ALGORITMO DE COLÔNIA DE FORMIGAS APLICADO AO PROBLEMA DO CAIXEIRO VIAJANTE: ESTUDO DE CASO.

Abdiel Jônatas Alves da Silva[1], Matheus da Silva Menezes[2]

Resumo: O Problema do Caixeiro Viajante é um problema de otimização combinatória, sendo considerado NP-Difícil, e tem o objetivo de encontrar a solução ótima em um tempo de processamento computacional aceitável, visto que o espaço de soluções cresce exponencialmente com o número de entradas do problema. Por esse motivo, as meta-heurísticas são estratégias bem adaptadas na obtenção da solução ótima ou resultados próximos a esta solução. Este artigo faz uso da implementação da meta-heurística de Colônia de Formigas, adaptada ao Problema do Caixeiro Viajante contido em uma empresa distribuidora de laticínios, localizada em Angicos/RN. O objetivo é validar a eficiência do método proposto, considerando o tempo de execução em cada rodada e a qualidade da solução obtida pelo algoritmo. Os resultados obtidos são comparados com outros trabalhos deste mesmo problema contidos na literatura, que fazem uso de outros métodos de busca. Através da análise comparativa, é possível verificar que a meta-heurística proposta é um método eficaz na busca por boas soluções para o problema estudado, tanto em relação aos resultados gerados quanto ao tempo de processamento computacional do algoritmo.

Palavras-chave: Problema do Caixeiro Viajante; Otimização Combinatória; Colônia de Formigas

1. INTRODUÇÃO

Diversos problemas científicos ou do cotidiano podem ser modelados para minimizar ou maximizar uma função. Dentre estes problemas, temos um sub-campo denominado de otimização combinatória, que engloba diferentes variáveis interligadas, de forma que em determinadas situações hajam diversos cenários para tomada de decisão, buscando por uma solução que faça melhor uso dos recursos envolvidos [7]. A Pesquisa Operacional é um campo de conhecimento interdisciplinar e, juntamente com a programação matemática, faz uso de algoritmos e técnicas para resolução de problemas reais, examinando um conjunto de soluções e respeitando um conjunto de restrições, para uma tomada de decisão eficiente. Para ilustrar a diversidade de áreas, os problemas de otimização combinatória podem ser empregados em: empacotamento de containers, localização de centros distribuidores, Problema do Caixeiro Viajante, Projeto de Redes de Telecomunicação, dentre outros [4].

No Problema do Caixeiro Viajante (PCV) clássico, deve-se efetuar um caminho hamiltoniano nas arestas de um grafo, ou seja, visitando cada vértice apenas uma vez. A analogia é que as arestas são as estradas e os vértices são as cidades visitadas por um caixeiro. O objetivo é visitar todas as cidades, passando apenas uma vez em cada, ao menor custo possível [4]. Uma das aplicações do PCV é o Roteamento de Veículos, que compreende na logística do atendimento de consumidores através de uma frota de veículos. A resolução de um PCV se dá por meio de *solvers*, que rodam algoritmos de programação inteira, como o Simplex. Contudo, o PCV é NP-Difícil, o que implica no fato de problemas com elevado número de cidades serem difíceis de resolver em um tempo computacional aceitável.

Nos últimos anos, técnicas computacionais inteligentes estão sendo bastante exploradas e difundidas, com alto nível de eficiência. Nesse contexto, surgiram as meta-heurísticas adaptadas ao problema, como a de Colônia de Formigas, baseada na população e com foco na inteligência coletiva. O comportamento das formigas na procura de alimento e o depósito de feromônio no percurso faz deste um ótimo método para problemas de roteamento. A estratégia consiste em inserir uma formiga em cada cidade e a escolha da próxima cidade a ser visitada é baseada na fórmula da formiga. Ao realizar novas iterações, algumas rotas tendem a serem fortalecidas, definindo boas soluções com potencial de otimalidade. O feromônio inicial evapora, evitando crescimento acumulado, permitindo esquecer pobres decisões do passado de busca e ampliando a busca por novas soluções, e portanto, o depósito do novo feromônio é realizado por todas as formigas que cruzaram alguma rota [6].

Atualmente, grande parte dos gastos existentes nas empresas condizem a atividade logística, na distribuição dos seus produtos aos consumidores, e por isso, há a necessidade de otimizar estes processos visando a redução dos custos envolvidos. O problema adotado nesta pesquisa caracteriza-se como uma aplicação prática do Problema do Caixeiro Viajante, que é o Problema de Roteamento de Veículos (PRV). A Associação dos Pequenos Agropecuaristas do Sertão de Angicos (APASA), realiza entregas de produtos com necessidade de transporte refrigerado, diariamente em um conjunto de cidades do Estado do Rio Grande do Norte. Deve-se encontrar uma rota que passe exatamente uma única vez por cada cidade, de tal forma que minimize o tempo total do percurso ou a distância total da rota. Os objetivos deste trabalho são: implementar o algoritmo de Colônia de Formigas adaptado ao problema da APASA; validar a eficiência do método proposto, considerando o desempenho computacional gasto em cada execução e a qualidade das soluções encontradas; e analisar os dados obtidos comparando-os com os trabalhos contidos na literatura, os quais resolvem o mesmo PCV através de métodos exatos e aproximativos.

2. REFERENCIAL TEÓRICO

Nesta seção, serão abordados os conceitos teóricos base pertinentes ao objetivo deste trabalho, sendo estes: o Problema do Caixeiro Viajante, o Algoritmo de Colônia de Formigas e as Meta-Heurísticas. Para um melhor entendimento do trabalho é desejável que o leitor tenha um conhecimento básico sobre alguns aspectos teóricos da área de Pesquisa Operacional. Para este fim, indicamos a leitura de [4].

2.1. O Problema do Caixeiro Viajante

Também conhecido como Problema do Vendedor Itinerante, o Problema do Caixeiro Viajante (PCV) é um dos mais famosos problemas de otimização combinatória. Antigamente, os caixeiros viajantes se deslocavam por diversas cidades com o objetivo de conhecer e vender os seus produtos. No entanto, os meios de locomoção eram bastante restritos, por isso, deviam encontrar rotas inteligentes para visitar cada cidade uma única vez antes de retornar ao ponto de partida, de modo a minimizar os custos envolvidos.

