

# Algoritmo Memético Aplicado ao Problema do Caixeiro Viajante de uma Empresa Situada no Município de Angicos/RN

Felipe R. dos S. Fernandes<sup>1</sup>, Ingridy M. P. Barbalho<sup>1</sup>, Francisco de A. P. V. de Arruda<sup>2</sup>, Matheus da S. Menezes<sup>2,\*</sup>

<sup>1</sup>Estudante de Graduação do Curso de Bacharelado em Sistemas de Informação  
Universidade Federal Rural do Semi-Árido (UFERSA) – Campus Angicos, RN – Brasil

<sup>2</sup>Professor do Departamento de Ciências Exatas, Tecnológicas e Humanas  
Universidade Federal Rural do Semi-Árido (UFERSA) – Campus Angicos, RN – Brasil

{felipe.ip,ingridymarina}@hotmail.com, {Matheus,xico}@ufersa.edu.br

**Abstract.** *This article describes the use of two Memetic Algorithm for the classic Traveling Salesman Problem for a company located in the municipality of Angicos / Rio Grande do Norte. The objective is to develop a Memetic Algorithm able to determine the best route to be traveled daily in order to minimize the cost of travel (distance and time). Are shown results obtained through the Exact Method and Memetic Algorithm. It is proposed a Memetic Algorithm using the Genetic Algorithm, with the mutation strategically modified, along with Local Search Shift. The results obtained are compared with the results of the Memetic Algorithm using traditional Genetic Algorithm.*

**Resumo.** *Este artigo descreve a utilização de dois Algoritmos Meméticos para o clássico Problema do Caixeiro Viajante de uma empresa localizada no Município de Angicos/Rio Grande do Norte. O objetivo do trabalho é desenvolver um Algoritmo Memético capaz de determinar a melhor rota a ser percorrida diariamente de modo que minimize o custo da viagem (distância e tempo). São apresentados resultados obtidos através do método exato e Algoritmo Memético. É proposto um Algoritmo Memético que utiliza o Algoritmo Genético, com a mutação estrategicamente alterada, juntamente com a Busca Local Shift. Os resultados obtidos são comparados com os resultados do Algoritmo Memético usando o Algoritmo Genético Tradicional.*

## 1. Introdução

O uso da Inteligência Artificial para resolução de problemas complexos se encontra cada vez mais em evidência nos últimos anos. Nessa categoria estão os problemas combinatórios, que são de simples resolução, porém a quantidade de combinações para encontrar a solução exata cresce de forma exponencial a medida que aumentamos a população a ser analisada (Maziero, 2003). Esses tipos de problemas influenciam diretamente no processo de Tomada de Decisão. A tomada de decisão está geralmente associada a problemas que surgem em processos operacionais (Oliveira, 2005). Para Fernandes et al (2014), a decisão faz parte do dia a dia e é inevitável deixar a deriva à importância de uma decisão correta, coerente e condizente com o problema enfrentado. Cardoso et al (2009) diz que tomar uma decisão é fazer uma escolha entre diversas

---

\* Orientador

opções disponíveis. Com isso, a criação de modelos é um fator importante para o apoio da tomada de decisão, onde é desejado a maximização da utilidade do decisor. Um sistema de apoio à tomada de decisão tem seu uso indicado nesses casos (Goldbarg; Luna, 2005). Segundo Gomes et al (2004), os problemas complexos de tomada de decisão são comuns em uma infinidade de áreas, como por exemplo, os jogos, administração de empresas, entre outros.

Para a resolução desses problemas complexos, como os Problemas de Redes (Lachtermarcher, 2009), há um campo de estudo, chamado Pesquisa Operacional, direcionado ao aprimoramento e desenvolvimento de técnicas adequadas para os problemas com a finalidade de encontrar soluções ótimas que ajudam a evitar desperdícios de tempo e dinheiro.

