www.relainep.ufpr.br

REVISTA LATINO-AMERICANA DE INOVAÇÃO E ENGENHARIA DE PRODUÇÃO

LOGISTICS FOR DISTRIBUTION ROUTES: A CASE STUDY

LOGÍSTICA PARA ROTAS DE DISTRIBUIÇÃO: UM ESTUDO DE CASO

José F. F. Ribeiro[1], Pedro R. Galli✉[1]

[1]*Universidade de São Paulo - Faculdade de Economia, Administração e Contabilidade de Ribeirão Preto - Departamento de Administração, Ribeirão Preto/SP, Brasil*

[1]✉ pedro.galli@usp.br

Recebido: 22 março 2021 / Aceito: 13 abril 2021 / Publicado: 28 junho 2021

ABSTRACT. This paper presents a method and a computer program to solve the problem of the traveling salesman (PCV) in order to minimize the distance traveled in the distribution routes of a company that distributes office products, stationery, computers, hygiene, cleaning and disposables. The software used to solve the problem is the Evolutionary add-on for Microsoft Excel Solver, which has a pre-programmed genetic algorithm. The computer program was tested on the company's distribution routes in São Paulo and Minas Gerais, and the results obtained were better than those previously established by the company in all tests performed, requiring calculation times always less than two minutes.

Keywords: Logistics, Traveling salesman problem, Excel.

RESUMO. Neste artigo é apresentado um método e um programa computacional para resolver o problema do caixeiro viajante (PCV) com o objetivo de minimizar a distância percorrida nas rotas de distribuição de uma companhia distribuidora de produtos para escritório, papelaria, informática, higiene, limpeza e descartáveis. O software utilizado para resolução do problema é o suplemento *Evolutionary* do *Microsoft Excel Solver*, que tem um algoritmo genético pré-programado. O programa computacional foi testado em rotas de distribuição da companhia em São Paulo e Minas Gerais, e os resultados obtidos foram melhores que os anteriormente estabelecidos pela empresa em todos os testes realizados, exigindo tempos de cálculo sempre inferiores a dois minutos.

Palavras-chave: Logística, Problema do caixeiro viajante, *Excel*.

1 INTRODUÇÃO

A Delta Distribuidora é uma empresa situada na cidade de Pedreira, no estado de São Paulo, especializada na distribuição de produtos para escritório, papelaria, informática, produtos para higiene, limpeza e descartáveis. Sua distribuição é feita pelo interior do estado de São Paulo e também em algumas cidades do estado de Minas Gerais.

Neste artigo é apresentado um método e um programa computacional desenvolvido para auxiliar a companhia Delta Distribuidora a realizar a programação da entrega de produtos em suas rotas de distribuição.

O método proposto consiste na utilização da formulação clássica do problema do caixeiro viajante (PCV) para modelar o problema estudado. Na Seção 2 deste artigo está disponível uma revisão dos modelos e métodos de resolução propostos na literatura para o problema do caixeiro viajante. A metodologia utilizada para a resolução do problema está disponível na Seção 3. A Seção 4 apresenta o modelo matemático do problema estudado. A Seção 5 traz um exemplo ilustrativo. Testes realizados com as rotas utilizadas pela Delta Distribuidora são apresentados na seção 6. A Seção 7 fornece uma discussão sobre o método e programa propostos. A conclusão final é apresentada na Seção 8.

No método proposto o problema do caixeiro viajante é resolvido por meio do *Solver Evolutionary*, um suplemento pré-programado disponibilizado no *Microsoft Excel*. Trata-se de um algoritmo genético, pertencente à família de algoritmos evolutivos. A população de um algoritmo genético evolui a partir de operadores inspirados pela evolução biológica: seleção, cruzamento, reprodução, mutação. Os princípios básicos e as aplicações iniciais dos algoritmos genéticos podem ser encontrados em Olivier et al. (1987), Goldberg (1989) e Linden (2012).