A Figura 1 representa um exemplo de grafo do PCV. Nele constam os custos entre cada vértice, denominados como cidades, representadas pelas letras A – E, conectadas por arestas. As arestas podem ser simétricas ou assimétricas, quando possui o mesmo custo para ambas as direções ou com custos distintos para diferentes direções, respectivamente. Para este trabalho, faremos uso das rotas simétricas.

Apesar do problema ter origem antiga, o PCV aplica-se igualmente no cotidiano em uma série de casos. Segundo [4], grande parte dos problemas de roteamento estão ligados a pontos de oferta e demanda, podendo estes pontos representar cidades, depósitos, centrais de distribuição ou coleta, etc. Por exemplo, no caso de uma transportadora com entregas para diferentes regiões de uma cidade, formando uma variedade de possíveis rotas.

Para uma pequena quantidade de pontos de entrega, problemas deste tipo podem ser sanados através de métodos exatos, como o Simplex. Este método é empregado na resolução de problemas de alocação de recursos e pertence a um ramo da Pesquisa Operacional chamado de Programação Linear [2]. A Programação Linear é usada na tomada de decisões com o intuito de encontrar a melhor resposta em um dado conjunto com várias soluções possíveis, visando otimizar recursos.

Quando a quantidade de pontos é maior, o método Simplex tende a se tornar inviável, devido a infinidade de cálculos e maior período de processamento computacional, tornando praticamente impossível a busca pela solução ótima. Assim, embora não seja possível garantir encontrar a melhor rota, é possível encontrar um subconjunto de soluções viáveis em um tempo de processamento computacional aceitável, baseando-se por métodos heurísticos.

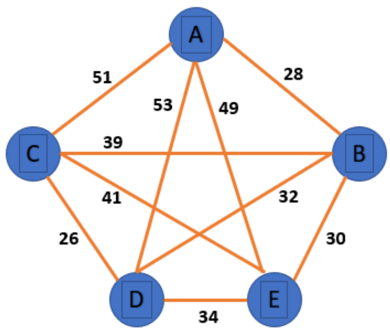

Figura 1. Exemplo ilustrativo de um grafo de PCV. (Autoria própria)

2.2. Meta-heurísticas

Existe uma diversidade de algoritmos que descrevem uma solução ótima para diferentes modelos da pesquisa operacional, estes algoritmos são representados via programação matemática com características de programação linear, programação inteira, programação mista e programação não-linear. Tais algoritmos são de grande importância na resolução de muitos problemas práticos, mas nem sempre conseguem alcançar a melhor solução devido à limitações de capacidade de processamento e tempo de resposta do algoritmo.

As meta-heurísticas são caminhos alternativos para encontrar uma boa solução viável com tempo de processamento aceitável. Segundo [1], um método heurístico é um procedimento que provavelmente vai encontrar uma excelente solução viável, mas não necessariamente uma solução ótima, para o problema específico em questão.

Uma meta-heurística é um método de resolução geral que fornece tanto uma estrutura, quanto diretrizes de estratégia gerais para desenvolver um método heurístico específico que se ajuste a um problema particular [1]. Algoritmos meta-heurísticos devem ser capazes de lidar com problemas complexos e iterativos e, ao fim do processamento, em um tempo razoável, deve informar o resultado da melhor entre todas as iterações realizadas. Esses métodos variam conforme cada caso, devendo se adequar individualmente conforme o problema de estudo. Desta forma, as técnicas meta-heurísticas são estratégias interessantes na busca de soluções de um problema de otimização combinatória.

2.3. Método de Colônia de Formigas

Segundo Marx, o homem transforma a realidade natural, "primeira natureza" em uma "segunda natureza", artificial, humanizada. Tal realidade moldada é denominada superação dialética do dado natural [5]. Diante disso, com o passar dos séculos, o homem tem observado a natureza e seus fenômenos, questionando e empenhando-se a compreende-los desde as suas origens e suas consequências. Por exemplo, ao observar o voo dos pássaros e partindo dos conhecimentos das ciências exatas aplicadas a engenharia, foram desenvolvidas as aeronaves. Da mesma maneira, ao observar as formigas, o homem percebeu que elas sempre tendem a encontrar o melhor caminho entre sua colônia e uma fonte de alimento, dando origem ao método de Colônia de Formigas.

A Otimização por Colônia de Formigas (OCF) é um método eficiente para resolver problemas de natureza combinatória, localizando rapidamente soluções com boa qualidade [3]. Este fenômeno, baseado de uma analogia biológica das formigas, permitiu a elaboração de um algoritmo computacional para resolver problemas gerais, como o PCV, estudado neste trabalho, sendo considerado relevante dentre os demais métodos heurísticos.

Este algoritmo foi desenvolvido pelo cientista da computação italiano Marco Dorigo, que ao observar o comportamento das formigas, percebeu que elas tendem a encontrar a menor rota entre o alimento e o formigueiro, devido a influência da liberação e captação do feromônio, substância volátil depositado pelas formigas ao longo do seu percurso, que permite a comunicação indireta entre elas. Ao encontrarem mantimento e retornar a colônia, a formiga libera feromônio ao longo da sua trilha, influenciando na escolha das demais formigas ao cruzarem com rotas mais fortes. Entretanto, o feromônio evapora ao longo do tempo, este evento foi observado durante o experimento da ponte binária realizado por Denebourg et al., 1990, para estudar o comportamento forrageiro das formigas [13] e ilustrado na Figura 2.