O problema apresentado no presente estudo foi extraído da rotina de uma empresa de laticínios, que consiste em realizar entregas de produtos diariamente em um conjunto de cidades do Estado do Rio Grande do Norte, partindo e encerrando a rota na cidade de origem, de tal forma que minimize o tempo total do percurso ou o custo total da rota. Este tipo de Problema de Roteamento de Veículos (PRV), que é uma generalização do Problema do Caixeiro Viajante (PCV), foi citado na literatura a partir do trabalho desenvolvido por Dantzig & Ramser (1959) e é um problema de otimização pertencente a classe NP-Hard.

Devido ao grande número de variáveis envolvidas, a modelagem do problema através de algoritmos exatos perde sua eficiência por se tratar de um problema de difícil solução exata (Goldbarg; Luna, 2005), pois exige algoritmos mais sofisticados que podem requerer mais recursos computacionais para se obter um limite de tempo viável. Com isso, a resolução de problemas de otimização através de algoritmos aproximativos surge como uma alternativa onde retorna soluções candidatas para o problema muito próximas da solução ótima, e em alguns casos, a própria solução ótima.

Na literatura há trabalhos relacionados ao mesmo PRV com diferentes abordagens para resolução do problema. Silva (2012) desenvolveu sua pesquisa utilizando um modelo exato, executado através do solver GLPK (Gnu Linear Programming Toolkit), para encontrar a solução ótima. Bezerra (2013) desenvolveu seu trabalho utilizando o Algoritmo Genético (AG) para buscar uma aproximação da solução ótima.

Este artigo aborda o problema como análogo ao clássico Problema do Caixeiro Viajante e busca otimizar o problema fazendo o uso de um Algoritmo Híbrido, também chamado de Algoritmo Memético, adicionando ao AG uma estratégia diferenciada na mutação e a Busca Local denominada Shift. Essa estratégia consiste em fazer permutações de clientes (ou Gene) antecedendo a Seleção Natural. Dessa forma, essa abordagem pode obter melhores soluções em relação ao Algoritmo Memético com o Algoritmo Genético tradicional.

## **2. O Problema do Caixeiro Viajante**

Na literatura é possível mensurar o quão explorado este problema é. O Problema do Caixeiro Viajante (PCV) é um dos mais tradicionais e conhecidos problemas de programação matemática (Goldbarg; Luna, 2005). O modelo clássico do PCV, bem como suas variações, ou, problemas correlatos, segundo Goldbarg e Luna (2005): o PCV Simétrico (PCVS), o PCV Generalizado (PCVG), o PCV com Janelas de Tempo

(PCVJT), o PCV Múltiplo (PCVM), o PCV com Coleta (PCV-C), o PCV Periódico (PCV-P) e outros, como o Problema do Caixeiro Alugador (Goldberg; Asconavieta; Goldberg, 2012) e o próprio PRV, são grandes representantes da área de Otimização Combinatória e, o desenvolvimento de algoritmos que resultem em soluções satisfatórias para esses problemas estão atreladas, geralmente, aos campos de estudo de Pesquisa Operacional e Inteligência Artificial ou Computacional.

O PCV consiste em visitar um conjunto de cidades constantes, descrito como um grafo  $G = (N, A)$ , onde  $N$  representa um conjunto de cidades e  $A$  é a distância ou o tempo entre pares de cidades conectadas, de tal forma que inicie e termine a rota na mesma cidade visitando as demais uma única vez, objetivando encontrar o caminho com menor custo. Segundo Goldberg e Luna (2005), “o PCV possui grande dificuldade de solução exata, devido a sua característica combinatória”. Dessa forma a solução determinística é exponencial, por isso considera-se esse problema pertencente a classe NP-Hard (Gutin; Punnen, 2002).