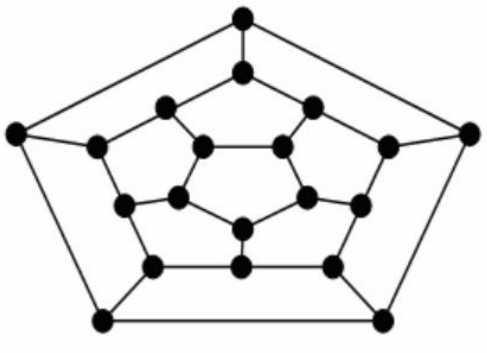
2 REFERENCIAL TEÓRICO

O modelo do caixeiro viajante (PCV) permite resolver o seguinte problema: Dada uma lista de cidades (ou vértices) e de estradas (ou arcos) entre cada par de cidades, encontrar o caminho de menor comprimento que um viajante pode utilizar para visitar todas as cidades, passando por cada uma delas apenas uma vez e por fim retornar ao ponto de partida (HILLIER e LIEBERMAN, 1990). Segundo Garey e Johnson (1977), o problema é NP-

completo, o que não permite a resolução de problemas reais de médio ou grande porte por meio de métodos exatos, abrindo um vasto campo de pesquisa e desenvolvimento de algoritmos aproximados.

O problema do caixeiro viajante foi originalmente proposto por Hamilton em 1857 (GOLDBARG e LUNA, 2005). O problema proposto consistia em encontrar uma rota através dos vértices de um dodecaedro (Figura 1), que iniciasse e terminasse no mesmo vértice, passando por todos, sem repetir algum vértice anteriormente já visitado. Cada vértice do dodecaedro estava associado a uma cidade. Por conta disso, um grafo é denominado hamiltoniano se ele dispõe de um ciclo com todos os seus vértices.

FIGURA 1 – DODECAEDRO DE HAMILTON



FONTE: GOLDBARG e LUNA (2005)

Posteriormente, Dantzig et al. (1954) formularam o problema do caixeiro viajante (PCV) como um problema de programação em variáveis 0/1 sobre um grafo G = (N, A), no qual N é o conjunto de nós e A é o conjunto de arcos. Essa formulação está presente na seção 4, onde é apresentado o modelo matemático do problema.

O problema do caixeiro viajante tem muitas aplicações práticas, tais como a programação de linhas aéreas (Clarke et al., 1997), o roteamento de veículos (Laporte, 1992), o projeto de circuitos integrados (Grotschel et al., 1991), o processamento de pedidos (Exemplo: Ratlif e Rosenthal, 1983) e o sequenciamento da produção (Lenstra et al., 1977). Em Dantzig e Ramser (1959) é apresentado um estudo do problema do caixeiro viajante para o roteamento de veículos.

Uma solução ótima para o PCV pode ser encontrada por métodos de programação inteira (Exemplos: Lawler et al., 1985; Junger et al., 1995) ou através da enumeração completa de todas as soluções possíveis. Porém, esses métodos não são viáveis para resolução

de problemas de grande porte, pois o numero de soluções possíveis é função fatorial do numero de nós, o que pode levar a um tempo computacional inviável. Pela dificuldade em encontrar soluções ótimas, as soluções aproximadas se tornam viáveis. Estas soluções podem ser encontradas através dos algoritmos heurísticos, que foram desenvolvidos com o propósito de resolver problemas de elevado nível de complexidade em tempo computacional razoável, sem, no entanto, conseguir definir se esta é a solução ótima, nem quão próxima ela está da solução ótima (CHAVES, 2003).

Existem três tipos de heurísticas para o PCV:

(1) Construção: Os métodos de construção inicializam a partir de um nó arbitrário, e através de algum critério de escolha, são selecionados os próximos nós. Exemplo: Rosenkrantz et al. (1977).

(2) Melhoria: Métodos de melhoria partem de uma solução factível (um circuito hamiltoniano), e a partir dessa solução são realizadas trocas para que seu comprimento seja reduzido, até que não seja possível reduzi-lo mais. Exemplo: Lin e Kernighan (1973).

(3) Metaheurísticas: Método de procedimentos destinados a encontrar uma boa solução (eventualmente ótima), consistindo na definição e aplicação de métodos heurísticos a um conjunto de diferentes problemas. Segundo Chaves (2003), a principal característica das metaheurísticas é a capacidade que estas possuem de escapar de ótimos locais dando certa flexibilidade às restrições do modelo. As heurísticas simples se baseiam na melhoria do movimento a ser executado, buscando um ótimo local, mas são limitadas, uma vez que fornecem sempre a mesma solução quando iniciadas num mesmo ponto inicial, enquanto que as metaheurísticas suprem essa deficiência, baseando-se na melhoria da solução, deixando temporariamente um ótimo local pré-estabelecido para procurar um ótimo local global, mesmo que para isso ocorra a perda temporária do valor da função objetivo. Uma descrição dos métodos metaheurísticos básicos está disponível em Reeves (1993). Uma comparação de desempenho entre seis heurísticas e metaheurísticas diferentes é apresentada por Halim e Ismail (2019).