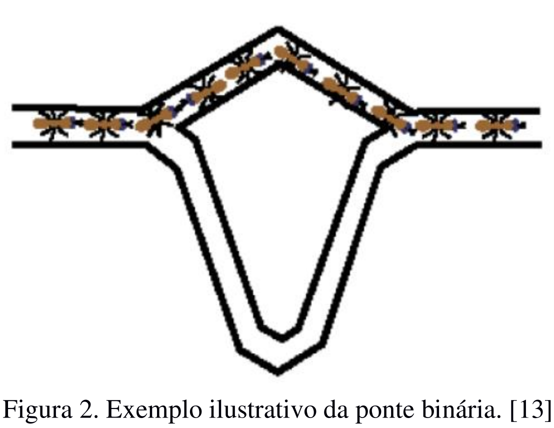

Figura 2. Exemplo ilustrativo da ponte binária. [13]

Inicialmente, no experimento da ponte binária, as formigas escolhem seu caminho sem a interferência do feromônio, escolhendo os dois caminhos com igual probabilidade. Após certo tempo, uma das pontes terá mais feromônio, atraindo as formigas com maior probabilidade, de tal forma que as formigas realizem sua rota pela ponte mais curta e em menos tempo.

A analogia de Dorigo entre o comportamento das formigas é uma abordagem de solução para um problema de otimização combinatória, implicando no mapeamento das formigas sobre qual caminho percorrer perante as decisões tomadas pela heurística probabilística. A probabilidade de selecionar determinadas rotas varia ao longo do tempo conforme a qualidade das decisões tomadas anteriormente. Para implementar um algoritmo OCF em um determinado problema, é preciso definir uma heurística satisfatória, que se preste à randomização e uma definição apropriada da rota [6]. Geralmente, esses algoritmos trabalham com uma população fixa de formigas e percorrem por um número de iterações, assim, na primeira iteração não há influência do feromônio na decisão da melhor rota, apenas dos custos envolvidos, assim, a formiga terá um conjunto de possíveis rotas com diferentes probabilidades de escolha. Após definir o seu percurso, a taxa de feromônio de cada rota é atualizado para iniciar uma nova iteração.

2.4. Aplicação do Algoritmo de Colônia de Formigas ao PCV

Seja $G = (V, A)$ um grafo orientado, onde V é o conjunto dos vértices $\{1, ..., m\}$ e A o conjunto das arestas $\{1,$

..., *n}*. Seja *K* o conjunto das formigas *{1, ..., f}*. A probabilidade de uma formiga $k \in \{K\}$ escolher um ponto $j \in V$ partindo de um ponto $i \in V$ é influenciado pela distância entre os pontos *i* e *j*, e da quantidade de feromônio presente entre esses pontos. Quanto maior for a distância entre o ponto de partida e o destino, menor a probabilidade de a formiga assumir este caminho, pois ela tende, com o passar do tempo, a escolher a rota de menor custo, o que representa uma relação inversa. E, quanto maior a quantidade de feromônio em uma rota, maior a probabilidade de a formiga escolher este caminho, representando uma relação direta. Considere as seguintes variáveis:

P_{ij}^{k} = Probabilidade da formiga *k* escolher a rota *ij*;
$T_{ij}^{(t)}$= Feromônio presente na aresta *ij* no tempo *t*;
η_{ij} = Variável relativa ao peso da distância referente à aresta *ij*;
α = Variável de controle do grau de influência do feromônio na escolha;
β = Variável de controle do grau de influência da rota na escolha.

Assim, ao estar em uma cidade, a probabilidade de a formiga escolher uma cidade não visitada é dada por:

$$P_{ij}^{k} = \frac{\left[T_{ij}^{(t)}\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}}{\sum_{l \in N_i^k}\left[T_{il}^{(t)}\right]^{\alpha}\left[\eta_{il}\right]^{\beta}} \tag{1}$$

onde N_i^ké o conjunto de cidades ainda não visitadas pela formiga k, $\eta_{ij} = 1 / d_{ij}$ é o inverso da distância entre os pontos *i* e *j* e os parâmetros α e β influenciam tanto na distância, quanto no feromônio das rotas e, por sua vez, no comportamento do algoritmo. Se α = 0, as probabilidades de seleção são iguais ao inverso da distância dos caminhos percorridos pela formiga *k*, assim, as cidades mais próximas tendem a serem escolhidas. Se β = 0, temos que há apenas a atualização do feromônio e, desta forma, todas as formigas tendem a construir a mesma solução.

Para definir o caminho percorrido por cada formiga, é utilizado o método da roleta. Este método consiste em uma analogia de a formiga escolher uma rota para trilhar com base na faixa de probabilidades de a formiga escolher a rota, ou seja, quanto maior for a probabilidade, maior a chance de escolha. O feromônio inicial de todas as arestas é uma constante. Após isso, o feromônio depositado pela formiga *k* após passar pela rota entre os pontos *i* e *j* é calculado com base na seguinte equação:

$$\Delta T_{ij}^{k} = \frac{Q}{d_k} \tag{2}$$

onde *Q* representa a constante de atualização do feromônio e d_k a distância total percorrida pela formiga *k*. A taxa de atualização do feromônio é calculado por:

$$T_{ij}^{(t)} = (1 - \rho)\, T_{ij}^{(t-1)} + \sum_{k=1}^{m} \Delta T_{ij}^{k\,(t)} \tag{3}$$

onde ρ é a taxa de evaporação do feromônio, $T_{ij}^{(t-1)}$representa o feromônio total da iteração anterior entre os pontos *i* e *j* e, o último termo representa o somatório do feromônio depositado por todas as formigas *k* que passam pela rota entre os pontos *i* e *j*.

Feitas todas as iterações, é possível observar que todas as formigas tendem a fazer o mesmo caminho, cumprindo com o objetivo do PCV. É importante destacar que as formigas podem convergir para um mesmo caminho obtendo-se uma solução convergente, porém sem garantia de otimalidade, mas nem sempre isso ocorre. Quando há um número maior de cidades e mesmo feitas muitas iterações, essa convergência pode não ocorrer. Desta forma, é necessário delimitar um número máximo de iterações para atingir um resultado satisfatório.

3. METODOLOGIA

Neste trabalho, será estudado a aplicação da meta-heurística de Colônia de Formigas adaptada as características do PCV contido em uma empresa distribuidora de laticínios localizada na cidade de Angicos/RN.