Na literatura existem diferentes formulações para esse problema. Este artigo irá adotar a comumente utilizada Formulação de Dantzig-Fulkerson-Johnson (DFJ), que é descrita a seguir, segundo Goldberg e Luna (2005):

$$\text{Minimizar } z = \sum_{j=1}^n \sum_{i=1}^n c_{ij} x_{ij} \quad (2.1)$$

Sujeito a:

$$\sum_{i=1}^n x_{ij} = 1 \quad \forall j \in N \quad (2.2)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad \forall i \in N \quad (2.3)$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1 \quad \forall S \subset N \quad (2.4)$$

$$x_{ij} \in \{0,1\} \quad \forall i, j \in N \quad (2.5)$$

Onde:

(2.1) representa a função-objetivo do problema, a qual define a minimização do custo total do caminho, considerando que  $c_{ij}$  é o custo de se transpor o arco  $(i, j) \in A$  e  $x_{ij}$  (figura 2.5) é a variável binária que assume valor igual a 1, se o arco  $(i, j) \in A$  for escolhido para integrar a solução, e 0 em caso contrário; (2.2) e (2.3) são restrições que asseguram que cada cidade é visitada uma única vez; a restrição (2.4) proíbe a formação de subrotas que não formam um circuito hamiltonianos.

Para Goldberg e Luna (2005) esta formulação destaca um importante aspecto do PCV que é sua natureza combinatória e que pela formulação fica claro que solucionar um PCV é determinar uma certa permutação legal de custo mínimo.

### 3. Algoritmo Evolucionário

Segundo Menezes (2014, p. 74), “Algoritmo evolucionário é um termo geral utilizado para a programação evolutiva, estratégias evolutivas e algoritmos genéticos.” Os Algoritmos Genéticos (AG) são métodos de otimização heurísticos de busca que se baseiam em mecanismos de seleção e evolução natural das espécies, declarado por Charles Darwin em 1859[(Goldberg e Luna, 2005) e (Linden, 2012)]. Os AG foram introduzidos por John Holland (Holland, 1975) e popularizado por David Goldberg (Goldberg, 1989), seu aluno.

As características principais que são intensamente usadas pelos AG, segundo Goldberg e Luna (2005), são:

**População**, que é um conjunto aleatório de indivíduos ou cromossomos (ou seja, conjunto de soluções do problema); **Cromossomo**, que representa um indivíduo ou uma solução na população. Apresentado normalmente como um vetor e também representa uma rota no caso PCV; **Fitness**, define a aptidão ou qualidade de dado indivíduo; **Gene**, representa uma componente do Cromossomo; **Alelo**, descreve os possíveis estados de um atributo do indivíduo (isto é, se é binário, simbólico ou real); **Locus**, representa a posição do atributo no cromossomo; **Operadores Genéticos** são regras que permitem a manipulação dos cromossomos através do **Crossover** (cruzamento), combinação dos pais para obtenção de indivíduos filhos, e **Mutação**, produz um novo filho alterando diretamente no cromossomo do pai; **Fenótipo**, denota o cromossomo codificado; e **Genótipo**, que representa a estrutura do cromossomo codificado.

A hibridização dos Algoritmos Evolucionários (AEH), também chamado de Algoritmo Memético (AM), acontece quando há o uso de algum tipo de processo de Busca Local no escopo do Algoritmo genético. Para Linden (2012), as tecnologias híbridas em geral têm um desempenho melhor do que cada uma das tecnologias separadas, pois utilizam ambas de forma sinérgica, combinando seus pontos fortes. A estratégia utilizada no presente estudo é denominada de Algoritmo Memético e foi proposta por Moscato (1989). É utilizada em uma vasta gama de aplicações, principalmente em problemas de Otimização Combinatória. Menezes (2014) cita que os algoritmos genéticos diferem dos algoritmos meméticos por este considerar a evolução Lamarckiana. O pseudocódigo do AM proposto é apresentado na Figura 1 e descrito a seguir.

