www.relainep.ufpr.br

REVISTA LATINO-AMERICANA DE INOVAÇÃO E ENGENHARIA DE PRODUÇÃO

LOGISTICS FOR FRUITS COLLECTION: A CASE STUDY

LOGÍSTICA PARA O RECOLHIMENTO DE FRUTAS: UM ESTUDO DE CASO

José Francisco Ferreira Ribeiro[1]✉, Lucas Campelo Ribeiro[1], Caio Vinicius de Aquino Siquitelli[1]

[1]Universidade de São Paulo, Ribeirão Preto, São Paulo, Brasil

✉ *jffr@fearp.usp.br*

Recebido: 17 abril 2019 / Aceito: 24 junho 2019 / Publicado: 11 julho 2019

ABSTRACT. In this paper we present a method and a computer program to solve the traveling salesman problem (TSP). The aim is to minimize the distance traveled in the transportation of fruits harvested for processing in a fruit juice industry. The corresponding program solves the problem using the Evolutionary add-in from Microsoft-Excel-Solver, which has a preprogrammed genetic algorithm, and has been tested to collect fruit harvested in Brazil and Ghana.

Keywords: Traveling Salesman Problem, Logistics, Optimization, Excel.

RESUMO. Neste artigo é apresentado um método e um programa computacional para resolver o problema do caixeiro viajante (PCV). O objetivo é minimizar a distância percorrida no transporte das frutas colhidas para processamento numa indústria de suco de frutas. O programa correspondente resolve o problema usando o suplemento *Evolutionary* do *Microsoft-Excel-Solver*, que tem um algoritmo genético pré-programado, e foi testado para efetuar o recolhimento das frutas colhidas no Brasil e Gana.

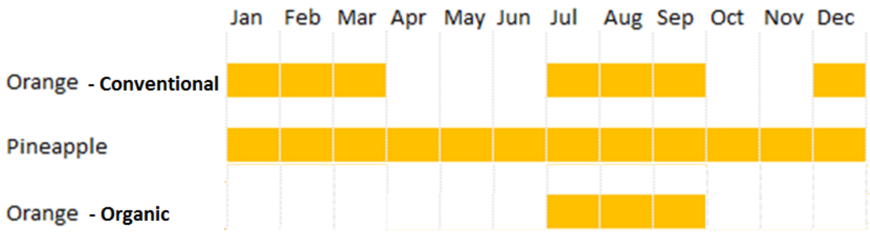
Palavras-chave: Problema do Caixeiro Viajante, Logística, Otimização, Excel.

1 INTRODUÇÃO

Sono Global é uma companhia com sede em Londres, Reino Unido, processadora de frutas para a produção de sucos (laranja, abacaxi, melancia, melão, maracujá, acerola, goiaba e outros), que produz, compra, coleta e industrializa as frutas colhidas em fazendas do Brasil e de Gana, na África Ocidental. O calendário para efetuar a colheita de frutas em Gana e no Brasil é fornecido nas Figuras 1 e 2, e constitui a base para o início das atividades de programação do recolhimento das frutas no campo.

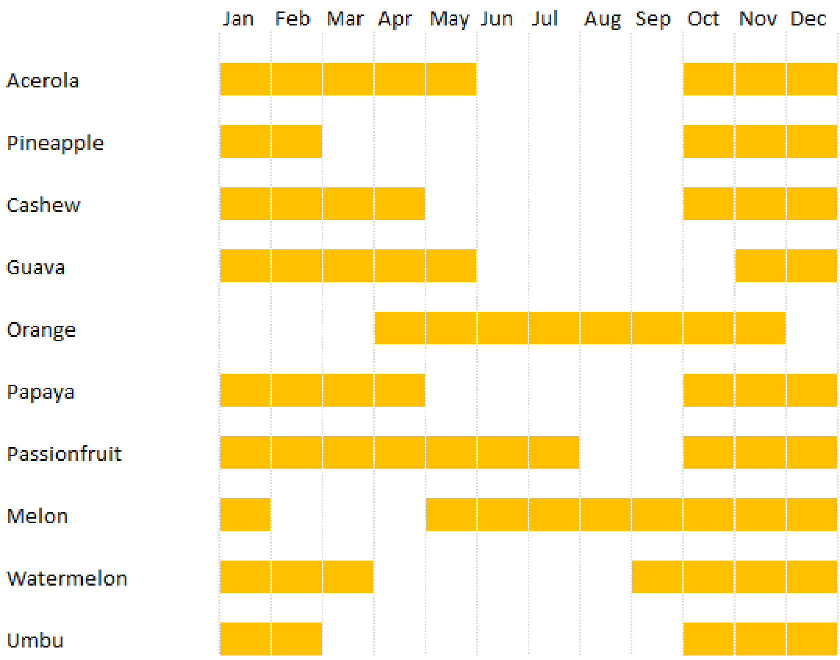
Neste artigo é apresentado um método e um programa computacional desenvolvido para auxiliar a Sono Global a programar o recolhimento das frutas colhidas em propriedades rurais do Brasil e de Gana.