3.1. Estudo de Caso: Problema da APASA

A Associação dos Pequenos Agropecuaristas do Sertão de Angicos (APASA), efetua entregas de produtos laticínios, em veículo com compartimento refrigerado, diariamente em um conjunto de cidades do Estado do Rio

Grande do Norte. O problema propõe encontrar a melhor rota para realizar a distribuição dos produtos sobre um agrupamento de *n* cidades, a fim de minimizar os custos, seja ao utilizar a rota de menor distância ou a rota que totalize o menor tempo de viagem. Segundo [8], a empresa opera em doze cidades, incluindo o centro distribuidor e os consumidores. Para fins de simulação, o número de cidades foi ampliado em [9], mantendo o mesmo fornecedor, considere que faremos uso de um caminhão refrigerado com combustível e compartimento de carga infinitos. O Quadro 1 indica o total de cidades envolvidas no problema, totalizando 48 cidades.

Quadro 1 – Identificação e numeração das Cidades do RN.

Nº	Cidade	Nº	Cidade	Nº	Cidade
1	<u>ANGICOS</u>	17	São Gonçalo do Amarante	33	Monte Alegre
2	São Rafael	18	Pau dos Ferros	34	Nísia Floresta
3	Itajá	19	Parnamirim	35	Nova Cruz
4	Assú	20	Macaíba	36	Touros
5	Jucurutu	21	Macau	37	Tangará
6	Mossoró	22	João Câmara	38	São Paulo do Potengi
7	Natal	23	Ipanguaçu	39	Santo Antônio
8	Afonso Bezerra	24	Currais Novos	40	Taipu
9	Pedro Avelino	25	Ceará-Mirim	41	Upanema
10	Fernando Pedrosa	26	Canguaretama	42	Pedro Velho
11	Santana do Matos	27	Baraúna	43	Parelhas
12	Lajes	28	Apodi	44	Caraúbas
13	Santa Cruz	29	Acari	45	Alexandria
14	Caicó	30	Areia Branca	46	Jardim de Piranhas
15	São Miguel	31	Extremoz	47	São José do Campestre
16	São José do Mipibú	32	Goianinha	48	Poço Branco

Fonte: [9].

Segundo [9], o problema remete a abordagem do PCV, fazendo uso de um único transporte para cada situação, de forma a suprir a demanda total. Para o problema em questão, serão consideradas seis simulações distintas: uma única rota, com 48 cidades, e cinco sub-rotas com 36, 24, 12, 7 e 6 cidades, respectivamente. Com relação as variáveis de minimização da função objetivo, temos o custo da distância percorrida (Km) e o custo do tempo gasto (min), com os dados coletados através do Google Maps. A Figura 2 ilustra a dificuldade do problema, tendo localizadas no mapa todas as cidades. O ponto de partida e chegada é Angicos, identificada com o símbolo de localização e letra "A", as demais, estão identificadas apenas com o símbolo de localização.

Figura 3. Mapa que identifica a localização do problema [9].

O Quadro 2 identifica os problemas (instâncias), a unidade de medida a qual o problema trata para encontrar a melhor rota, o número de cidades *(n)* e, as cidades contidas em cada caso.

Quadro 2 – Instâncias.

Instância	Medida	n	Cidades pertencentes a instância (Nº - Quadro 1)
Problema 1	Km	48	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47 e 48.
Problema 2	Min		
Problema 3	Km	36	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35 e 36.
Problema 4	Min		
Problema 5	Km	24	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23 e 24.
Problema 6	Min		
Problema 7	Km	12	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 e 12.
Problema 8	Min		
Problema 9	Km	7	1, 7, 8, 9, 10, 11 e 12.
Problema 10	Min		
Problema 11	Km	6	1, 2, 3, 4, 5 e 6.
Problema 12	Min		

Fonte: [9].

Trabalhos correlatos abordam o mesmo PCV, [10] fez uso de um algoritmo genético tradicional (AGT) para encontrar soluções aproximadas dos resultados ótimos obtidos em [8] via método exato, e [9] apresentou via algoritmo memético (AM). A abordagem feita por [11] consiste em uma meta-heurística aproximativa, *Simulated Annealing*, comparando sua eficiência com os demais estudos. Nesse contexto, este artigo apresenta uma nova meta-heurística de aproximação, denominada Colônia de Formigas, e será empregado para verificar sua aplicabilidade e validar sua eficiência mediante os demais estudos prévios citados anteriormente.

3.2. Algoritmo de Colônia de Formigas

A seguir, apresentamos a Figura 4 que ilustra o funcionamento do algoritmo do método de Colônia de Formigas utilizado neste trabalho.

```
início do algoritmo
01. LEITURA dos dados iniciais;
02. DEFINIR ALFA, BETA, RHO, Q, lim_iter;
03. mtz_ferom = inicializa_matriz_feromonios ( );
04. melhorSol = infinito;
05. PARA i = 1 até lim_iter FAÇA
06.     ENQUANTO rota_nao_completa
07.             Calcula lista de Probabilidade de cada formiga;
08.             Escolhe próxima cidade via roleta;
09.             Atualiza custos para cada formiga;
10.     FIM ENQUANTO
11.     SE (min(Sol) < melhorSol);
12.         melhorSol = min(Sol);
13.     FIM SE
14.     mtz_ferom = atualiza_matriz_feromonios ( );
15. FIM PARA
fim do algoritmo
```

Figura 4. Algoritmo de Colônia de Formigas adaptado ao problema da APASA. (Autoria própria)

Na linha 01 temos a leitura dos dados do problema, como nome da instância e dados referentes aos custos (distância ou tempo de rota). Os hiperparâmetros são definidos na linha 02. A matriz de feromônios *mtz_ferom* é criada e atribuída com o feromônio inicial na linha 03. Na linha 04 definimos um valor infinito para a variável *melhorSol*, que irá armazenar a melhor solução encontrada durante a execução do algoritmo. O laço da linha 05 à linha 15 controla as iterações da formiga. O comando condicional da linha 06 à linha 10 é responsável pela construção da rota pela formiga. A probabilidade de escolha de cada cidade é definida pela equação (1). A escolha constante na linha 08 é realizada de acordo com uma roleta com área de escolha proporcional à probabilidade da rota. A cidade escolhida é inserida na rota e os custos são atualizados. As linhas 11 a 13 atualizam o valor da variável *melhorSol*. Na linha 14 a matriz de feromônios é atualizada com as melhores formigas.