Uma revisão bibliográfica sobre algoritmos genéticos desenvolvidos para a resolução do problema do caixeiro viajante pode ser encontrada em Kumar e Karimbir (2012), Rao e Hedge (2015), Vaishna et al. (2017) e Alzyadat et al. (2020).

O *Evolutionary Solver*, utilizado na resolução do problema, é um procedimento baseado em genética, evolução e sobrevivência do mais apto. Em decorrência, é chamado de algoritmo genético (metaheurístico). O *Evolutionary* começa gerando aleatoriamente uma população, e ao longo do processo mantém o controle de toda a população de soluções candidatas, de modo a evitar a armadilha de um ótimo local (LINDEN, 2012). Após gerar a população, o *Evolutionary* cria uma nova geração a partir da população inicialmente gerada. Influenciado pelos princípios da genética, nesse procedimento a prole combina alguns elementos de cada um dos pais.

Entre a população de soluções de qualquer geração, algumas serão aptas e outras inaptas, sendo isso determinado pela avaliação da função objetivo em cada uma das soluções candidatas da população. Subtrai-se uma penalidade de qualquer solução que não satisfaça uma ou mais das restrições. De acordo com Linden (2012), a partir dos princípios da evolução e da sobrevivência do mais apto, os membros aptos da população têm permissão para reproduzir com frequência, enquanto os membros inaptos não a recebem. Dessa forma a população evolui para ficar cada vez mais apta.

Segundo Linden (2012) outra característica importante dos algoritmos genéticos é a mutação. Do mesmo modo em que há mutação genética em biologia, o *Evolutionary Solver* irá ocasionalmente fazer uma mudança aleatória de um membro da população. Por exemplo, o valor de uma célula variável pode ser substituído por um novo valor aleatório. Esta mutação resulta na segregação de alguns descendentes do resto da população. Isto é importante, uma vez que pode auxiliar o algoritmo a se livrar, caso fique preso próximo de um ótimo local.

O *Evolutionary Solver* continua criando novas gerações de soluções até que não haja melhorias em varias gerações consecutivas. O algoritmo termina e a melhor solução encontrada até aquele momento é apresentada ao usuário.

3 MÉTODO

As planilhas de cálculo constituem a parte fundamental da implementação do método quantitativo desenvolvido neste trabalho para a resolução do problema do caixeiro viajante. Elas permitem análises de perspectivas diferentes e podem ser modificadas e melhoradas para refletir novas situações e opções a qualquer momento, sendo uma ferramenta muito efetiva para analisar e resolver problemas de logística e cadeia de suprimentos (DUAN et al., 2016).

A metodologia usada para desenvolver modelos de planilhas integradas é semelhante à modelagem com software especializado: o usuário desenvolve um modelo de referência refletindo as operações atuais, cria cenários alternativos e compara esses cenários à situação básica. A planilha e suas ferramentas complementares permite analisar o impacto das decisões sobre uma série de variáveis, tais como estratégias logísticas, planejamento de fluxo, controle de estoque, alocação e planejamento em geral.

O *Solver* foi projetado como uma extensão das planilhas do *Microsoft Excel* para modelar e resolver problemas de otimização lineares, não lineares e inteiros (FYLSTRA et al., 1998). A versão utilizada para o desenvolvimento desse trabalho inclui o suplemento *Evolutionary*, baseado em algoritmos genéticos, capaz de encontrar soluções de boa qualidade em tempo computacional razoável para o problema do caixeiro viajante (PCV).

Por outro lado, o *Microsoft Excel Solver* foi utilizado para o desenvolvimento deste trabalho por estar presente na maioria dos computadores, por ter uma interface simples e seus resultados serem apresentados de forma direta, o que pode facilitar no treinamento dos operadores e na tomada de decisão, tornando-a mais intuitiva, prática e satisfatória.