FIGURA 1 – CALENDÁRIO DA COLHEITA DE FRUTAS EM GANA



FONTE: SONO GLOBAL (2018)

FIGURA 2 – CALENDÁRIO DA COLHEITA DE FRUTAS NO BRASIL



FONTE: SONO GLOBAL (2018)

O método proposto utiliza a formulação clássica do problema do caixeiro viajante (PCV) para modelar o problema estudado. Uma revisão dos modelos e métodos de resolução propostos na literatura para o problema do caixeiro viajante está disponível na Seção 2. A metodologia utilizada para a resolução do problema é fornecida na Seção 3. A Seção 4 apresenta o modelo matemático do problema estudado. A Seção 5 traz um exemplo ilustrativo. Um teste realizado para coletar frutas colhidas em Gana, com dados fornecidos pela *Sono Global*, é apresentado na Seção 6. A Seção 7 fornece uma discussão sobre o método e programa propostos. A conclusão é apresentada na Seção 8.

No método proposto o problema do caixeiro viajante é resolvido por meio do suplemento *Evolutionary* do *Microsoft-Excel-Solver*. O *Evolutionary* está pré-programado no *Solver* com um algoritmo genético. O algoritmo genético pertence 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 dos algoritmos genéticos estão elencados em Oliver *et al.* (1987). Goldberg (1989) deu início à utilização desses princípios para resolver problemas específicos de otimização.

2 REFERENCIAL TEÓRICO

O problema do caixeiro viajante (PCV) permite modelar e resolver a seguinte questão: dada uma lista de cidades (vértices) e de rotas (arcos) entre cada par de cidades, encontrar o caminho de comprimento mínimo que um viajante pode utilizar para visitar cada cidade exatamente uma única vez e retornar ao ponto de partida (HILLIER e LIEBERMAN, 1990; GONDRAN e MINOUX, 1984). Esse problema é NP-completo (GAREY e JOHNSON, 1977), 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.

Trata-se de um problema com 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 (RATLIF e ROSENTHAL, 1983) e o sequenciamento da produção (LENSTRA *et al.,* 1977).

O problema do caixeiro viajante foi originalmente proposto por Hamilton em 1857 (GOLDBARG e LUNA, 2005) sobre um dodecaedro, no qual cada vértice estava associado a uma cidade. A solução do PCV é chamada de ciclo hamiltoniano. Um grafo é denominado hamiltoniano se ele dispõe de um ciclo com todos os seus vértices. O PCV foi formulado por DANTZIG *et al.* (1954) como um problema de programação binária sobre um grafo $G = (N, A)$, no qual N é o conjunto de nós e A é o conjunto de arcos. Dantzig e Ramser (1959) apresentaram um estudo para o problema de roteamento de veículos.

Uma solução ótima para o PCV pode ser encontrada por métodos de programação inteira (LAWLER *et al.,* 1985; JUNGER *et al.*, 1995) ou através da enumeração completa de todas as soluções possíveis. Contudo, esses métodos não são viáveis para se resolver problemas de grande porte, pois o número de soluções possíveis é função fatorial do número de nós, o que pode levar a um tempo computacional inaceitável. Devido à dificuldade de encontrar a solução ótima, soluções aproximadas podem ser obtidas por meio de algoritmos heurísticos.

Existem três tipos de heurísticas para o PCV: (1) construção, (2) melhoria e (3) meta-heurísticas. Os métodos de construção inicializam a partir de um nó arbitrário, e então seleciona-se os próximos nós de acordo com um critério, como o menor preço ou a menor distância (Exemplo: ROSENKRANTZ *et al.*, 1977). Métodos de melhoria partem de uma solução factível, e então são efetuadas alterações com o objetivo de encontrar um caminho mais curto (Exemplo: LIN e KERNIGHAN, 1973). Meta-heurísticas tais como o *simulated annealing*, a busca tabu, os algoritmos genéticos e redes neurais artificiais implementam a busca de um ótimo local na vizinhança, e então usam essa informação para encontrar soluções melhores, sem se prenderem a um ótimo local. Uma descrição dos métodos meta-heurísticos básicos está disponível em Reeves (1993).

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

3 MÉTODO

Planilhas, incluindo o *Microsoft-Excel*, podem ser usadas de forma muito efetiva para analisar e resolver muitos problemas de logística e da cadeia de suprimentos (DUAN *et al.*, 2016). Planilhas permitem análises de perspectivas diferentes e podem ser modificadas e melhoradas para refletir novas situações e opções a qualquer momento.

A metodologia usada para desenvolver modelos de planilhas integradas é similar 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 *Microsoft-Excel-Solver* foi projetado para modelar e resolver problemas de otimização lineares, não lineares e inteiros (FYLSTRA *et al.*, 1998). A versão usada para desenvolver a ferramenta de cálculo apresentada nesse 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).

Além disso, o *Microsoft-Excel-Solver* foi utilizado no desenvolvimento deste trabalho pelas seguintes razões: (i) o *Excel* está disponível na maioria dos computadores; (ii) por ser uma interface simples, o treinamento dos operadores pode ser facilitado; (iii) os resultados gerados são apresentados de forma direta, tornando a tomada de decisão intuitiva, prática e satisfatória.

4 MODELO MATEMÁTICO

O PCV pode 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 o nó *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

O teste computacional apresentado nesse artigo foi realizado com dados reais fornecidos pela *Sono Global*. O objetivo é resolver o problema do caixeiro viajante (PCV) e definir um caminho de comprimento mínimo que percorra todas as cidades e retorne ao ponto de partida para efetuar a coleta de frutas em propriedades rurais de Gana, África Ocidental.

Para tanto, a Sono dispõe de caminhões de tamanhos variados, com capacidade suficiente para efetuar o transporte das frutas. A Tabela 2 fornece as cidades a serem visitadas e as distâncias (em quilômetros) entre elas.

TABELA 2 – CIDADES E DISTÂNCIAS EM GANA

	Pinora	Abakr	Abuen	Ajuma	Amoa	Asebu	Asuan	Benya	Efutu
Pinora	0	93,46	98,70	187,01	67,76	93,57	89,66	121,08	104,00
Abakr	93,46	0	5,24	96,78	49,89	5,20	5,55	28,20	10,80
Abuen	98,7	5,24	0	91,92	53,78	7,40	9,95	23,13	5,84
Ajuma	187,01	96,78	91,92	0	43,91	98,80	99,13	68,96	86,80
Amoa	67,76	49,89	53,78	43,91	0	46,37	50,19	75,40	59,46
Asebu	93,57	5,20	7,40	98,80	46,37	0	9,93	29,84	13,08
Asuan	89,66	5,55	9,95	99,13	50,19	9,93	0	31,46	14,46
Benya	121,08	28,20	23,13	68,96	75,40	29,84	31,46	0	17,40
Efutu	104,00	10,80	5,84	86,80	59,46	13,08	14,46	17,40	0

FONTE: SONO GLOBAL (2018)

A Figura 3 mostra os dados e as células preparadas para a aplicação do suplemento *E-volutionary* na planilha do *Microsoft-Excel-Solver*. A matriz de distâncias é composta dos pontos de partida e chegada (Cidade de Pinora) e mais oito cidades, a saber: Abakrampa, Abuenu, Ajumako, Amoanda, Asebu, Asuansi, Benyadzi and Efutu, onde estão localizadas as propriedades rurais para coletar as frutas. Nessas localidades serão coletadas as frutas colhidas em 111 propriedades rurais.

FIGURA 3 – PLANILHA COM AS CIDADES E DISTÂNCIAS EM GANA

			1	2	3	4	5	6	7	8
		PINORA	Abakrampa	Abuenu	Ajumako	Amoanda	Asebu	Asuansi	Benyadzi	Efutu
	PINORA	0,00	93,46	98,70	187,01	67,76	93,57	89,66	121,08	104,00
1	Abakrampa	93,46	0,00	5,24	96,78	49,89	5,20	5,55	28,20	10,80
2	Abuenu	98,70	5,24	0,00	91,92	53,78	7,40	9,95	23,13	5,84
3	Ajumako	187,01	96,78	91,92	0,00	143,91	98,80	99,13	68,96	86,80
4	Amoanda	67,76	49,89	53,78	143,91	0,00	46,37	50,19	75,40	59,46
5	Asebu	93,57	5,20	7,40	98,80	46,37	0,00	9,93	29,84	13,08
6	Asuansi	89,66	5,55	9,95	99,13	50,19	9,93	0,00	31,46	14,46
7	Benyadzi	121,08	28,20	23,13	68,96	75,40	29,84	31,46	0,00	17,40
8	Efutu	104,00	10,80	5,84	86,80	59,46	13,08	14,46	17,40	0,00

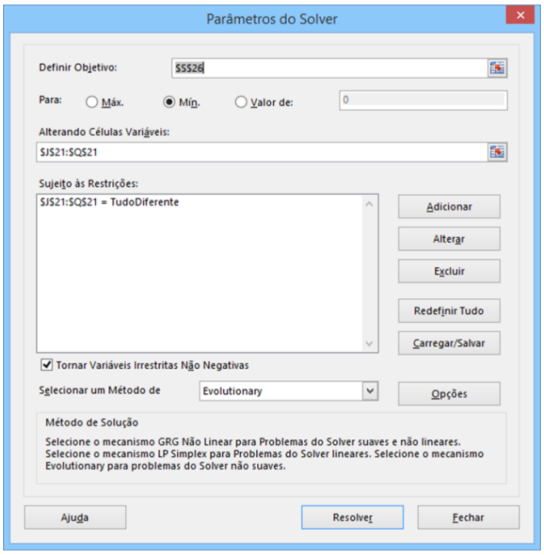
Circuito	Início	1º	2º	3º	4º	5º	6º	7º	8º	Fim
Rota										
LOCALIZAÇÃO	PINORA	#VALOR!	#VALOR!	#VALOR!	#VALOR!	#####	#######	#VALOR!	######	PINORA
Distância (Km)		#VALOR!	#VALOR!	#VALOR!	#VALOR!	#####	#######	#VALOR!	######	#VALOR!

Distância Total
#VALOR!

FONTE: SONO GLOBAL (2018)

A Figura 4 fornece os parâmetros do *Microsoft-Excel-Solver*: Função-objetivo, variáveis, restrições e o método de resolução escolhido (*Evolutionary*).

FIGURA 4 – PARÂMETROS DO SOLVER



FONTE: Os autores (2018)

A Figura 5 apresenta os resultados obtidos. O caminho mínimo é constituído por: Cidade de Pinora – Asuansi – Ajumako – Benyadzi – Abuenu – Abakrampa – Asebu – Amoanda – Cidade de Pinora, e o comprimento total é de 405,56 Km.

FIGURA 5 – CAMINHO MÍNIMO

Circuito	Inínio	1º	2º	3º	4º	5º	6º	7º	8º	Fim
Rota		6	3	7	8	2	1	5	4	
LOCALIZAÇÃO	PINORA	Asuansi	Ajumako	Benyadzi	Efutu	Abuenu	Abakram	Asebu	Amoan	PINORA
Distância (Km)		89,66	99,13	68,96	17,4	5,84	5,24	5,2	46,37	67,76

Distância Total
405,56

FONTE: Os autores (2018)

7 DISCUSSÃO

A resolução do problema do caixeiro viajante é realizada com auxílio do suplemento *Evolutionary* (algoritmo genético) disponível no *Microsoft-Excel-Solver*. O *Solver* é uma ferramenta simples e fácil de usar, e o suplemento *Evolutionary* está disponível para uso imediato, o que facilita sua utilização por técnicos e tomadores de decisão.

A Tabela 3 fornece os resultados obtidos para 10 testes computacionais realizados com dados fornecidos pela *Sono Global*. Os resultados do *Evolutionary* reduziram ou encontraram o mesmo comprimento do caminho mínimo obtido manualmente. O tempo computacional foi sempre inferior a 5 minutos, o que mostra a rapidez da ferramenta de cálculo. A melhoria alcançada atingiu 20,8%.

TABELA 3 – TESTES COMPUTACIONAIS

Teste	Número de cidades	Tempo Computacional (segundos)	Resultado Manual (Km)	Resultado *Evolutionary* (Km)	Melhoria (%)
1	4	14	165, 62	165, 62	0
2	5	17	218,46	218,46	0
3	6	20	329,47	287,27	12,8
4	9	28	405,56	405,56	0
5	10	93	435,67	435,67	0
6	12	101	548,22	500,30	8,7
7	13	112	644,34	632,78	1,8
8	14	124	670,12	670,12	0
9	16	165	832,35	690,44	17
10	20	290	884,88	700,46	20,8

FONTE: SONO GLOBAL (2018)

Para cada uma das cidades visitadas para a coleta de frutas, um novo problema do caixeiro viajante (PCV), de dimensão aproximadamente igual, tem que ser resolvido novamente para estabelecer a sequência das visitas às fazendas, uma vez que o teste apresentado na seção anterior estabelece somente a sequência das cidades.

A capacidade dos veículos não foi levada em consideração nesse trabalho, pois a empresa dispõe de veículos com capacidade suficiente para efetuar a coleta.

8 CONCLUSÃO

Nesse artigo é apresentado um método e um programa computacional para resolver o problema do caixeiro viajante (PCV) com o objetivo de determinar a ordem de passagem dos veículos para recolher as frutas colhidas nas propriedades rurais do Brasil e de Gana para posterior processamento de sucos de frutas.

O método proposto resolve o modelo matemático do problema do caixeiro viajante com auxílio do suplemento *Evolutionary*, disponível no *Microsoft-Excel-Solver*. O suplemento *Evolutionary* tem um algoritmo genético pré-programado, capaz de definir uma solução de boa qualidade para o problema. O *Excel* é um dos softwares mais difundidos e populares da história da computação, fácil de usar, e está disponível na maioria dos computadores.

No teste computacional com dados reais apresentado nesse artigo o *Evolutionary* resolveu em menos de 30 segundos de tempo de cálculo um problema de caixeiro viajante com nove cidades. O tempo computacional para testes realizados com exemplos de 4 a 20 cidades sempre foi inferior a cinco minutos, utilizando-se um computador com processador Intel Core 2 Duo e 4GB de RAM.

Em todos os testes realizados para a coleta de frutas no Brasil e em Gana para a *Sono Global* o programa computacional mostrou bom desempenho, com resultados melhores ou iguais àqueles obtidos anteriormente com base na experiência dos tomadores de decisão da companhia *Sono Global*.

O trabalho terá continuidade com a aplicação do método proposto para resolver o problema do roteamento de veículos que busca trabalhadores em suas residências para levá-los aos locais de trabalho, um problema especificamente relevante para empresas fornecedoras de mão de obra terceirizada.

REFERÊNCIAS

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. **Genetic algorithm in search, optimization, and machine learning**. Addison Wesley, 1989.

GONDRAN, M., MINOUX, M. **Graphs and Algorithms**. John Wiley, 1984.

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.

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.). Am-sterdam: 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.

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 Travel-ing 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, C. R. (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.

SONO GLOBAL, Disponível em< http://sono-global.com/>. Acesso em 24 de maio de 2018.

TAHA, H. A. **Operations research: An introduction**. 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.