Os hiperparâmetros são variáveis definidas manualmente pelo programador e que influenciam diretamente no resultado do algoritmo. No caso do algoritmo de Colônia de Formigas, temos os seguintes hiperparâmetros: α, β, ρ e *Q*, *feronômio inicial*. Para definir os valores de *Q* e o *feromônio inicial*, utilizamos valores empíricos baseados em resultados iniciais, sendo *Q* = 10 e *feromônio inicial* = 0.1.

Para os hiperparâmetros α, β e ρ, fizemos uma busca em grid (*grid search*) fazendo todas as possíveis combinações de valores consideradas para as variáveis. Os intervalos considerados para os problemas menores e maiores que 10 cidades foram:

$0.5 \leq \alpha \geq 1.5$, com amplitude de busca 0.2
$0.5 \leq \beta \geq 2.5$, com amplitude de busca 0.5
$0.05 \leq \rho \geq 0.2$, com amplitude de busca 0.05

Esses parâmetros foram calibrados nas instâncias 1 e 4, para os problemas menores e 3 e 6 para os problemas maiores. Foram executados 20 vezes cada instância com os valores pré-definidos. Ao fim do processo, adotamos os seguintes valores de α, β e ρ:

Tabela 1. Parâmetros utilizados pelo Algoritmo descrito pela Figura 4. (Autoria própria)

Problemas	**Parâmetros**
Problemas 1 a 6	α = 1,5
	β = 2,0
	ρ = 0,05
Problemas 7 a 12	α = 0,9
	β = 1,5
	ρ = 0,05

No decorrer deste trabalho, serão analisados os dados coletados do problema da APASA com o método das formigas. Para gerar uma base de soluções para análise, o algoritmo adaptado a este estudo realizou 20 rodadas com 200 e 300 iterações para os problemas menores e maiores, respectivamente. Foram gerados através do algoritmo os seguintes dados: a melhor formiga, responsável pela melhor solução encontrada, sendo o ponto de maior interesse deste trabalho; o total de vezes que a melhor solução foi encontrada; a média das melhores soluções detectadas a cada rodada e iteração; o tempo médio de solução; a melhor das melhores formigas finais e sua respectiva média.

O algoritmo foi executado em um *notebook* Dell com processador Intel® Core™ i5 de 7º geração, 8 GB de memória RAM e SSD de 222 GB. O sistema operacional é o *Windows 10 Home Single Language*, versão 21H2 e o ambiente computacional é o OCTAVE GNU, versão 6.3.0.

4. RESULTADOS E DISCUSSÕES

Nesta seção, serão apresentados os resultados gerados através dos experimentos computacionais, com análise geral e por partes da meta-heurística abordada e comparativa com os demais trabalhos deste problema.

4.1. Resultados Encontrados

A seguir, apresentamos as Tabelas 2, 3 e 4 que ilustram os dados obtidos pelo algoritmo do método de Colônia de Formigas utilizado neste trabalho, onde na primeira tabela é realizado um comparativo desses resultados com o *solver* GLPK.

Na Tabela 2, a primeira coluna tem o nome da instância, a segunda coluna o número de cidades, a terceira coluna a solução encontrada pelo *solver* GLPK no trabalho de [9], seguida do tempo utilizado pelo *solver* para encontrar a solução, que foi limitado em 80.000 segundos. Soluções marcadas com * não obtiveram garantia de otimalidade no tempo limite. A quinta coluna informa a melhor solução encontrada pelo algoritmo de Colônia de Formigas seguida da média de tempo de processamento computacional em cada rodada. Destacamos que como os ambientes computacionais são diferentes, o tempo não é comparável. Por fim, apresentamos a diferença percentual entre as soluções encontradas, dada pela fórmula (4). Valores negativos indicam que nosso algoritmo conseguiu encontrar melhor solução que o *solver*, enquanto valores positivos indicam melhor solução por parte do *solver* e valores nulos indicam que as estratégias encontraram a mesma solução.

$$\boldsymbol{D}\ (\%) = \frac{\boldsymbol{Sol.Mín.} - \boldsymbol{Sol.GLPK}}{\boldsymbol{Sol.GLPK}}\ \boldsymbol{x\ 100} \tag{4}$$

Para fins de comparação, [9] utilizou o modelo determinístico no GLPK – GNU linear programming kit, de forma que para os problemas 03, 04, 05 e 06 não houve garantia de solução ótima em até 80000 segundos de processamento, deixando GAP de 5,8; 4,1; 5,8 e 4,5; respectivamente. Segundo [12], GAP é o desvio percentual do valor apresentado a partir de um limite global para a solução exata calculada pelo GLPK.

Tabela 2. Comparativo da solução encontrada com o Algoritmo de Colônia de Formigas e o GLPK. (Autoria própria)

Nome	n	Sol. GLPK	T(s) GLPK	Sol.Mín.	T(s)	D (%)
Problema 1	48	**1942,30**	6737,7	1953,40	389,5594	0,571487412
Problema 2		**1973,00**	5708,5	1977,00	354,9902	0,202736949
Problema 3	36	1719,20*	86325,3	**1705,10**	165,8902	-0,820148906
Problema 4		1676,00*	85204,0	**1671,00**	152,2530	-0,298329356
Problema 5	24	1339,90*	85439,0	**1327,80**	50,4806	-0,903052467
Problema 6		**1223,00***	73266,0	**1223,00**	50,6344	0
Problema 7	12	**672,70**	0,1	**672,70**	5,6237	0
Problema 8		**606,00**	0,1	**606,00**	5,6416	0
Problema 9	7	**438,30**	0	**438,30**	1,5647	0
Problema 10		**364,00**	0	**364,00**	1,5661	0
Problema 11	6	**344,90**	0	**344,90**	1,1145	0
Problema 12		**305,00**	0	**305,00**	1,0939	0

Ao observar a Tabela 2, é possível visualizar que o algoritmo de formigas encontrou boas soluções em um tempo computacional aceitável e inferior ao tempo gasto pelo GLPK, de forma que não se deve comparar ambos os períodos, apenas as soluções encontradas, pois fazem uso de ambientes computacionais e métodos distintos.