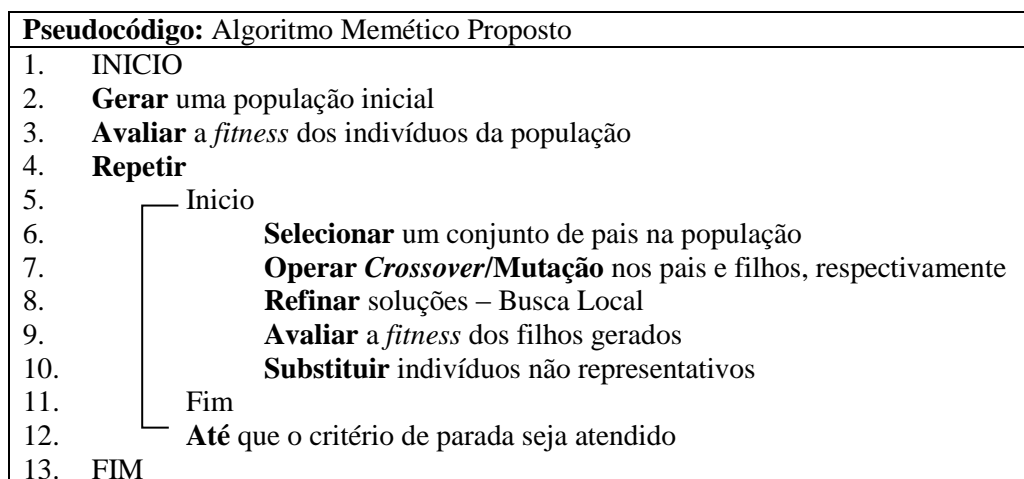


Figura 1: Pseudocódigo do Algoritmo Memético. Fonte: Autoria Própria, 2015.

### 3.1 Algoritmo Memético

O Algoritmo Memético (AM) proposto obedece o algoritmo da Figura 1. É adicionado uma nova estratégia nos operadores genéticos, diferenciando-o do Algoritmo Memético proposto por Fernandes et al (2015), que utiliza o AG tradicional. A seguir é apresentado as características do AM 1, de Fernandes et al (2015), e do AM 2, abordado por este trabalho.

#### 3.1.1 Algoritmo Memético 1 (AM 1)

**População Inicial:** A população inicial foi gerada aleatoriamente e seu tamanho é definido de acordo com o trabalho de Bezerra (2013). Para a simulação contendo uma única rota, com todas as cidades, o tamanho da população foi definido como sendo oito vezes o número de cidades (*i.e.*,  $8 * Num\_Cidades$ ); Já para as simulações com rotas menores, o tamanho da população recebe duas vezes o número de cidades (*i.e.*,  $2 * Num\_Cidades$ ).

**Cromossomo:** O cromossomo é representado por um vetor, conforme ilustrado na Figura 2, com representação por inteiros. O último elemento representa a aptidão (*Fitness*) do indivíduo e, vale salientar que este elemento não faz parte do Cromossomo (indivíduo), esta é apenas uma ilustração onde é mostrado a *Fitness* do indivíduo.

**Crossover:** O cruzamento é efetuado utilizando o *crossover* de 1 ponto (Fontes, 2006). A taxa de cruzamentos foi definida considerando dois casos: para uma única rota, que será selecionado 60% da população para a recombinação, e para as simulações com rotas menores, onde serão selecionados 40% dos indivíduos para o cruzamento.

**Mutação:** O procedimento considerado para a mutação consiste em permutar dois elementos de posição de forma aleatória. A mutação é realizada em 20% da população para ambos os casos (uma única rota e duas sub-rotas).

#### 3.1.1 Algoritmo Memético 2 (AM 2)

O que difere do AM 2 para o AM 1, é a forma da realização da mutação. No AM 1 a mutação é realizada de forma paralela ao *crossover* e recebe uma taxa para sua execução. Neste algoritmo, propomos a realização de maneira integrada da mutação e *crossover*. Assim, todos descendentes sofrerão a mutação, de forma a manter a diversidade da população. A taxa será única e será de 60% e 40%, respectivamente.

### 3.2 Busca Local

A busca local é uma técnica baseada em uma estrutura de vizinhança. É utilizada para refinar uma solução inicial intensificando a procura de uma solução final que represente um ótimo local ou uma solução aceitável. A estratégia de Busca Local é comumente usada em heurísticas genéricas para os problemas de otimização.

