^{n} c_{ij} x_{ij} \tag{1}$$

$$\sum_{i=1}^{n} x_{ij} = 1, \forall\, j \in N \tag{2}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \forall\, i \in N \tag{3}$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1, i, j = 1, \ldots, n \tag{4}$$

$$x_{ij} = \{0,1\}, i, j = 1, \ldots, n \tag{5}$$

A função objetivo (1) busca minimizar o somatório das distâncias entre as cidades da rota. As restrições (2)-(3) garantem que cada cidade $i \in N$ será visitada apenas uma vez. A restrição (4) garante que não haverá sub-rotas e a restrição (5) define x_{ij} como variável binária.

Essa modelagem geralmente esta descrita em trabalhos que lidam com problemas de rotas e suas variantes, porém a restrição de eliminação de sub-rotas (restrição (4)), na prática, gera um número exponencial de restrições, sendo inviável sua utilização sobretudo quando o problema é resolvido de forma exata. A fim de contornar esse problema com tal restrição RASMUSSEN (2011) propõe um conjunto de restrições em substituição à restrição (4). Vejamos:

Seja y_{ij} o fluxo do produto y ao longo do arco $(i,j) \in A$. As restrições de fluxo são:

$$\sum_{j=1}^{n} y_{ij} - \sum_{j=1}^{n} y_{ji} = 1, i = 2, 3, \ldots, n. \qquad (6)$$

$$\sum_{j=1}^{n} y_{1j} = n - 1 \qquad (7)$$

$$y_{1j} \geq 0 \qquad (8)$$

$$(n-1)\, x_{ij} - y_{ij} \geq 0 \qquad (9)$$

Essas restrições garantem que os nós sejam visitados somente uma vez, já que o fluxo começa com n-1 unidades, e diminui-se uma unidade para cada nó visitado, sendo que quando retorna ao nó de origem o fluxo será zero.

3. Metodologia de Solução

O problema de rotas e demais variantes do PCV possuem complexidade exponencial e no problema aqui estudado, a maior dificuldade é o alto número de possíveis rotas à medida em que o número de cidades aumenta.

Além dos métodos exatos, existem vários métodos heurísticos que podem ser utilizados e neste trabalho, optamos por utilizar dois deles: algoritmo genético e otimização por colônia de formigas. A seguir, temos uma explicação detalhada dos métodos utilizados neste trabalho.

3.1 *Branch and Bound*

O método Branch and Bound (branch significa ramo e bound significa limite) é um método exato para resolução de problemas de programação inteira e/ou mista. Foi proposto em 1960 (Belfiore & Fávero, 2013), sendo um método de divisão e conquista, em que o problema original é dividido ou ramificado em subproblemas menores. Resolve-se os subproblemas, sendo que estes geram limites superiores (para problemas de maximização) ou inferiores (para problemas de minimização) sobre o valor da função objetivo. As soluções dos subproblemas são combinadas até que se obtenha a solução ótima do problema original.

Para resolver o PCV via métodos exatos, geralmente os métodos se baseiam em procedimentos de enumeração implícita em árvore, conhecidos como *Branch-and-Bound* (B&B), para os quais têm sido propostas diferentes funções limitadoras (Cunha et al., 2002). Conforme dito anteriormente, os métodos exatos têm aplicação limitada para a solução do PCV, tendo em vista a complexidade combinatória deste problemas,

mas para os problemas que serão apresentados neste trabalho ele foi utilizado e a solução ótima foi encontrada em frações de segundo.

3.2 Algoritmos genéticos

Os Algoritmos Genéticos (AGs) vêm sendo utilizados com sucesso para encontrar boas soluções para uma ampla variedade de problemas de otimização, desde a sua introdução por Holland na década de 70 (Holland, 1975). Em um AG, uma população de possíveis soluções para um determinado problema evolui de acordo com operadores probabilísticos concebidos a partir de conceitos biológicos, de modo que há uma tendência de que os indivíduos representem soluções cada vez melhores à medida que o processo avança (Pinho et al., 2013).

LINDEN (2006) destaca as seguintes vantagens dos AGs sob outras técnicas de otimização:

- São robustos e aplicáveis a uma grande variedade de problemas, apresentando um bom desempenho para uma grande escala de problemas;
- Não são afetados por descontinuidades na função ou em suas derivadas. Isto faz com que os AGs sejam adequados para funções com descontinuidades, ou para funções com as quais não se podem calcular derivadas;
- São capazes de lidar com funções discretas e contínuas, podendo inclusive trabalhar com funções mistas;
- São apropriados para resolver problemas com espaços de busca grandes demais para serem resolvidos por técnicas de otimização tradicionais.

Os parâmetros de um AG são fundamentais para o seu bom funcionamento e determina-los é algo complexo que depende, sobretudo, das características do problema a ser resolvido e, em geral, os valores dos parâmetros são ajustados com base em trabalhos prévios, da literatura.
Os principais parâmetros são: o tamanho da população, o número de gerações, a taxa de cruzamento e a taxa de mutação.Segue abaixo a estrutura de um AG simples.

Algoritmo 1: Algoritmo AG

Início
$t = 0$
Inicializa população
Avalia população
Enquanto critério de parada não atingido
faça
$t = t + 1$
seleção dos pais;
reprodução (crossover)
mutação
avaliação
Fim-faça
Fim-enquanto.

3.3. Otimização por Colônia de Formigas - ACO

Otimização por Colonia de Formigas (Ant Colony Optimization - ACO) é inspirada no comportamento de formigas na busca por alimentos. Sempre que uma formiga precisa decidir para onde ir, ela usa informação proveniente de feromônio depositado previamente por outras formigas que já passaram por determinado local. A direção escolhida é aquela que tiver a maior quantidade de feromônio. Através desse processo de busca, as formigas são capazes de encontrar o menor caminho do seu ninho até uma fonte de alimento (Carvalho, 2007) (Dorigo & Caro, 1999). O feromônio é uma substância que evapora com o tempo, então quanto maior é a concentração de formigas passando pelo mesmo lugar, mais atrativo ele se torna para as próximas formigas.

3.3.1. Otimização por Colônia de Formigas para o PCV.

Aqui apresentamos o funcionamento do algoritmo de colônia de formigas proposto por (Dorigo & Gambardella, 1997). No ACO duas tarefas são principais:

1. A representação do problema de uma forma versátil, em geral uma estrutura na forma de um grafo, que permita uma regra probabilística de transição entre os nós baseada nos feromônios e no valor de cada arco;

2. Uma heurística para transição de nós que avalie a qualidade dos caminhos percorridos.

Para cada arco $(i,j) \in A$ do grafo, é definida uma variável τ_{ij}, conhecida como trilha artificial de feromônio e inicialmente este é igual para todos os arcos da rede. Cada formiga k constrói uma solução a partir de um dos nós do grafo de forma aleatória.

A cada nó, a formiga artificial executa uma função probabilística e constrói seu caminho movendo-se através de uma sequência de locais vizinhos, selecionados segundo a função descrita na Equação (10).

$$p_{ij}^{k}=\frac{\tau_{ij}^{\alpha}.\eta_{ij}^{\beta}}{\sum_{j\in J_i^k}\tau_{ij}^{\alpha}.\eta_{ij}^{\beta}} \qquad (10)$$

onde:

p_{ij}^{k} : probabilidade da formiga k, que se encontra na cidade i, escolher o nó j como próximo nó a ser visitado;

τ_{ij} :quantidade de feromônio existente no arco (i; j). Inicialmente, adota-se um mesmo valor para todos os arcos da rede;

η_{ij} : atratividade do arco (i,j). No PCV, adota-se $1/d_{ij}$ (inverso do valor da distância entre os nós i e j);

J_i^k : conjunto de pontos ainda não visitados pela formiga k, que se encontra atualmente no nó i;

α : parâmetro que pondera a importância relativa da trilha de feromônio na decisão de movimentação da formiga;

β : valor heuristicamente escolhido, que pondera a influência relativa da distância η_{ij} entre os nós i e j no processo de decisão.

A Equação (10) mostra que a preferência da formiga por um determinado caminho é maior para os caminhos com maior nível de feromônio e com menor distância. A atualização do feromônio é realizada local e globalmente. Na atualização local, feita através da Equação (11), deseja-se evitar um arco que está sendo escolhido por todas as formigas. Esse processo aumenta o poder de exploração das formigas.

$$\tau_{ij}=(1-\rho)\tau_{ij}+\rho\tau_0,\tau_0=(n.L_{nn}^{-1}) \qquad (11)$$

onde n é o número de nós do grafo, L_{nn} é um valor heurístico do menor caminho encontrado por uma formiga e ρ é um parâmetro que determina a velocidade de evaporação do feromônio.

Na atualização global, feita através da Equação (12), deseja-se recompensar os arcos que pertencem a rotas mais curtas, adicionando-se uma quantidade de feromônio inversamente proporcional ao tamanho da rota obtida (quanto mais curta a rota, maior será a quantidade de feromônio depositada).

$$\tau_{ij}=(1-\rho)\tau_{ij}+\rho\Delta\tau_{ij},\rho\in[0,1] \qquad (12)$$

onde,

$$\Delta\tau_{ij}=\frac{1}{L^k}\, se(i,j)\acute{e}usado. \qquad (13)$$

O primeiro termo das Equações (11) e (12) é responsável pela evaporação do feromônio. O parâmetro é utilizado para que os caminhos menos utilizados sejam esquecidos com o passar do tempo. O segundo termo da Equação (12) é responsável por aumentar a concentração de feromônio na menor rota. Tal procedimento se repete até que um número máximo de iterações tenha sido alcançado ou não aconteça melhorias nas soluções encontradas.
A seguir, apresentamos o algoritmo do método.

Algoritmo 2: Algoritmo ACO - TSP.

Inicialização
Para cada arco (i, j) do grafo, estabelece-se um nível inicial de feromônio.
Para cada formiga k, escolhe-se um nó para iniciar o percurso.
para t indo de 1 até máximo de iterações **faça**
para k indo de 1 até m **faça** (m: número de formigas)
Cada formiga k constrói uma caminho selecionando a próxima cidade a ser visitada segundo a regra probabilística:

$$p_{ij}^{k} = \frac{[\tau_{ij}]^{\alpha}.[\eta_{ij}]^{\beta}}{\sum_{j \in J_i^k} [\tau_{ij}]^{\alpha}.[\eta_{ij}]^{\beta}}.$$

Após cada transição da formiga k, aplique a regra de atualização local, motivada pela evaporação do feromônio,

$$\tau_{ij} = (1-\rho)\tau_{ij} + \rho\tau_0.$$

fim-do-para
Para cada formiga, calcule a distância $L_k(t)$ do caminho descoberto pela formiga k,
Se $L_k(t) < L^*$ **então** $S^* \leftarrow S_k(t)$;
Para cada arco (i, j), atualize o feromônio:

$$\tau_{ij} = (1-\rho)\tau_{ij} + \rho\Delta\tau_{ij},\ \Delta\tau_{ij} = \frac{1}{L^k}.$$

fim-do-para
Retorne a melhor solução S^*.

4. Estudo de caso

Neste trabalho estudamos o problema de roteirização real da fábrica TEMDA ESTOFADOS, que está localizada no distrito de Pirapó-PR, pertencente à cidade de Apucarana-PR. Os dados, a seguir, são referentes aos locais onde um caminhão deve passar para realizar as entregas. Utilizamos dois romaneios cedidos pela empresa e nomeados como Romaneio 325 e Romaneio 327. O AG e o ACO foram implementados em Matlab 7.0.1 numa plataforma core I3, 6Gb de memória RAM, sistema operacional Windows 7 e os testes com o solver do LibreOffice foram feitos nessa mesma plataforma.

Tabela 1: Romaneio 325

Romaneio a carregar - OP:325	Endereços
0: Temda estofados	Rua José Antônio Dante, 590 - Pirapó-PR, 86818-000
1: Briare móveis e decorações Ltda	Av. Piracicaba, 90 - Vila São João, Limeira-SP, 13480-316
2: Marsola móveis e decorações Ltda-ME	Av. Afonso Pansan, 997 - Vila Bertini, Americana-SP, 13473-620
3: José Alécio Torricelli - ME	R. João Lino, 47 - Centro, Santa Barbara´ D'Oeste - SP, 13450-033
4: Móveis Bertini – comércio varejista de móveis	R. Bento de Arruda Camargo, 1126 - Jd. Santana, Campinas - SP
5: Casa e Arte móveis e decorações - ME	Av. José Bonifacio,´ 2371 - Jd. das Paineiras, Campinas - SP

6: Thomaz Martins Rodrigues Jr	R. Cel. Nogueira Padilha, 479 - Vila Hortência, Sorocaba - SP, 18020-000
7: Joana Aparecida Cerqueria Arthur - ME	R. Dr. Campos, 170 - Centro, Cerquilho - SP, 18520-000

Na Figura 1, os dados da Tabela 1 estão expressos na forma de mapa.

Figura 1: Mapa do romaneio 325.

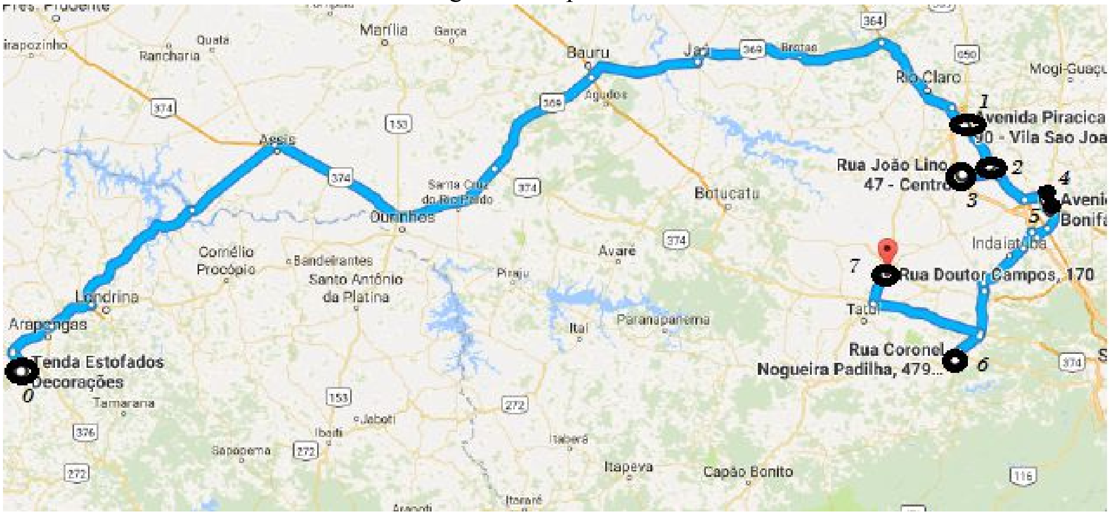
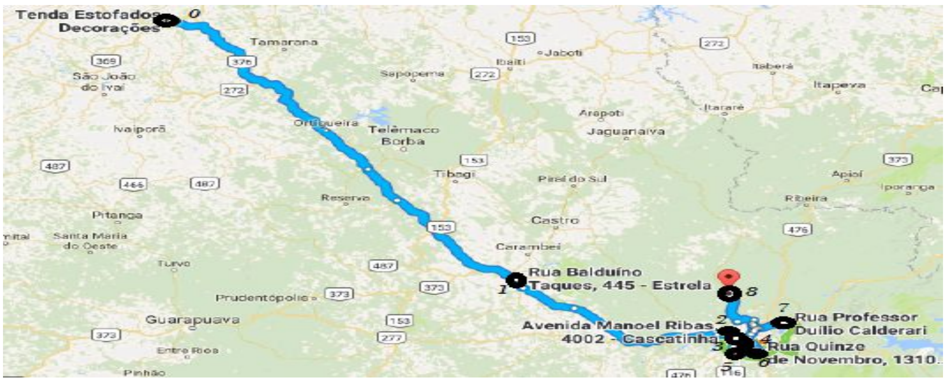


Tabela 2: Romaneio 327.

Romaneio a carregar - OP:327	Endereços
0: Temda estofados	Rua José Antônio Dante, 590 - Pirapó-PR, 86818-000
1: W. Sistak móveis	R. Balduíno Taques, 445 - Ronda, Ponta Grossa - PR, 84052-460
2: Katula comércio de móveis e decorações	Av. Manoel Ribas, 4002 - Cascatinha, Curitiba-PR, 82025-160
3: Dionisio Decio Jarosk	Rua Dom Pedro I, 191, Água Verde, Curitiba-PR, 80620-130
4: Fernando Oliveira Spait - ME	Rua Izaac Ferreira Cruz, 3700, Sítio Cercado, Curitiba-PR, 81910-000
5: Claudete Aparecida da Silva	Av. Mal. Floriano Peixoto, 6170 - Hauer, Curitiba - PR, 81630-000
6: Dom Antonio comércio de móveis Eireli	R. Quinze de Novembro, 1310 - Centro, São José dos Pinhais - PR, 83005-000
7: Dewitt e Zloty comércio de móveis	R. Prof. Duílio Calderari - Jd. Paulista, Campina Grande do Sul - PR, 83430-000
8: Tienda móveis e decorações Ltda - ME	R. Padre Ribeiro, 498 - Centro, Rio Branco do Sul - PR, 83540-000

Na Figura 2, os dados da Tabela 2 estão expressos na forma de mapa.

Figura 3: Mapa do romaneio 327.



4.1. Resultados obtidos via AG

Foi necessário realizar vários testes a fim de calibrar os parâmetros do AG. Após esta calibração, utilizamos os seguintes valores para os parâmetros: tamanho da população=10; número de gerações = 30; taxa de cruzamento = 0.7; taxa de mutação = 0.15.

Para o romaneio 325, as cidades devem ser visitadas na seguinte ordem: 0 - 7 - 1 - 3 - 2 - 5 - 4 - 6 - 0 e a distância total percorrida é 1352 km.

A fim de mostrar como o AG se comporta frente a variação de parâmetros, realizamos outro teste para o mesmo romaneio, com os seguintes valores de parâmetros: tamanho da população = 50; Número de gerações = 200; taxa de cruzamento = 0.6; taxa de mutação = 0.2. O resultado obtido, neste caso, foi: 0 - 7 -3 - 1 - 2 - 4 - 5 - 6 - 0 e a distância total percorrida é 1329 km.

Para o romaneio 327, utilizamos os mesmos valores de parâmetros citados anteriormente e obtivemos os seguintes resultados:

Conjunto de parâmetros 1: 0 - 1 - 8 - 7 - 2 - 3 - 4 - 6 - 5 - 0 e a distância total percorrida é 902 km.

Conjunto de parâmetros 2: 0 - 1 - 2 - 3 - 4 - 6 - 5 - 7 - 8 - 0 e a distância total percorrida é 885 km.

4.2 Resultados obtidos via ACO

O ACO foi implementado também em Matlab em um computador com as mesmas características descritas no AG. Neste algoritmo também foram necessários testes a fim de calibrar os parÂmetros, e após os mesmos, utilizamos os seguintes valores: α = 1, β = 2, ρ = 0.6, número de iterações = 10.

Após a execução do ACO, temos os seguintes resultados para cada romaneio:

Romaneio 325: 0 - 6 - 5 - 4 - 2 - 1 - 3 - 7 - 0 e a distância total percorrida é 1329 km.