No que diz respeito aos resultados, foi possível superar apenas 3 das 12 instâncias, especificamente nos problemas 3, 4 e 5. É importante destacar que apesar de terem sido encontradas boas soluções, melhores que as obtidas pelo GLPK para estas instâncias, não há garantia da otimalidade destas. Além disso, para o problema 6, o método encontrou a mesma solução gerada pelo GLPK, indicando que existe uma probabilidade de esta ser uma solução com otimalidade. Para os problemas menores, foram obtidos os mesmos resultados do GLPK, de modo que os dados se igualam dos obtidos por [9] e sem superação pelo método. Na última coluna da Tabela 2 é possível perceber o grau de eficiência do algoritmo descrito na Figura 4.

As Tabelas 3 e 4 apresentam os melhores resultados obtidos pelo método das formigas para os problemas menores e maiores, respectivamente. Para cada coluna, estão descritos os seguintes dados: o problema (Nome), o número de cidades *(n)* utilizadas na resolução do problema, a melhor solução encontrada (Sol.Mín), a quantidade de vezes que esta solução foi encontrada (Vezes), a média das melhores soluções encontradas (Média), o tempo médio gasto pelo algoritmo em cada rodada (em segundos), a melhor das melhores formigas finais (M.M.F.F) e a média das melhores formigas finais (Méd.M.F.F).

Tabela 3. Resultados obtidos para os problemas menores. (Autoria própria)

Nome	n	Sol.Mín	Vezes	Média	T(s)	M.M.F.F	Méd.M.F.F
Problema 7	12	672,70	18	672,72	5,6237	672,90	733,95
Problema 8		606,00	18	606,10	5,6416	613,00	649,50
Problema 9	7	438,30	20	438,30	1,5647	438,30	454,26
Problema 10		364,00	20	364,00	1,5661	364,00	377,15
Problema 11	6	344,90	20	344,90	1,1145	348,30	358,50
Problema 12		305,00	20	305,00	1,0939	305,00	311,65

A solução mínima foi encontrada em todas as instâncias menores em concordância com as obtidas pelo *solver* GLPK, conforme apresentadas na Tabela 2. Nos problemas 7 e 8 a menor solução foi obtida em 18/20 rodadas realizadas, desta forma a média das melhores soluções está bastante próxima da melhor solução, o que indica que o método conseguiu efetuar sua busca com boa eficiência ao longo das 12 cidades envolvendo o percurso em (Km) e o tempo em (minutos), respectivamente. Nas demais instâncias, a solução mínima é equivalente à média das melhores soluções, pois o algoritmo conseguiu encontrar em 20/20 rodadas feitas a mesma solução, tais resultados gerados também são os mesmos dos obtidos pelo *solver* GLPK.

O tempo de processamento computacional gasto foi mínimo, devido a natureza dos problemas. Desta forma, a solução mínima demorou pouco mais de 5 segundos em problemas com 12 cidades e pouco mais de 1 segundo para as instâncias com 7 e 6 cidades.

Na penúltima coluna, estão as melhores das melhores formigas finais, ou seja, contempla das soluções obtidas pelas formigas finais ao completarem seu percurso, apresentando a melhor dentre as obtidas em cada rodada e iteração realizada pelo algoritmo. É possível notar que em todas as instâncias as formigas reduzem seu espaço de busca ao máximo, estando próximas da melhor solução, porém algumas se prendem em regiões entre algumas cidades, e não conseguem diversificar este espaço de busca para obter a solução mínima, como nas instâncias 9, 10 e 12.

A última coluna indica a média das melhores formigas finais, essa média é calculada com base em todas as soluções de custo das melhores formigas finais com o menor resultado e, os dados indicam que ao aumentar o problema, as formigas tendem a ter mais dificuldade, influenciando em todos os parâmetros da Tabela 3. Para os problemas menores, os dados estão próximos da região de boa otimalidade dentro do conjunto de soluções.

Tabela 4. Resultados obtidos para os problemas maiores. (Autoria própria)

Nome	n	Sol.Mín	Vezes	Média	T(s)	M.M.F.F	Méd.M.F.F
Problema 1	48	1953,40	1	1971,33	389,5594	2012,10	2048,63
Problema 2		1977,00	8	1982,25	354,9902	1990,00	2025,10
Problema 3	36	1705,10	1	1722,85	165,8902	1735,80	1776,61
Problema 4		1671,00	1	1675,85	152,2530	1691,00	1725,65
Problema 5	24	1327,80	1	1339,99	50,4806	1343,70	1387,88
Problema 6		1223,00	2	1231,45	50,6344	1238,00	1269,25

Na Tabela 4, temos os resultados obtidos para com os problemas maiores. Pode-se notar que a solução mínima não é encontrada em todas as instâncias, conforme apresentado pela Tabela 2 e, os resultados são obtidos com certa dificuldade em um tempo de processamento maior. A maioria dos problemas encontrou apenas uma vez a menor rota, com exceção de 2 e 6 que encontraram 8 e 2 vezes respectivamente, de tal forma que possa ter tido uma relação com a natureza dos dados de tempo de uma cidade i para j. O tempo de processamento computacional foi em média de 6.21 minutos, 2.65 minutos e 50.56 segundos para os problemas com 48, 36 e 24 cidades, nesta ordem.

Por fim, as últimas colunas indicam a melhor solução entre as formigas finais e a média para as 20 rodadas. Logo, é possível perceber que a melhor das melhores formigas finais obteve resultado pouco próximo do espaço com a solução mínima encontrada pelo algoritmo, indicando que há eficiência na redução do espaço de busca amostral, mesmo que limitado, devido a natureza do ambiente de programação.