4 MODELO MATEMÁTICO

Segundo Goldbarg e Luna (2005) existem diversas formulações para o problema do caixeiro viajante (PCV). Neste artigo o problema foi formulado como um problema de programação binária sobre um grafo G = (N, A), onde N é o conjunto de nós e A é o conjunto de arcos conforme (HILLIER e HILLIER, 1999):

$$\text{Minimizar } f = \sum_{ij} c_{ij}\ x_{ij} \tag{1}$$

Sujeito a:

$$\sum_{i} x_{ij} = 1 \qquad (j = 1..N) \tag{2}$$

$$\sum_{j} x_{ij} = 1 \qquad (i = 1..N) \tag{3}$$

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

$$x_{ij} = 0/1 \tag{5}$$

Nesse modelo a variável $x_{ij} = 1$ caso a solução encontrada determinar o movimento do nó i para j; caso contrário, $x_{ij} = 0$; c_{ij} = distância do nó i para o nó j; c_{ii} = M, onde M é um valor suficientemente grande para garantir que o caixeiro viajante não retorne ao nó i após tê-lo deixado (pode-se também eliminar do modelo as variáveis x_{ii}); $c_{ij} \geq 0$ e |S| representa o número de vértices do subgrafo S.

A função objetivo (1) permite obter o comprimento total de todos os arcos incluídos no caminho hamiltoniano. As restrições (2) garantem a chegada a cada nó apenas uma vez. As restrições (3) estabelecem a saída de cada nó apenas uma vez. As restrições (4) eliminam os circuitos que não passam por todos os nós (pré-hamiltonianos). Pra cada um desses circuitos, uma restrição do tipo (4) é necessária, o que explica o alto número de restrições do modelo. A restrição (5) estabelece o tipo de variáveis (binárias).

5 EXEMPLO ILUSTRATIVO

Dadas as distâncias entre o ponto de partida (ponto 1) e os pontos 2, 3 e 4, resolver o problema do caixeiro viajante (PCV) e encontrar o caminho de comprimento mínimo que passa por todos os pontos apenas uma vez e retorna ao ponto de partida. A Tabela 1 fornece as distâncias (TAHA, 1971).

TABELA 1 – DISTÂNCIAS

	PONTO 1	**PONTO 2**	**PONTO 3**	**PONTO 4**
PONTO 1	0	10	17	15
PONTO 2	20	0	19	18
PONTO 3	50	44	0	25
PONTO 4	45	40	20	0

FONTE: TAHA (1971)

O modelo do caixeiro viajante correspondente é dado por:

Minimizar $f = 10x_{12} + 17x_{13} + 15x_{14} + 20x_{21} + 19x_{23} + 18x_{24} + 50x_{31} + 44x_{32} + 25x_{34} + 45x_{41} + 40x_{42} + 20x_{43}$

Sujeito a:

$x_{12} + x_{13} + x_{14} = 1;$ $x_{21} + x_{23} + x_{24} = 1;$ $x_{31} + x_{32} + x_{34} = 1;$

$x_{41} + x_{42} + x_{43} = 1;$ $x_{21} + x_{31} + x_{41} = 1;$ $x_{12} + x_{32} + x_{42} = 1;$

$x_{13} + x_{23} + x_{43} = 1;$ $x_{14} + x_{24} + x_{34} = 1;$

$x_{12} + x_{21} \leq 1;$ $x_{13} + x_{31} \leq 1;$ $x_{14} + x_{14} \leq 1;$

$x_{23} + x_{32} \leq 1;$ $x_{24} + x_{42} \leq 1;$ $x_{34} + x_{43} \leq 1;$

$x_{12} + x_{23} + x_{31} \leq 2;$ $x_{13} + x_{32} + x_{21} \leq 2;$ $x_{12} + x_{24} + x_{41} \leq 2;$

$x_{14} + x_{42} + x_{21} \leq 2;$ $x_{13} + x_{34} + x_{41} \leq 2;$ $x_{14} + x_{43} + x_{31} \leq 2;$

$x_{23} + x_{34} + x_{42} \leq 2;$ $x_{24} + x_{43} + x_{32} \leq 2;$

$x_{ij} = 0/1.$

O resultado obtido pelo *Microsoft-Solver-Excel (Evolutionary)* é igual a: $x_{12} = x_{24} = x_{43} = x_{31} = 1$ e todas as outras variáveis são iguais a zero. Portanto, o caixeiro viajante deverá percorrer a seguinte rota: ponto de partida 1 - pontos 2, 4 e 3 e retornar para o ponto de partida 1. O comprimento do caminho obtido é igual a 10 + 18 + 20 + 50 = 98. No modelo proposto foram eliminadas as variáveis x_{11}, x_{22}, x_{33} e x_{44}.

6 RESULTADOS

Para os testes computacionais apresentados nesse artigo foram utilizados dados reais fornecidos pela companhia Delta Distribuidora, uma empresa especializada na distribuição de produtos para escritório, papelaria, informática, produtos para higiene, limpeza e descartáveis. A Delta tem necessidade de definir rotas otimizadas de custo mínimo (ou distância mínima) para percorrer em seu trabalho de distribuição.

O objetivo dos testes computacionais é resolver o problema do caixeiro viajante (PCV) para verificar o desempenho do procedimento adotado e, assim, definir o caminho de comprimento mínimo (o menor possível), partindo da cidade de origem (Pedreira-SP),

passando por todas as cidades da rota, retornando ao ponto de partida, e efetuando as entregas nas cidades listadas.

São 9 rotas a partir de Pedreira, a saber: (1) Pedreira – Estiva, (2) Pedreira – Muzambinho, (3) Pedreira – Poços de Caldas, (4) Pedreira – Jundiaí, (5) Pedreira – Bauru 1, (6) Pedreira – Bauru 2, (7) Pedreira – Piracicaba, (8) Pedreira – Bueno Brandão e (9) Pedreira – Pouso Alegre.

A título de ilustração, a Tabela 2 fornece as cidades que compõem a rota entre as cidades de Pedreira e Jundiaí, bem como as distâncias em quilômetros entre elas.

TABELA 2 – ROTA PEDREIRA - JUNDIAÍ

	Pedreira	**Campinas**	**Valinhos**	**Vinhedo**	**Atibaia**	**Itatiba**	**Jundiaí**
Pedreira	0	32,7	37,8	49,6	84,9	72,9	67,3
Campinas	44,6	0	10,1	18,2	63,8	35,5	39,0
Valinhos	38,6	10,6	0	7,5	60,0	24,0	32,3
Vinhedo	62,5	19,9	7,0	0	57,2	21.7	24,2
Atibaia	85,1	65,6	73,5	58,8	0	37.0	51,6
Itatiba	72,5	37,6	24,7	21,9	35,7	0	28,7
Jundiaí	80,7	38,1	29,3	22,2	50,6	27,1	0

FONTE: AUTORES (2020)

A Tabela 3 apresenta o resultado obtido para a rota Pedreira – Jundiaí, obtido com o auxílio do *Solver Evolutionary*.

TABELA 3 – RESULTADO PEDREIRA-JUNDIAÍ

	Início	**1º**	**2º**	**3º**	**4º**	**5º**	**6º**	**Fim**
Circuito	Pedreira	Campinas	Valinhos	Vinhedo	Jundiaí	Itatiba	Atibaia	Pedreira
Distância	32,7	10,1	7,5	24,2	27,1	35,7	85,1	

FONTE: AUTORES (2020)

O caminho de comprimento mínimo entre as cidades de Pedreira e Jundiaí é constituído pelas cidades: Pedreira – Campinas – Valinhos – Vinhedo – Jundiaí – Itatiba – Atibaia e, finalmente, Pedreira (nessa ordem), com o comprimento total de 222,4 Km.

7 DISCUSSÃO

O método e a resolução do problema do caixeiro viajante (PCV) neste artigo foram desenvolvidos sobre a planilha *Microsoft Excel*, com auxílio do *Solver* e seu suplemento *Evolutionary*.

O *Solver* é uma das ferramentas presentes no *Excel*, simples e fácil de usar, e seu suplemento *Evolutionary* encontra-se disponível para uso, sem a necessidade de *downloads* ou instalação de novas extensões, o que facilita sua utilização.

Os testes computacionais realizados com os dados fornecidos pela companhia Delta Distribuidora para 9 rotas reduziram o comprimento mínimo de todas as rotas de distribuição da empresa, como mostra a Tabela 4. A melhoria obtida oscilou entre 2,98% no pior caso (rota Pedreira – Bauru 2) e 27,42% no melhor caso (rota Pedreira – Poços de Caldas).

O tempo computacional exigido pelo suplemento *Solver* do *Excel* para determinação de cada uma das rotas foi sempre inferior a 2 minutos (*Intel Core* i5 e 8 GB de RAM), mostrando a rapidez do *Solver Evolutionary* nos cálculos.

TABELA 4 – TESTES COMPUTACIONAIS

Rota	Número de cidades	Resultado Manual (Km)	Resultado *Evolutionary* (Km)	Melhoria (%)
Estiva	10	366,0	342,4	6,45
Muzambinho	11	596,8	507,9	14,90
Poços de Caldas	10	508,0	368,7	27,42
Jundiaí	7	253,9	222,4	12,41
Bauru 1	8	738,3	675,2	8,55
Bauru 2	9	771,2	748,2	2,98
Piracicaba	7	278,1	227,4	18,23
Bueno Brandão	7	226,4	187,2	17,31
Pouso Alegre	6	374,3	324,5	13,30

FONTE: AUTORES (2020)

A capacidade dos veículos não foi levada em consideração na resolução do problema, pois para todas as rotas de entregas disponibilizadas, a empresa possui veículos com capacidade suficiente para realizar a distribuição.

8 CONCLUSÃO

Neste artigo são apresentados um método e um programa computacional para resolver o problema do caixeiro viajante (PCV) com o objetivo de determinar o caminho de comprimento mínimo e a correspondente ordem das cidades a serem percorridas nas rotas de distribuição da companhia Delta Distribuidora, que atua no estado de São Paulo e em algumas cidades de Minas Gerais.

O método proposto consiste na resolução do problema do caixeiro viajante (PCV) através do *Solver*, com o auxílio de seu suplemento *Evolutionary*, disponíveis no *Microsoft*

Excel. O *Excel* é um software extremamente popular, está disponível na maioria dos computadores, e é de fácil utilização. O suplemento *Evolutionary* dispõe de um algoritmo genético pré-programado, capaz de definir uma solução de boa qualidade para o problema.

Nos testes computacionais realizados com dados reais fornecidos pela companhia Delta Distribuidora, o *Solver Evolutionary* resolveu o problema de maior porte (Pedreira – Muzambinho), com 11 cidades a serem visitadas, em menos de 2 minutos. O tempo computacional exigido para a obtenção dos resultados pelo *Solver Evolutionary* sempre foi inferior a 2 minutos em todos os testes realizados, utilizando-se um computador com processador Intel Core i5 e 8 GB de RAM, o que mostra a rapidez da ferramenta de cálculo.

Em todos os testes computacionais o *Solver-Evolutionary* determinou caminhos de comprimento mínimo sempre inferiores àqueles anteriormente estabelecidos pela empresa, com base na longa experiência de seus tomadores de decisão, o que demonstra o bom desempenho do método e do programa computacional desenvolvido. Em um dos testes realizados (rota Pedreira – Poços de Caldas) o resultado obtido permitiu uma redução no comprimento da rota de 508,0 Km para 368,7 Km, ou seja, uma melhoria de 27,42%.

REFERÊNCIAS

ALZYADAT, T., YAMI, M., CHETTY, G. Genetic algorithms for the travelling salesman problem: a crossover comparison. **International Journal of Information Technology**. 12, 209-213, 2020.

CHAVES, A. A. **Modelagens Exata e Heurística para Resolução do Problema do Caixeiro Viajante com Coleta de Prêmios**. Universidade Federal de Ouro Preto, Instituto de Ciências Exatas e Biológicas, Departamento de Computação. Ouro Preto, MG. 2003.

CLARKE, L.; JOHNSON, E.; NEMHAUSER, G.; ZHONGXI, Z. The aircraft rotation problem. **Annals of Operations Research**, 69, p. 33-46, 1997.

DANTZIG, G. B.; FULKERSON, D. R.; JOHNSON, S. M. **Solution of a large scale traveling salesman problem**, Technical Report P-510 RAND Corporation, Santa Monica, California, USA, 1954.

DANTZIG, G. B.; RAMSER, J. H. The truck dispatching problem. **Management Science**, 6, 1, p. 80-91, 1959.

DUAN, C. J.; HU, J.; GARROT, S. C. Using Excel Solver to solve Braydon farms' truck routing problem: A case study, **South Asian Journal of Management Sciences**, 10, 1, p. 38-47, 2016.

FYLSTRA, D.; LASDON, L.; WATSON, J.; WAREN, A. Design and use of the Microsoft Excel Solver. **Interfaces**, 28, 5, p. 29-55. 1998.

GAREY, M. R.; JOHNSON, D. S. **Computers and intractability: a guide to the theory of NP-completeness**. Freeman, 1977.

GOLDBARG, M. C.; LUNA, H. P. L. **Otimização combinatória e programação linear**. Elsevier, 2005.

GOLDBERG, D. E. **Genetic algorithm in search, optimization, and machine learning**. Boston: Addison Wesley, 1989.

GROTSCHEL, M.; JUNGER, M.; REINELT, G. Optimal control of plotting and drilling machines: a case study. **Mathematical Methods of Operations Research**, 35, 1, p. 61-84, 1991.

HALIM, A. H., ISMAIL, I. Combinatorial Optimization: Comparison of Heuristic Algorithms in Travelling Salesman Problem. **Archives of Computational Methods in Engineering**, 26, 367-380, 2019.

HILLIER, F. S.; HILLIER, M. S. **Introduction to management science**. McGraw-Hill, 1999.

HILLIER, F. S.; LIEBERMAN, G. J. **Introduction to Operations Research**. McGraw-Hill, 1990.

JUNGER, M. G.; REINELT, G.; RINALDI, G. **The travelling salesman problem. Handbook in Operations Research and Management Science**. M. O. Ball, Magnant, T. L., Monma, C. L., G. L. Nemhauser (eds.). Amsterdam: North Holland, 1995.

KUMAR, N.; KARAMBIR, K. R. A genetic algorithm approach to study travelling salesman problem, **Journal of Global Research in Computer Science**, 3, 3, p. 33-37, 2012.

LAPORTE, G. The vehicle routing problem: an overview of exact and approximate algorithms. **European Journal of Operational Research**, 59, 3, p. 345-358, 1992.

LAWLER, E. L.; LENSTRA, J. K.; RINNOOY KAN, A. H. J. R., Shmoys, B. B. (eds.). **The travelling salesman: a guided tour of combinatorial optimization**. Chichester, England: J. Wiley and Sons, 1985.

LENSTRA, J. K.; KAN, A. H. J. R.; BRUCKER, P. Complexity of machine scheduling problems. **Annals of Discrete Mathematics**, 1, p. 343-62, North Holland, 1977.

LIN, S.; KERNIGHAN, B. W. An effective heuristic algorithm for the travelling salesman problem. **Operations Research**, 21, p. 498-516, 1973.

LINDEN, R. **Algoritmos Genéticos**. São Paulo: Ciência Moderna, 2012.

OLIVER, I. M; SMITH, D. J.; HOLLAND, J. R. C. **A study of permutation crossover operators on the traveling salesman problem**. 2nd International Conference on Genetic Algorithms, Cambridge, USA, 1987.

RATLIF, H. D.; ROSENTHAL, A. S. Order-picking in a rectangular warehouse: A solvable case for the traveling salesman problem. **Operations Research**, 31, 3, p. 507-521, 1983.

RAO, I A.; HEDGE, S. K. Literature survey on travelling salesman problem using genetic algorithms. International **Journal of Advanced Research in Education Technology**, 2, 1, p. 42-45, 2015.

REEVES, (ed.): **Modern Heuristic Techniques for Combinatorial Problems**. New York: Wiley & Sons, 1993.

ROSENKRANTZ, D. J.; STEARNS, R. E.; LEWIS, P. M. An analysis of several heuristics for the travelling salesman problem. **SIAM Journal of Computing**, 6, 5, p. 63-581, 1977.

TAHA, H. A. **Operations research: an introduction**. New York: Pearson, 1971.

VAISHNA, P.; CHOUDHARY, N.; JAIN, K. Traveling salesman problem using genetic algorithm: A survey, **International Journal of Scientific Research in Computer Science**, Engineering and Information Technology, 2, 3, p. 105-108, 2017.