Romaneio 327: 0 - 2 - 3 - 4 - 6 - 5 - 7 - 8 - 1 - 0 e a distância total percorrida é 885 km.

4.3 Resultados obtidos via Solver do LibreOffice.

A fim de resolver o problema via métodos exatos, utilizamos o solver do LibreOffice, que resolver problemas de Programação Linear através do método *Branch and Bound.*

Romaneio 325: 0 - 6 - 5 - 4 - 2 - 1 - 3 - 7 - 0 e a distância total percorrida é 1329 km.

Romaneio 327: 0 - 2 - 3 - 4 - 6 - 5 - 7 - 8 - 1 - 0 e a distância total percorrida é 885 km.

4.4 Análise dos resultados obtidos

A fim de avaliar os resultados obtidos pelo AG e pelo ACO comparamos com a distÂncia percorrida pelos caminhões da TEMDA ESTOFADOS. A seguir temos a rota e a distância percorrida pela empresa:

Para o romaneio 325, a distância total percorrida é 1359 km e as cidades são visitadas na ordem: 0 - 1 - 2 - 3 - 4 - 5 - 6 - 7 - 0.

Para o romaneio 327 a distância total percorrida é 893 km e as cidades são visitadas na ordem: 0 - 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 0.

Neste trabalho, foram implementados dois métodos heurísticos a fim de resolver o problema de rotas em uma empresa de estofados localizada no interior do Paraná. Além disso, utilizou-se o método exato *Branch and Bound*.

Tanto o AG quanto o ACO se mostraram eficientes na obtenção de soluções para os problemas testados. O AG é mais sensível à variação de parâmetros e, por esse motivo, foram apresentados testes para dois conjuntos de valores dos parâmetros.

Para o primeiro conjunto de parâmetros, a solução obtida para o romaneio 325 é 0.5% menor que a solução utilizada pela empresa. Já para o segundo conjunto de parâmetros, a solução obtida é 3.1% menor.

Quanto ao romaneio 327, a solução obtida pelo AG para o primeiro conjunto de parâmetros foi 1.6% maior quando comparada com a solução utilizada pela empresa. Já para o segundo conjunto de parâmetros, a solução foi 0.9% menor que a da empresa.

Quanto ao ACO, para o romaneio 325, a solução obtida é 2.2% menor que a solução utilizada pela empresa. Para o romaneio 327, a solução obtida é 0.9% menor.

A fim de obter a solução ótima do problema, utilizamos o método exato *Branch-and-Bound* através do solver do *LibreOffice* e observamos que os métodos heurísticos encontram a solução ótima para ambos os romaneios.

5. Considerações finais

Estudar e propor soluções para os diversos problemas reais tem desafiado os pesquisadores no sentido de se buscarem novas técnicas que permitam obter melhores resultados e resolver problemas de dimensões crescentes e cada vez mais complexos. No estudo de caso aqui apresentado, as dimensões são pequenas, mas nos permite realizar análises interessantes do ponto de vista da minimização de custos. Apesar da dimensão do problema ser pequena, optamos por utilizar dois métodos heurísticos clássicos e conhecidos na literatura: AG e o ACO. Ambos os métodos são utilizados com frequência em problemas semelhantes ao aqui apresentado e os resultados, em geral, são promissores.

A resolução via *Branch and Bound* foi feita com o intuito de verificar a qualidade da solução obtida pelos métodos heurísticos. Ressaltamos, que em problemas como esse, nem sempre é possível resolver via métodos exatos (se as instâncias forem muito grandes não é possível). Ao comparar as soluções, vemos que os métodos heurísticos encontraram a solução ótima.

Ao compararmos as soluções obtidas com a solução da empresa, apenas para um romaneio e um conjunto de parâmetros do AG, os resultados são ligeiramente piores e em todos os demais temos diminuições na quantidade de quilômetros rodados que, apesar do percentual se pequeno, são relevantes considerando que a empresa possui cinco caminhões que percorrem essas e outras rotas (que não foram analisadas neste estudo) com frequência. Se considerarmos, por exemplo, 1 ano com essas rotas sendo percorridas, certamente a empresa terá uma economia considerável relativa ao custo de transporte de produtos.

Referências

ALINAGHIAN, M. & SHOKOUHI, N. Multi-depot multi-compartment vehicle routing problem, solved by a hybrid adaptive large neighborhoord search. Omega, v. 76, p. 85–99, 2018.

ARENALES, M.; ARMENTANO, V.; MORABITO, R. & YANASSE, H. *Pesquisa Operacional*. Rio de Janeiro: Editora Campus, 2006.

BARBOZA, A. *Simulação e técnicas da computação evolucionária aplicadas a problemas de Programação Linear Inteira Mista.* 113p. Tese de Doutorado. CPGEI, Universidade Tecnológica Federal do Paraná, Curitiba-PR, 2005.

BELFIORE, P.& FÁVERO, L. P. *Pesquisa operacional para cursos de Engenharia.* Rio de Janeiro: Elsevier, 2013.

CARVALHO, M. B. *Aplicações de meta-heurística genética e fuzzy no sistema de colônia de formigas para o problema do caixeiro viajante.* 78p. Dissertação de Mestrado. Faculdade de Engenharia Elétrica, Universidade Estadual de Campinas, Campinas-SP, 2007.

CUNHA, C.;BONASSER, U. & ABRAHÃO, ***F.*** *Experimentos computacionais com heurísticas de melhorias para o problema do caixeiro viajante.* In: Anais do XVI Congresso da Associação Nacional de Pesquisa em Transportes. Natal, RN, 2002.

DORIGO, M. & CARO, G. *Ant colony optimization: a new meta-heuristic.* In: Proceedings of the congress on evolutionary computation. Piscataway, USA. IEEE, 1999. p. 1470–1477.

DORIGO, M. & GAMBARDELLA, L. *Ant colonies for the traveling salesman problem*. BioSystems, v.43, n. 2, p. 73–81, 1997.

EDGAR, T. & HIMMELBLAU, M. *Optimization of chemical processes*. New York: McGraw-Hill, 1989.

HOLLAND, J. *Adaptation in natural and artificial systems*. USA: MIT Press, 1975.

LAWLER, E.; RINNOOY-KAN, A.; LENSTRA, J. & SHMOYS, D. *The traveling salesman problem: a guided tour of combinatorial optimization*. New York: Wiley, 1985.

LINDEN, R. *Algoritmos Genéticos: uma importante ferramenta de inteligência computacional*. Rio de Janeiro: Editora Brasport, 2006.

MALAQUIAS, N. *Uso dos algoritmos genéticos para a otimização de rotas de distribuição*. 113p. Dissertação de Mestrado. Faculdade de Engenharia Elétrica, Universidade Federal de Uberlândia, Uberlândia-MG, 2006.

MICHALEWICZ, Z. *Genetic Algorithms + Data Structures = Evolution Programs*. London: Springer-Verlag, 1996.

PINHO, A.; MONTEVECHI, J.; MARINS, F., & MIRANDA, R. *Meta-heurísticas em Pesquisa Operacional (Algoritmos Genéticos: Fundamentos e Aplicações- Capítulo 2)*. Curitiba: Editora Omnipax, 2013.

RASMUSSEN, R. *TSP in spreadsheets -a fast and flexible tool*. Omega, v. 39, p. 51–63, 2011.

RODRIGUES, G. *Otimização de rotas através da aplicação de algoritmos exatos.* Trabalho de conclusão de curso, Universidade Presidente Antônio Carlos, 2004.

SANTOS, F. & MUNARI, P. O*timização do agrupamento de ordens e roteirização de coleta: um estudo de caso em um armazém de e-commerce.* Pesquisa Operacional para o Desenvolvimento, v. 9, n. 2, p. 62–81, 2017.

SIMON, D. *Evolutionary optimization algorithms-biologically inspired and population based approaches to computer intelligence*. New York: Wiley, 2013.