Assim, é possível compreender pelas Tabelas 3 e 4 que quanto maior a instância, menor é a eficiência do algoritmo na busca pelas soluções com boa otimalidade e maior é o tempo de processamento computacional para obter uma solução mínima perante as demais encontradas ao findar as 200/300 iterações e 20 rodadas estabelecidas inicialmente como restrição de busca.

4.2. Estado da Arte para o Problema

A Tabela 5 apresenta a comparação dos resultados conquistados com o método de Colônia de Formigas através do ambiente computacional, OCTAVE GNU, com outros trabalhos deste mesmo estudo de caso contidos na literatura, que fez uso de métodos meta-heurísticos, como o Algoritmo Genético e os Algoritmos Meméticos 1 e 2 [9], e do *Simulated Annealing* [11].

Tabela 5. Comparativo dos dados obtidos com os demais trabalhos na literatura. (Autoria própria)

Nome	n	Sol. GLPK	Sol.Mín (AGP e AMs)	Sol.Mín (S.A.)	Sol.Mín (ACF)
Problema 1	48	**1942,30**	2070,80	2951,20	1953,40
Problema 2		**1973,00**	2101,00	2740,00	1977,00
Problema 3	36	1719,20*	1705,60	2134,90	**1705,10**
Problema 4		1676,00*	**1668,00**	1979,00	1671,00
Problema 5	24	1339,90*	**1321,00**	1351,00	1327,80
Problema 6		**1223,00***	**1223,00**	1256,00	**1223,00**
Problema 7	12	**672,70**	**672,70**	**672,70**	**672,70**
Problema 8		**606,00**	**606,00**	**606,00**	**606,00**
Problema 9	7	**438,30**	**438,30**	**438,30**	**438,30**
Problema 10		**364,00**	**364,00**	**364,00**	**364,00**
Problema 11	6	**344,90**	**344,90**	**344,90**	**344,90**
Problema 12		**305,00**	**305,00**	**305,00**	**305,00**

Para avaliar a eficiência de todos os experimentos computacionais com os algoritmos meta-heurísticos, foram extraídos os melhores resultados para análise, descritos na Tabela 5. Vale ressaltar que, para este trabalho o tempo não é comparável, apenas os resultados gerados para obter o menor custo, em virtude de os resultados terem sido confeccionados em ambientes computacionais distintos.

Em relação aos problemas menores, o algoritmo de Colônia de Formigas, juntamente com os demais algoritmos foram eficientes, e conquistaram resultados análogos aos obtidos pelo GLPK, não havendo ganhos de resultados em comparação as demais meta-heurísticas. Para os problemas maiores, o algoritmo de Colônia de Formigas ganhou em algumas instâncias e perdeu em outras, em relação aos demais trabalhos. É possível observar através da Tabela 5 que para os problemas 04 e 05 os dados gerados não conseguiram ser menores em comparação com os encontrados por algoritmos meméticos, mas que atingiram superioridade de resolução quando comparado ao

Simulated Annealing. Contudo, para o problema 3, houve ganho de resultado e no problema 6 houve similaridade de solução com [9].

Em relação aos problemas maiores 3/6 instâncias foram superadas de [9] e 6/6 instâncias foram superadas de [11] e, para os problemas menores, todas as soluções foram equivalentes das obtidas pelo *solver* GLPK e pelos trabalhos passados.

Para corroborar a análise da eficiência da meta-heurística proposta neste artigo, foram realizados um total de 400 experimentos computacionais para coleta e comparação dos melhores resultados obtidos, buscando por regiões propícias com as melhores soluções conforme os parâmetros utilizados no estudo. De modo geral, o algoritmo se mostrou promissor de acordo com a análise comparativa realizada entre o método das formigas, os demais métodos e o exato, de maneira que o algoritmo foi eficiente na sua busca pelos menores resultados em um intervalo de tempo considerável e que, para alguns casos dos problemas maiores, foram preferíveis que o método exato.

5. CONSIDERAÇÕES FINAIS

O Problema do Caixeiro Viajante (PCV) é um dos problemas da otimização combinatória com aplicação em diversos problemas reais, sendo uma de suas aplicações o Roteamento de Veículos. Desta forma, a depender do tipo de problema e do seu tamanho, são necessários métodos de resolução que faça bom uso dos dados existentes. E, através da análise comparativa, foi possível perceber que o método de Colônia de Formigas é eficiente na obtenção de boas soluções para o problema de otimização estudado, tanto na qualidade das rotas encontradas, quanto no tempo de processamento.

Neste artigo, o objeto de estudo foi o problema da APASA expandido a 48 cidades na busca pelas melhores rotas a cada caso, e foi possível concluir que a meta-heurística de Colônia de Formigas é uma alternativa eficiente na busca por soluções, com soluções viáveis e com boa otimalidade do que as obtidas por outros métodos contidas na literatura em um tempo de processamento computacional aceitável.

Uma condição observada durante este estudo foi a dependência no ajuste dos parâmetros, os quais determinam por melhores reduções do espaço de busca pelas soluções mínimas, fazendo com que os algoritmos sejam sensíveis a cada configuração imposta à equação das formigas. Vale lembrar que esses parâmetros foram selecionados através da escolha e busca por regiões propícias aos melhores resultados. Logo, é possível aperfeiçoar estes resultados para obter novas soluções fazendo uso dos parâmetros ideais, realizando seu estudo com base neste problema específico.

Diante disso, outros estudos podem ser feitos para gerar a melhor solução, além do estudo dos parâmetros, podemos aperfeiçoar o algoritmo idealizado para este trabalho, reduzindo possíveis laços, reduzindo seu tempo para execução; usar novas alternativas de uso das formigas; buscar por novas fontes de trabalhos na literatura para auxiliar na complementação e/ou melhoria do processo.

Sendo assim, o trabalho conseguiu atingir seus objetivos ao formular, aplicar, autenticar e apresentar os dados gerados pelo método de Colônia de Formigas para resolver o PCV da APASA contido na literatura. O algoritmo retornou resultados promissores, com possibilidade de melhoria para serem feitas em novos trabalhos.