Neste trabalho, foi utilizado um algoritmo baseado na Busca Local *Shift(1,0)*, onde é realizado um movimento de permutação consecutiva entre duas cidades da solução (Mine et al, 2010). A Figura 2 ilustra a estratégia apresentada, onde para um dado problema com o número de cidades igual a cinco ( $n=5$ ), a Busca Local selecionará um indivíduo pai, que originará  $n-2$  indivíduos filhos, de tal forma que o indivíduo filho que apresentar a melhor *Fitness* dentre todos os gerados e tiver melhor adequação que a

do indivíduo pai, este irá o substituir na População. Mine et al (2010) ressalta que somente são permitidos movimentos que conduzam a soluções viáveis, ou seja, caso o refinamento não apresente rotas ou indivíduos bem sucedidos, permanece na população aquele com melhor aptidão. O AM efetua a busca local com os parâmetros ajustados para refinar 20% e 50% da população, respectivamente.

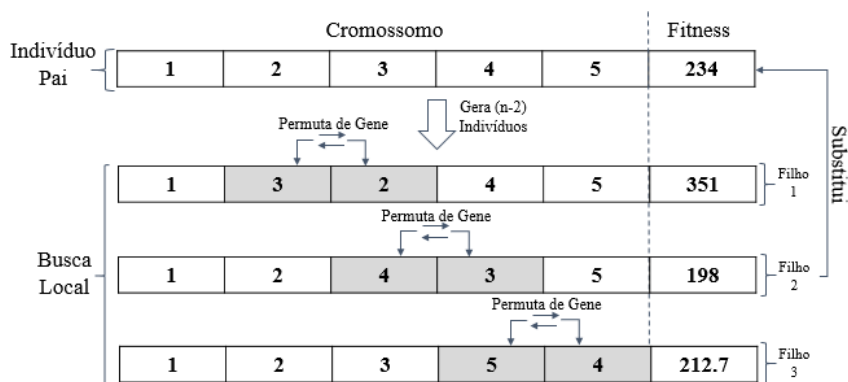


Figura 2: Processo da Busca Local Shift (1,0). Fonte: Autoria Própria, 2015.

#### 4. Definição do Problema

O PCV consiste em determinar a melhor rota sobre um conjunto de  $n$  cidades (Quadro 1), de forma a minimizar os custos da empresa de laticínios situada na cidade de Angicos/RN. Segundo Silva (2012), as características do problema remetem a utilização da abordagem do PCV, utilizando apenas um transporte em cada situação, de forma que supra toda demanda. Ainda segundo Silva (2012), podemos considerar três simulações distintas: uma **única rota**, com 12 cidades e duas **sub-rotas (SR)**, SR1 e SR2, com 6 e 7 cidades cada uma, respectivamente. Além disso, podemos considerar como parâmetro de minimização a distância percorrida (Km) e o tempo gasto (min).

Quadro 1. Identificação e numeração das Cidades de Estudo. Fonte: Silva (2012)

Nº	Cidade	Nº	Cidade	Nº	Cidade	Nº	Cidade
1	Angicos	4	Assu	7	Natal	10	Fernando Pedrosa
2	São Rafael	5	Jucurutu	8	Afonso Bezerra	11	Santana do Matos
3	Itajá	6	Mossoró	9	Pedro Avelino	12	Lajes

Como é um problema de otimização NP-Hard (Seção 2), Bezerra (2013) propôs o uso de um algoritmo genético para encontrar soluções que se aproximem dos resultados ótimos obtidos em Silva (2012), via método exato. Nessas condições, este artigo apresenta um Algoritmo Evolucionário Híbrido e compara o seu desempenho em relação ao AG e um AM que utiliza o AG na forma tradicional.

#### 5. Experimentos Computacionais

Para fazer uma comparação entre os métodos, o algoritmo proposto foi previamente modelado utilizando a plataforma Scilab e seus parâmetros foram ajustados de acordo com o trabalho de Bezerra (2013), com exceção da Busca Local (Seção 3.2).

Os experimentos computacionais foram realizados considerando os seguintes problemas:

- **Problema 1 e 2:** uma única rota (12 cidades);
- **Problema 3 e 5:** com a SR1 (cidades 1, 7, 8, 9, 10, 11 e 12); e,
- **Problema 4 e 6:** representada pela SR2 (cidades 1, 2, 3, 4, 5 e 6).

Em cada par de problema é realizado a análise da seguinte forma: Um considerando o caminho mínimo em quilômetros (Km) e outro o menor tempo em minutos (Min).

Os algoritmos foram executados em um Notebook Intel Core i5, com memória RAM de 8GB e Sistema Operacional Windows 8.1 64bits. A execução do Algoritmo Genético também foi realizada nas mesmas condições do Algoritmo Evolucionário Híbrido. Para todos os casos e problemas, considerou-se como critério de parada o limite de 500 Gerações. Cada problema foi executado 20 vezes.

## 6. Resultados

Os resultados dos experimentos computacionais são apresentados nas tabelas a seguir. A Tabela 1 apresenta os resultados obtidos pelo modelo exato. Na tabela 2 e 3 são apresentados os resultados obtidos pelo Algoritmo Memético 1 e pelo Algoritmo Memético 2, ambos com a Busca Local ocorrendo em 20% e 50% de sua população total. Para cada algoritmo é designado 3 colunas apresentando, nesta ordem: a melhor solução encontrada, a quantidade de vezes que esta solução ocorreu e o tempo médio gasto pelo algoritmo (em segundos). A Tabela 4 mostra a média e o desvio padrão resultante dos experimentos utilizando o Algoritmo Memético 1 e o Algoritmo Memético 2.

**Tabela 1. Resultados apresentados pelo modelo exato executado no Solver GLPK. Fonte: Silva (2012).**

<b>Método Exato</b>						
<b>Problema</b>	<b>1 (Km)</b>	<b>2 (Min)</b>	<b>3 (Km)</b>	<b>4 (Km)</b>	<b>5 (Min)</b>	<b>6 (Min)</b>
<b>Solução</b>	672,7	606	438,3	344,9	364	305

**Tabela 2. Resultados apresentados pelo Algoritmo Genético tradicional. Fonte: Bezerra (2013).**

<b>Algoritmo Genético Tradicional</b>						
<b>Problema</b>	<b>1 (Km)</b>	<b>2 (Min)</b>	<b>3 (Km)</b>	<b>4 (Km)</b>	<b>5 (Min)</b>	<b>6 (Min)</b>
<b>Solução</b>	672,7	606	438,3	344,9	364	305
<b>Qtd.</b>	8	13	20	13	20	20
<b>Tempo (s)</b>	232,45	231,32	22,49	22,28	22,26	22,15

**Tabela 3. Resultados do Experimento Computacional – BL 20%.**

Problema	AM 1 - BL de 20%			AM 2 - BL de 20%		
	Solução	Qtd.	Tempo (s)	Solução	Qtd.	Tempo(s)
1 (Km)	672,7	13	230,59	672,7	20	221,78
2 (Min)	606	15	231,87	606	20	225,06
3 (Km)	438,3	20	22,24	438,3	20	21,68
4 (Km)	344,9	19	22,27	344,9	20	21,19
5 (Min)	364	20	23,42	364	20	21,27
6 (Min)	305	20	22,77	305	20	21,09

**Tabela 4. Resultados do Experimento Computacional – BL 50%.**

Problema	AM 1 - BL de 50%			AM 2 - BL de 50%		
	Solução	Qtd.	Tempo (s)	Solução	Qtd.	Tempo(s)
1 (Km)	672,7	14	237,791	672,7	20	226,213
2 (Min)	606	16	238,252	606	20	227,484
3 (Km)	438,3	20	22,7095	438,3	20	22,127
4 (Km)	344,9	20	22,421	344,9	20	23,226
5 (Min)	364	20	22,7103	364	20	21,651
6 (Min)	305	20	22,4431	305	20	21,886

**Tabela 5. Média e Desvio Padrão da *Fitness* dos experimentos computacionais**

Problema	AG Tradicional		AM 1 BL de 20%		AM 2 BL de 20%		AM 1 BL de 50%		AM 2 BL de 50%	
	$\bar{x}$	$\sigma$	$\bar{x}$	$\sigma$	$\bar{x}$	$\sigma$	$\bar{x}$	$\sigma$	$\bar{x}$	$\sigma$
1 (Km)	683,2	12,07	679,2	10,38	672,7	0	675,02	6,56	672,7	0
2 (Min)	608,2	6,85	606,5	0,88	606	0	606,65	1,18	606	0
3 (Km)	438,3	0	438,3	0	438,3	0	438,3	0	438,3	0
4 (Km)	346,1	1,66	345,0	0,76	344,9	0	344,9	0	344,9	0
5 (Min)	364	0	364	0	364	0	364	0	364	0
6 (Min)	364	0	305	0	305	0	305	0	305	0

É possível observar que ambos algoritmos encontram soluções ótimas. Porém, analisando a coluna *Qtd.* das Tabelas 2, 3 e 4, verificamos que o algoritmo proposto apresenta melhores resultados, encontrando mais vezes solução globalmente ótima. Este fato é evidenciado pela análise da média das soluções na Tabela 5. Notamos que é dado um grau de confiabilidade maior ao AM 2, pois possui média semelhante as soluções globalmente ótimas obtidas pelo modelo exato, para todos os problemas. O desvio padrão corrobora a análise.



Em relação ao tempo de execução, apesar do algoritmo proposto conter a busca local e uma nova estratégia no AG, isso não implicou em uma grande diferença no tempo de execução, onde observamos nas Tabelas 2, 3 e 4, que o tempo nos problemas menores são equivalentes ou melhor. Nos problemas maiores, onde o espaço de busca é maior, existe uma pequena melhora no tempo de execução.

## 7. Conclusões

Este artigo apresentou uma proposta de Algoritmo Memético que utiliza a estratégia de Busca Local *Shift*(1,0) e que integra os operadores de cruzamento e mutação, aplicado ao Problema do Caixeiro Viajante apresentado por Silva (2012). Uma análise experimental foi realizada para verificar o desempenho do Algoritmo proposto em relação ao Algoritmo Memético que utiliza o AG tradicional e ao próprio AG tradicional. As soluções e o tempo de processamento obtido em cada metaheurística foram submetidas a uma comparação.

O experimento computacional realizado mostra que houve uma evolução em relação aos resultados apresentados pelo Algoritmo Genético (Bezerra, 2013) e Memético (Fernandes, 2015). Assim, podemos inferir que, neste caso, de acordo com os dados apresentados, quanto mais complexo ou maior for o problema dessa tipologia, mais adequado é a utilização dos Algoritmos Evolucionários Híbridos. Para o caso em questão, obteve-se melhoria nas soluções com tempo de processamento equivalente.

A estratégia de modificação dos operadores genéticos juntamente com a busca local teve pouco impacto no custo de tempo de processamento, e obteve melhor desempenho qualitativo quando comparada a estratégia do AG e AM 1. De modo geral, o Algoritmo proposto apresenta um comportamento satisfatório demonstrando melhorias em relação ao AG tradicional e AM 1.

Como trabalhos futuros, pretendemos expandir o uso dessas estratégias em Problemas do Caixeiro Viajante contidos na Traveling Salesman Problem Library TSPLIB, com o objetivo de analisar veementemente a eficiência do Algoritmo proposto.

## 8. Referências

- Bezerra, T. L. A.; Algoritmo Genético Aplicado ao Problema de Roteamento de Veículos nas Entregas Realizadas por uma Empresa de Laticínios: Estudo de Caso. 53 f. Monografia (Graduação em Ciência e Tecnologia) – Universidade Federal Rural do Semi-Árido, Angicos. 2013.
- Cardoso, R. S. et al . Uso de SAD no apoio à decisão na destinação de resíduos plásticos e gestão de materiais. *Pesqui. Oper.*, Rio de Janeiro , v. 29, n. 1, p. 67-95, Apr. 2009. Disponível em: <<http://dx.doi.org/10.1590/S0101-74382009000100004>>. Acesso em 29 de junho 2015.
- Dantzig, G. B.; Ramser, J. H. The truck dispatching problem. *Management science*, v. 6, n. 1, p. 80-91, 1959.
- Fernandes, F. R. S; Barbalho, I. M. P.; Avelino, J. G.; Correia, V. M. S; O Processo de Tomada de Decisão: A Utilização do Modelo Multicritério Macbeth, como Ferramenta Essencial no Apoio a Seleção de Gerente de Projetos. In: Escola Potiguar de Computação e suas Aplicações, 7. 2014, Santa Cruz, RN. Anais... Natal: Universidade do Estado do Rio grande do Norte, 2014. p. 216-223.

- Fernandes, F. R. S.; Barbalho, I. M. P.; Menezes, M. S.; Proposta de Um Algoritmo Memético Para o Problema de Otimização em Roteamento de Veículos em Uma Empresa de Laticínios. SBPO 2015. Disponível em: <<http://www.sbpo2015.iltc.br/pdf/146088.pdf>>. Acesso em 19 de dez. de 2015.
- Fontes, F. F. C.; Algoritmo Memético com Infecção Viral: Uma Aplicação ao Problema do Caixeiro Viajante Assimétrico. 72 f. Tese (Mestrado) -Universidade Federal do Rio Grande do Norte, Programa de Engenharia de Produção. Natal, 2006.
- Goldberg, M. C. Pacca Luna, H. Otimização Combinatória e Programação Linear. 2.ed. Rio de Janeiro: Elsevier, 2005.
- Goldberg, M. C.; Asconavieta, P. H.; Goldberg, E. F. G.; Memetic algorithm for the Traveling Car Renter Problem: an experimental investigation. Memetic Computing, v. 4, n. 2, p. 89-108, 2012.
- Goldberg, D. E. (1989). Genetic Algorithms in Search, optimization, and machine learning. Addison-Wesley.
- GOMES, L. F. A. M.; ARAYA, M. C. G.; CARIGNANO, C. Tomada de Decisões em Cenários Complexos. São Paulo:Thompson, 2004.
- Gutin, G.; Punnen, A. P. (Ed.). The traveling salesman problem and its variations. Springer Science & Business Media, 2002.
- Holland, J. H.; (1975). Adaptation in Natural and Artificial Systems. MIT Press.
- Lachtermarcher, G. Pesquisa operacional na tomada de decisões. São Paulo: Pearson Prentice Hall, 2009.
- Linden, R.; Algoritmos Genéticos. 3.ed. Rio de Janeiro: Editora Ciência Moderna Ltda., 2012.
- Maziero, E. A; Algoritmos - Simulated Annealing em Paralelo + Genético Crossover - Uma Abordagem Híbrida -. 154 f. Dissertação (Mestrado) - Universidade Federal de Santa Catarina - Sistemas de Computação. Florianópolis, 2003.
- Menezes, M. S.; O problema do Caixeiro alugador com coleta de prêmios: um estudo algorítmico. 126 f. Tese (Doutorado) - Universidade Federal do Rio Grande do Norte Centro de Ciências Exatas e da Terra. Programa de Pós-Graduação em Sistemas e Computação. Natal, 2014.
- Mine, M. T.; Ochi, L. S. et al.; O problema de roteamento de veículos com coleta e entrega simultânea: uma abordagem via Iterad Local Search e GENIUS. Transportes, v. XVIII, n. 3, p. 60-71, set. 2010.
- Moscato, P.; On evolution, search, optimization, genetic algorithms and martial arts: Towards Memetic Algorithm. [S.l.], 1989.
- Oliveira, A. C. M; Introdução à Pesquisa Operacional. Disponível em: <[http://www.deinf.ufma.br/~acmo/grad/PO\\_c01\\_v2005.pdf](http://www.deinf.ufma.br/~acmo/grad/PO_c01_v2005.pdf)>. Acesso em: 01 de set. 2015.
- Silva, G. L. S.; Roteamento de veículos nas entregas realizadas por empresa de laticínios - estudo de caso. 52 f. Monografia (Graduação em Ciência e Tecnologia) – Universidade Federal Rural do Semi-Árido, Angicos. 2012.