6. REFERÊNCIAS BIBLIOGRÁFICAS

[1] FREDERICK S. H.; GERALD J. L. Introdução à pesquisa operacional. Tradução: Ariovaldo Griesi; revisão técnica João Chang Junior. Título original: Introduction operations research. São Paulo: McGraw-Hill, 2006.

[2] ANDRADE, E. L. Introdução à pesquisa operacional: métodos e modelos para análise de decisões. 5. ed. Rio de Janeiro: LTC, 2018.

[3] DORIGO, M. and STUTZLE, T. The And Colony Optimization Metaheuristic: Algorithms, Applications, and Advances, Handbook of Metaheuristics, v. 57, p. 250 – 285, 2002.

[4] GOLDBARG, M. C.; LUNA, H. P. Otimização combinatória e programação linear: modelos e algoritmos. 2a ed. Rio de Janeiro: Editora Elsevier, 2005.

[5] CHÂTELET, F. Lagos e Práxis. Rio de Janeiro: Paz e Terra, 1972.

[6] DOWSLAND, K. A. and THOMPSON J. M. An improved ant colony optimization heuristic for graph colouring. Discrete Applied Mathematics, v. 156, p. 313-324, may. 2007.

[7] KANDA, J. Y.; et al. Selection of algorithms to solve traveling salesman problems using meta-learning. International Journal of Hybrid Intelligent Systems, vol. 8, no. 3, p. 117–128, 2011.

[8] 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.

[9] FERNANDES, F. R. S. Metaheurísticas Evolutivas aplicadas ao Problema de Roteamento de Veículos em uma empresa de lacticínios no interior do RN: uma abordagem via algoritmos genético e memético. 85 f. Monografia (Graduação em Sistemas de Informação) – Universidade Federal Rural do Semi-Árido. Angicos, 2016.

[10] 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.

[11] GÊ, M. C. O. Simulated annealing aplicado ao problema do caixeiro viajante: um estudo de caso. – Universidade Federal Rural do Semi-Árido – Centro de Ciências Exatas e Naturais. Trabalho de Conclusão de Curso. Mossoró, 2018.
[12] MENEZES, M. S. O problema do Caixeiro alugador com coleta de prêmios: um estudo algorítmico. 126f. 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.
[13] FELIX, T. S. Chan and Manoj Kumar Tiwari. Swarm Intelligence, Focus on Ant and Particle Swarm Optimization. InTech, 2007.

7. ANEXOS

ANEXO I

Para exemplificarmos o comportamento da meta-heurística de Colônia de Formigas aplicada ao Problema do Caixeiro Viajante, tomemos o exemplo da Figura 1.

Tabela 6. Custos entre cada vértice do grafo da Figura 1. (Autoria própria)

Cidades	**A**	**B**	**C**	**D**	**E**
A	0	28	51	53	49
B	28	0	39	32	30
C	51	39	0	26	41
D	53	32	26	0	34
E	49	30	41	34	0

Considere os seguintes parâmetros: $\alpha = 1$; $\beta = 1$; $\rho = 0{,}01$; $Q = 10$ e *feromônio inicial* = 0,1.
Na 1º iteração as formigas ainda não iniciaram a busca, logo não há influência do feromônio na decisão da melhor rota, somente da distância. Assim, utilizando a equação (1), vamos construir a tabela de probabilidades referente a formiga 1 (F_1) inserida em A.

Tabela 7. Probabilidades da formiga 1 para cada vértice do grafo. (Autoria própria)

Formiga	**Rota**	$\mathbf{d}_{ij}$	$\boldsymbol{\eta}_{ij}$	$\mathbf{T}_{ij}$	$\boldsymbol{\eta}_{ij}\mathbf{T}_{ij}$	P^k_{ij}	P^k_{ij} %
F_1	A – B	28	0,035714	0,1	0,003571	0,377537	37,7537
F_1	A – C	51	0,019608	0,1	0,001961	0,207275	20,7275
F_1	A – D	53	0,018868	0,1	0,001887	0,199453	19,9453
F_1	A – E	49	0,020408	0,1	0,002041	0,215735	21,5735

Pelo método da roleta, a formiga escolhe sua próxima cidade ainda não visitada e atualiza a probabilidade para as demais ainda não visitadas. Lembrando que quanto maior a probabilidade, maior a tendencia da formiga escolher, contudo ela pode optar por qualquer uma das disponíveis.

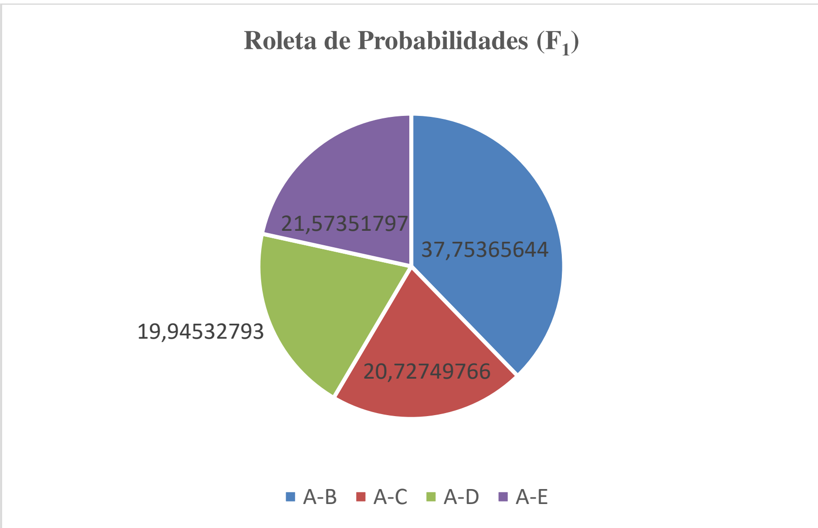

Figura 5. Roleta de probabilidades de F_1 em A.

Realizado o cálculo de probabilidades e o método da roleta para todas as formigas, temos: