

UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE – UERN

FACULDADE DE CIÊNCIAS EXATAS E NATURAIS – FANAT

DEPARTAMENTO DE INFORMÁTICA – DI

Caio César de Freitas Dantas

**Estratégias de Uso da Metaheurística Colônia de Formigas Para o Problema do
Caixeiro Viajante**

MOSSORÓ - RN

2017

Caio César de Freitas Dantas

**Estratégias de Uso da Metaheurística Colônia de Formigas Para o Problema do
Caixeiro Viajante**

Monografia apresentada à Universidade do Estado do Rio Grande do Norte como um dos pré-requisitos para obtenção do grau de bacharel em Ciência da Computação, sob orientação do Prof. Dr. Dario José Aloise.

MOSSORÓ - RN

2017

Ficha catalográfica gerada pelo Sistema Integrado de Bibliotecas
e Diretoria de Informatização (DINF) - UERN,
com os dados fornecidos pelo(a) autor(a)

D192e Dantas, Caio César de Freitas.
Estratégias de Uso da Metaheurística Colônia de Formigas Para o
Problema do Caixeiro Viajante / Caio César de Freitas Dantas - 2017.
42 p.

Orientador: Dario José Aloise.
Coorientadora: .
Monografia (Graduação) - Universidade do Estado do Rio Grande do
Norte, Ciência da Computação, 2017.

1. Metaheurística. 2. Problema do Caixeiro Viajante. 3. Otimização
Combinatória. 4. Colônia de Formigas. I. Aloise, Dario José, orient. II.
Título.

Caio César De Freitas Dantas

ESTRATÉGIAS DE USO DA METAHLURÍSTICA COLÔNIA DE FORMIGAS PARA O PROBLEMA DO CAIXEIRO VIAJANTE

Monografia apresentada como pré-requisito para a obtenção do título de Bacharel em Ciência da Computação da Universidade do Estado do Rio Grande do Norte – UERN, submetida à aprovação da banca examinadora composta pelos seguintes membros:

Aprovada em: 02/05/2017

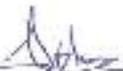
Banca Examinadora



Prof. Dr. DÁRIO JOSÉ ALOISE
Universidade do Estado do Rio Grande do Norte - UERN



Prof. Dr. FRANCISCO CHAGAS DE LIMA JÚNIOR
Universidade do Estado do Rio Grande do Norte - UERN



Prof. Dr. CARLOS HEITOR PEREIRA LIBERALINO
Universidade do Estado do Rio Grande do Norte - UERN

*Dedico este trabalho aos meus pais,
Davi Batista Dantas e Geiza Souza de
Freitas, que sempre me apoiaram e
acreditaram em mim.*

AGRADECIMENTOS

Agradeço a Deus por me dá força e saúde para conseguir superar as dificuldades. A minha família por sempre me apoiar e acreditar em meu potencial, em especial aos meus pais, Davi e Geiza que sempre me incentivaram e investiram em meus estudos. Aos meus amigos da graduação pela força e cumplicidade durante todo o tempo, Ao colega Aldemário, pela ajuda na implementação das Metaheurística. Gostaria de agradecer também a todos os Professores do DI pelos conselhos e ensinamentos. Em especial ao meu Professor, Orientador Dario José Aloise, por acreditar em meu potencial, pelo conhecimento transmitido e sempre com extrema dedicação e atenção.

RESUMO

Este trabalho apresenta uma metodologia baseada em variações da Metaheurística *Ant Colony Optimization* (ACO) para resolver o Problema do caixeiro viajante (PCV). Por se tratar de um problema de Otimização Combinatória NP-Difícil, algoritmos de soluções exatas se mostram ineficientes. Neste caso é necessário a utilização de métodos heurísticos para obter uma solução eventualmente ótima em um tempo computacional viável. Foram implementados três diferentes estratégias, com características específicas, baseadas na Metaheurística ACO. A metodologia desenvolvida foi aplicada na resolução de instâncias da biblioteca TSPLIB (*Library of Traveling Salesman Problems*). Os resultados apresentados pelos experimentos computacionais das três estratégias, foram comparados e avaliados, mostrando a viabilidade da utilização das estratégias.

Palavras-chave: Metaheurística ACO, Problema do Caixeiro Viajante (PCV), Otimização Combinatória.

ABSTRACT

This work presents a methodology based on variations of the Metaheuristic Ant Colony Optimization (ACO) to solve the traveling salesman Problem (PCV). It is a problem of Combinatorial Optimization NP-Hard, algorithms for exact solution, prove ineffective. In this case it is necessary the use of heuristic methods to obtain a solution, possibly optimal in a computational time feasible. Were implemented for three different strategies, with specific characteristics, based on the Metaheuristic ACO. The developed methodology was applied in solving instances of the library TSPLIB (Library of Traveling Salesman Problems). The results shown by the experiments the computing of the three strategies, were compared and evaluated, showing the feasibility of the use of the strategies.

Keywords: Metaheuristic ACO, Traveling Salesman Problem (TSP), Combinatorial Optimization.

LISTA DE SIGLAS

ACO	<i>Ant Colony Optimization</i>
AS	<i>Ant System</i>
PCV	Problema do Caixeiro Viajante
TSPLIB	<i>Library of Traveling Salesman Problems</i>
TSP	<i>Traveling Salesman Problems</i>

LISTA DE FIGURAS

Figura 2.1: Comportamento das formigas	16
Figura 2.2: Pseudocódigo Metaheurística ACO	18
Figura 2.3: Formulação para o Cálculo da Atratividade	19
Figura 2.4: Fluxograma da Metaheurística ACO	21
Figura 2.5: <i>The Icosian Game</i>	23
Figura 3.1: Pseudocódigo Estratégia 1	27
Figura 3.2: Pseudocódigo Estratégia 2	28
Figura 3.3: Pseudocódigo Estratégia 3	29
Figura 3.1: Gráfico de comparação das médias	32

LISTA DE TABELAS

Tabela 2.1: Aplicações do ACO em Problemas de Otimização Combinatória	16
Tabela 2.2: Aplicações do ACO em problemas reais	17
Tabela 3.1: Instâncias Utilizadas	26
Tabela 4.1: Resultado Estratégia 1	30
Tabela 4.2: Resultado Estratégia 2	30
Tabela 4.3: Resultado Estratégia 3	31
Tabela 4.4: Comparação da média dos resultados	31
Tabela 4.5: Comparação das estratégias com o ótimo global	32
Tabela 4.6: Diferença das estratégias para o ótimo global %	33
Tabela 4.7: Comparação do tempo das três estratégias	33

SUMÁRIO

1 INTRODUÇÃO	12
1.1 Objetivos Gerais	14
1.2 Objetivos Específicos	14
1.3 Organização do Trabalho	14
2 REFERENCIAL TEORICO	15
2.1 Metaheurística ACO	15
2.1.1 Origem	15
2.1.2 Características e Aplicações	16
2.1.3 Formulação do ACO	19
2.1.4 Estratégia Elitista	22
2.1.5 Estratégia ASrank	22
2.2 Circuito Hamiltoniano	22
2.3 O Problema do Caixeiro Viajante (PCV)	24
2.3.1 Formulação do PCV	24
3 METODOLOGIA	26
3.1 Estratégia 1	27
3.2 Estratégia 2	28
3.3 Estratégia 3	29
4 RESULTADOS	30
4.1 Comparação dos Resultados	31
5 CONCLUSÃO E TRABALHOS FUTUROS	34
6 REFERÊNCIAS	35
ANEXOS	37

1 INTRODUÇÃO

A Ciência da Computação depara-se com problemas que são improváveis de resolução por algoritmos exatos, por exemplo, os problemas de Otimização Combinatória NP-Difíceis, Karp (1972). São problemas para os quais não existem algoritmos que os resolvam em tempo polinomial, a menos que $P=NP$. Logo, métodos heurísticos vem sendo frequentemente utilizados para a solução deste tipo de problema em situações reais.

Os métodos heurísticos são utilizados para encontrar uma solução de boa qualidade, próxima do ótimo, não necessariamente a melhor, em um tempo computacional aceitável. Normalmente são aplicados a problemas em que a solução ótima é computacionalmente custosa quando realizada por meio de métodos exatos (CARABETTI, 2010).

Segundo Diaz et al. (1996), diversos fatores tornam interessante a utilização de algoritmos heurísticos na resolução de um problema específico:

- Inexistência de um método exato para resolução do problema ou um consumo excessivo do tempo de processamento
- Quando a solução ótima é desnecessária, visto que as soluções obtidas são eficientes;
- Quando limitações de tempo e/ou dinheiro obriguem a utilização de métodos de resposta rápida;

Os métodos heurísticos podem ser divididas em heurísticas construtivas, de refinamento e Metaheurística. De acordo com Cordenonsi (2008), as Metaheurística são utilizadas para contornar os problemas encontrados pelas heurísticas clássicas, pois guiam estas heurísticas para escapar de ótimos locais prematuros. Embora, assim como as demais heurísticas, os parâmetros e critérios dos algoritmos devem ser analisados com atenção para que possa gerar soluções viáveis e de boa qualidade em um tempo computacional razoável.

Conforme Ribeiro (1996):

“As metaheurísticas são procedimentos destinados a encontrar uma boa solução, eventualmente a ótima, consistindo na aplicação, em cada passo, de uma heurística subordinada, a qual tem que ser modelada para cada problema específico.”

Um dos problemas pertencente a classe dos NP-Difíceis é o do Ciclo Hamiltoniano. Este procura um caminho em um grafo $G(V, E)$ direcionado, onde passa por todos os vértices uma única vez, voltando ao vértice inicial. O problema do Caixeiro Viajante é o problema de Otimização que lhe é associado.

O Problema do Caixeiro Viajante (PCV) é descrito por um conjunto de n cidades e uma matriz de distância entre elas, com o objetivo do caixeiro viajante sair de uma cidade chamada origem, visitar cada uma das $n - 1$ cidades restantes uma única vez e retornar à cidade origem percorrendo a menor distância possível. Ou seja, deve ser encontrada um circuito (ciclo hamiltoniano) de custo mínimo, passando uma única vez por cada cidade. A quantidade de ciclos Hamiltoniano é da ordem de $O(n!)$.

Para resolver problemas dessa complexidade, é necessário o uso de Metaheurística. Neste trabalho, faremos uso da Metaheurística Colônia de Formigas sobre o PCV. Esta Metaheurística apresentada por Dorigo et al. (1992, 1996, 1999) é inspirada em comportamento de busca e exploração por alimentos das formigas reais. Esse comportamento permite que as formigas ao encontrar caminhos mais curtos entre as fontes de alimento e seu ninho e vice-versa, depositem feromônio na trilha. Quando decidem sobre a direção para onde ir, elas escolhem caminhos com maior probabilidade de ser o menor, marcados pela concentração de feromônio.

Neste trabalho, Passamos a investigar estratégias de uso dessa Metaheurística para encontrar o circuito hamiltoniano de menor custo em um grafo, buscando pelo método científico proceder experimentos computacionais e verificação de resultados.

1.1 Objetivo Geral

O objetivo deste trabalho é implementar e investigar diferentes estratégias de uso da Metaheurística Colônia de Formigas para encontrar o Circuito Hamiltoniano de menor custo em um grafo, com foco no problema do Caixeiro Viajante.

1.2 Objetivos Específicos

- Estudar a Metaheurística ACO como solução para o problema do Caixeiro Viajante.
- Implementar 3 (três) diferentes estratégias da Metaheurística ACO.
- Realizar testes com instâncias do problema encontradas na literatura.
- Analisar e comparar a performance das estratégias da Metaheurística ACO implementadas.

1.3 Organização do Trabalho

Este trabalho está organizado da seguinte forma: No capítulo 2 são apresentadas as fundamentações teóricas sobre o tema proposto. O capítulo 3 aborda a metodologia do trabalho e as estratégias desenvolvidas. No capítulo 4 são descritos os resultados das estratégias e a comparação entre elas. Por fim, o capítulo 5 apresenta as considerações finais e trabalhos futuros.

2 REFERENCIAL TEORICO

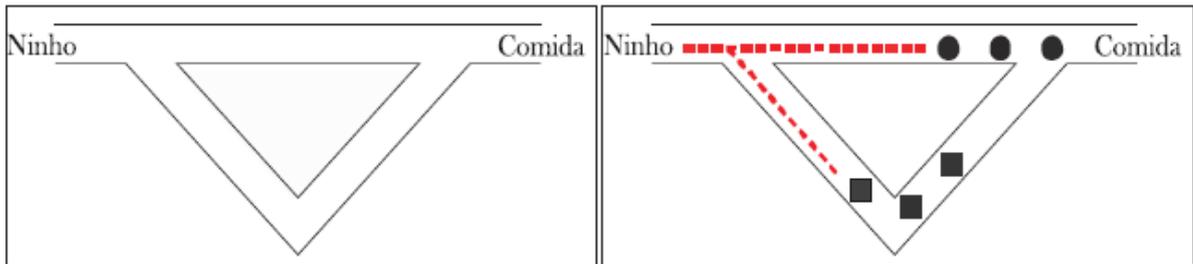
Este capítulo apresenta a base teórica para o desenvolvimento deste trabalho, sendo dividido em 3 tópicos principais, Metaheurística ACO, Circuito Hamiltoniano e o Problema do Caixeiro Viajante. Onde são descritos a origem, conceitos, aplicações, formulações e exemplos.

2.1 Metaheurística ACO

2.1.1 Origem

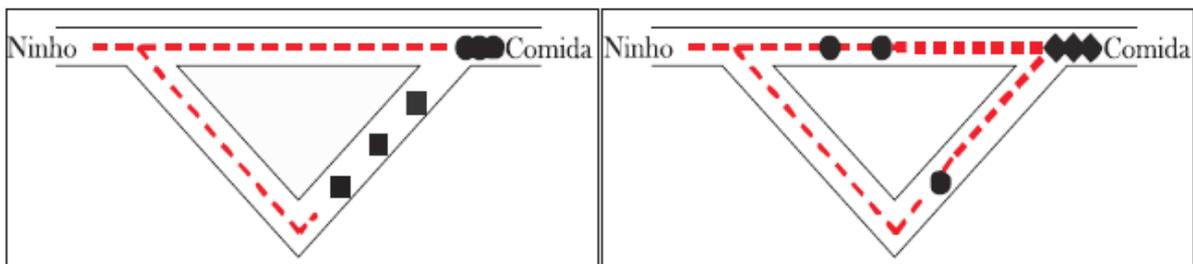
Otimização com Colônia de Formiga ou Ant Colony Optimization (ACO) é uma técnica de otimização que foi introduzida por Marco Dorigo no início da década de 1990. Esta Metaheurística foi inspirada na observação de colônias de formigas. O comportamento responsável pela inspiração para o desenvolvimento do ACO é a busca por alimento, em particular, como elas podem encontrar os caminhos mais curtos entre o alimento e o seu ninho. Ao buscar alimento, as formigas inicialmente exploram de forma aleatória a área em torno do seu ninho e enquanto caminham deixam uma trilha de feromônio químico no chão. As formigas tendem a escolher caminhos com maior concentração de feromônio. Quando encontra o alimento, avalia a qualidade e quantidade, e retorna para o ninho deixando uma trilha de feromônio que guiará as outras formigas para a fonte do alimento (BLUM, 2005).

Quando uma formiga tem que tomar uma decisão sobre qual caminho seguir, tende a procurar o caminho com maior quantidade de feromônio, deixando assim a trilha ainda mais atraente, fazendo com que o número de formigas seguindo esse caminho seja maior, como mostra a figura abaixo.

Figura 2.1: Comportamento das formigas

(a) Todas as formigas estão no ninho. Não há feromônio no ambiente.

(b) A busca se inicia. Em probabilidade, 50% das formigas vão para o caminho menor (simbolizado pelos círculos), e 50% tomam o caminho maior para a fonte de alimento (simbolizado pelos losangos).



(c) As formigas que tomaram o menor caminho chegaram mais cedo ao ninho. Portanto, quando retornarem, a probabilidade de tomar o menor caminho novamente é alta.

(d) A trilha de feromônio no menor caminho recebe, em probabilidade, um reforço mais forte, e a probabilidade de tomar esse caminho aumenta.

Fonte: Blum (2005)

2.1.2 Características e Aplicações

Atualmente ACO se caracteriza como uma ferramenta de otimização poderosa e versátil, com um número crescente de publicações e aplicações em diversas áreas da pesquisa operacional, gestão e tecnologia, segundo Gutjahr (2007). A Tabela abaixo mostra aplicações de Algoritmos de Colônias de Formigas a diversos problemas de otimização combinatória.

Tabela 2.1: Aplicações do ACO em Problemas de Otimização Combinatória

Problemas	Autores
Problema do Caixeiro Viajante (PCV)	<ul style="list-style-type: none"> • Dorigo, Maniezzo, e Colorni (Dorigo et al., 1991), (Dorigo, 1992), (Dorigo et al., 1996) • Dorigo e Gambardella (Dorigo e Gambardella, 1997), • Stützle e Hoos (Stützle e Hoos, 1997), (Stützle e Hoos, 2000)
Problema Quadrático de Alocação	<ul style="list-style-type: none"> • Maniezzo (Maniezzo, 1999) • Maniezzo e Colorni (Maniezzo e Colorni, 1999) • Stützle e Hoos (Stützle e Hoos, 2000)
Problemas de Agendamentos (<i>scheduling</i>)	<ul style="list-style-type: none"> • Stützle (Stützle, 1998) • den Besten, Stützle e Dorigo (den Besten et al., 2000) • Gagné, Price e Gravel (Gagné et al., 2002) • Merkle, Middendorf e Shmeck (Merkle et al., 2002) • Blum (respectivamente, Blum e Sampels) (Blum, 2004), (Blum, 2005)
Problema de Roteamento de Veículos	<ul style="list-style-type: none"> • Gambardella, Taillard e Agazzi (Gambardella et al., 1999) • Reimann, Doerner e Hartl (Reimann e Hartl, 2004)
Coloração de Grafos	<ul style="list-style-type: none"> • Costa e Hertz (Costa e Hertz, 1997)
Problema de Ordenação Sequencial	<ul style="list-style-type: none"> • Gambardella e Dorigo (Gambardella e Dorigo, 2000)
Problema de Clique Máximo	<ul style="list-style-type: none"> • Bui e Rizzo Jr (Bui e Rizzo, 2004)

Fonte: BLUM (2005)

Já a tabela 2.2 mostra algumas aplicações em situações reais:

Tabela 2.2: Aplicações do ACO em problemas reais

Problemas	Autores
Data Mining	<ul style="list-style-type: none"> • Parpinelli, Lopes e Freitas (Parpinelli et al., 2002)
Colocação de Baterias em Projeto de Circuitos Integrados	<ul style="list-style-type: none"> • Alupoaei e Katkooi (Alupoaei e Katkooi, 2004)
Projeto em Redes de Comunicação	<ul style="list-style-type: none"> • Maniezzo, Boschetti e Jelasity (Maniezzo et al., 2004)

Problema em Bioinformática	<ul style="list-style-type: none"> • Shmygelska, Aguirre-Hernández e Hoos (Shmygelska et al., 2002) • Moss e Johnson (Moss e Johnson, 2003) • Karpenko, Shi e Dai (Karpenko et al., 2005) • Shmygelska e Hoos (Shmygelska e Hoos, 2005)
Música	<ul style="list-style-type: none"> • Guéret, Monmarché e Slimane (Guéret e Slimane, 2004)

Fonte: BLUM (2005)

De acordo com (Dorigo et al., 1999), os algoritmos baseados em colônias de formigas possuem, de uma forma geral, as seguintes características:

- Apesar de cada formiga ser capaz de encontrar uma solução, soluções de boa qualidade só emergirão como resultado da iteração coletiva entre as formigas;
- Cada formiga faz uso apenas de suas próprias informações, além das informações locais sobre o nó que visita;
- As formigas se comunicam apenas de forma indireta, através da leitura e escrita nas variáveis de rastro de feromônio;
- As formigas não são adaptáveis, pelo contrário, modificam a forma como o problema é representado e percebido para as outras formigas.

A figura 2.2, apresenta o pseudocódigo da Metaheurística ACO conforme descrito:

Figura 2.2: Pseudocódigo Metaheurística ACO

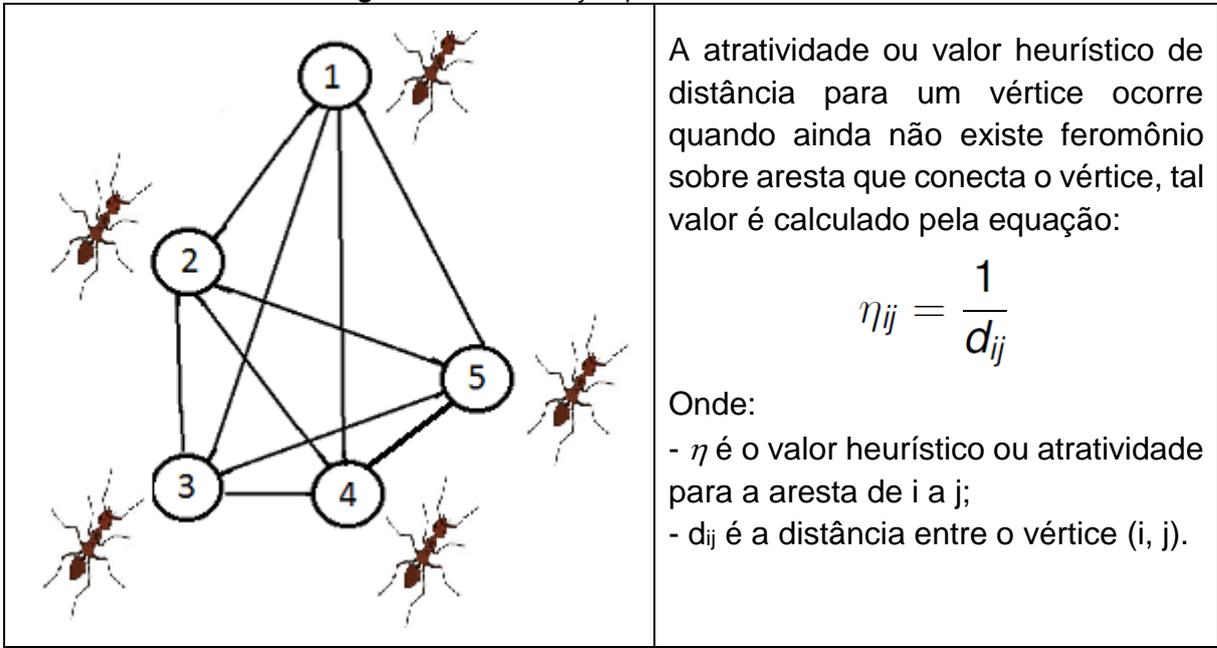
<p>Iniciar estrutura de dados com grafo Iniciar vetor de formigas Iniciar solução = ∞ Calcular a atratividade para as arestas Para $i = 0$ Até total de iterações Faça: Para formiga[k] = 0 Até total de formigas Faça: Enquanto a formiga[k] não construir um circuito completo Selecione o próximo vértice pela regra do feromônio ou da atratividade Fim do enquanto Calcule a distância da solução de cada formiga Se Solução da formiga[k] < Solução Então Solução = Solução formiga[k] Fim do Para Atualizar feromônio pela solução de cada formiga Fim do Para Fim ACO</p>

Fonte: Adaptado de Dorigo e Stutzle (2004)

2.1.3 Formulação do ACO

Inicialmente cada formiga encontra-se em um vértice do grafo para realizar um circuito completo, como mostra a figura abaixo.

Figura 2.3: Formulação para o Cálculo da Atratividade



Fonte: Autoria Própria.

De acordo com Carabetti *et al.* (2010), se a formiga k está em um vértice i , a probabilidade de ir para a outro vértice j é pseudoaleatória (parte determinística e aleatória juntas), dada pela função:

$$p_{ij}^k = \frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta}{\sum_{l \in N_i^k} (\tau_{il})^\alpha (\eta_{il})^\beta}, \text{ se } j \in N_i^k$$

Onde N_i^k é a lista dos vértices ainda não visitados pela formiga que se encontra no ponto i , sendo l um ponto ainda não visitado; τ_{ij} a quantidade de feromônio na aresta (i, j) ; η_{ij} um valor heurístico que representa a atratividade da formiga visitar o ponto i depois de visitar o ponto j . α e β são parâmetros que regulam a influência do feromônio e da heurística, ou seja, definem o grau de importância de τ e η respectivamente.

De acordo com Tavora (2013), o feromônio na construção da solução tem efeitos de *diversificação* e *intensificação*, ocorrendo em dois momentos:

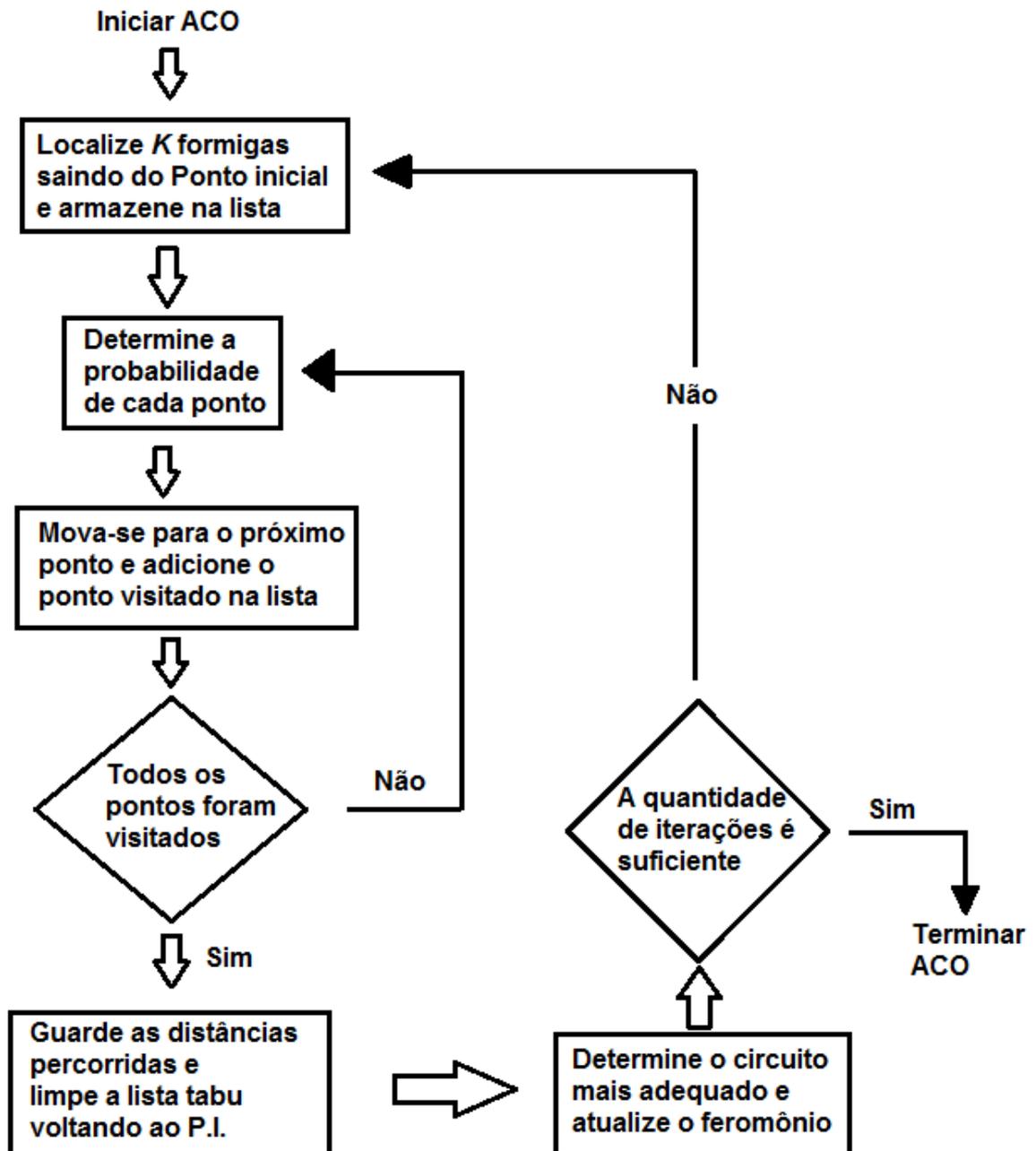
- Evaporação: Evita que o feromônio acumulado cresça indefinidamente, permitindo obter soluções diferentes e abandonando antigas soluções de busca.
- Deposição: Uma vez que as formigas percorreram seus circuitos e foram medidos seus caminhos, o feromônio $\tau_{i,j}$, associado a junção entre as cidades i e j e atualizado (depositado) em diferentes quantidades em cada aresta do grafo, de acordo com o desempenho de cada formiga. Como mostra a seguinte equação:

$$\tau_{ij} = (1 - \rho) + \sum_{k=1}^m \Delta \tau_{ij}^{(k)}$$

Onde $\Delta \tau_{ij}$ é a quantidade de feromônio que será depositada pela formiga k na aresta (i, j) ; τ_{ij} é o feromônio associado a aresta (i, j) ; 1 é um valor constante; ρ é a taxa de evaporação do feromônio, que deve ser entre 0 e 1; m é o total de formigas.

O funcionamento de todo esse procedimento é mostrado na figura 2.4 abaixo:

Figura 2.4: Fluxograma da Metaheurística ACO



Fonte: Adaptado de Tavora (2013).

A partir de sua formulação inicial, diferentes variantes da Metaheurística ACO têm sido propostas na literatura. Duas dessas variantes bem conhecidas das quais foram deduzidas neste trabalho, são mostradas a seguir.

2.1.4 Estratégia Elitista

Uma das primeiras variantes do algoritmo Ant System (AS) foi a estratégia elitista apresentada por (Dorigo et al., 1996), que consiste em premiar as formigas cujas arestas sejam pertencentes à melhor solução até o momento, assim procurando soluções ainda melhores em sua vizinhança. Um reforço adicional de feromônio é dado cada vez que a trilha de feromônio for atualizada, caso seja a melhor rota. As arestas com melhor solução global são reforçadas com uma quantidade $d \frac{1}{L^{gb}}$, em que L^{gb} é o comprimento da melhor solução global e d é um número inteiro positivo.

2.1.5 Estratégia ASrank

Proposto por (Bullnheimer et al., 1997), a Estratégia ASrank classifica as formigas de acordo com o tamanho de suas rotas geradas e, após cada fase de construção, somente as melhores formigas e a melhor formiga global é que podem depositar feromônio nos arcos. A melhor formiga da colônia contribui para a atualização de feromônio com um determinado peso, enquanto a melhor global reforça as trilhas de feromônio.

Problemas de otimização combinatória tratados com ACO são normalmente codificados através de um grafo de construção completo $G(V,A)$, onde os nós de V são componentes da solução e os arcos de A são conexões entre os componentes. Construir uma solução significa encontrar um caminho viável em G . Por exemplo, no problema do caixeiro viajante, os nós correspondem aos clientes, os arcos correspondem as ruas que conectam os clientes e uma solução viável é um caminho hamiltoniano no grafo (Bianchi, 2006).

2.2 Circuito Hamiltoniano

Conforme Ore (1990), em 1859, o matemático irlandês William Hamilton inventou um jogo chamado “The Icosian Game”, com um peculiar enigma envolvendo um dodecaedro, em que cada um dos 20 vértices foram nomeados com nomes de cidades importantes. Como mostra a figura 2.5 abaixo.

Figura 2.5: *The Icosian Game*



Fonte: The Puzzle Museum (2017)

O objetivo do jogo era, utilizando as 30 arestas do dodecaedro, passar por cada uma das cidades apenas uma vez, começando e terminando na mesma cidade.

Tendo isso em vista, encontrar um circuito Hamiltoniano em um grafo $G = (V, E)$, que Cormen (2012) explica sendo um conjunto de cidade $C = (c_1, c_2, c_3, c_4, \dots, c_n)$, para calcular a distância $d_{ij} = (c_i, c_j)$ ($i, j \in \{1, \dots, n\}$, $d_{ij} = d_{ji}$ e $d_{ii} = 0$), com o objetivo de encontrar a permutação $s^* \in S$, que faça a função minimizar o custo do circuito em grafo, sem repetir vértice e voltando ao vértice inicial (Problema do Caixeiro Viajante).

A dificuldade está em encontrar dentre todas as combinações possíveis a que satisfaz as restrições e a função objetivo com o menor custo, sendo intratável tal processamento de dados por algoritmo exato para computar uma grande entrada de dados, por causa da imensa quantidade de soluções possível, na ordem de $(n - 1)! / 2$, onde n é a quantidade de vértices de um grafo completo.

2.3 O Problema do Caixeiro Viajante (PCV)

O Problema do Caixeiro Viajante (PCV) é um dos ilustres representantes da Otimização Combinatória, segundo Ragsdale (2009). Tem sido muito utilizado em experimentos de diversos métodos de otimização, por ser um problema de fácil descrição e compreensão, e por possuir inúmeras aplicações.

Applegate (2007) cita que “dado um conjunto de n cidades, um caixeiro viajante deve partir de uma cidade inicial, visitar todas as $(n - 1)$ cidades restantes, passando uma única vez em cada e regressar a cidade de origem, de tal forma que a distância percorrida seja mínima”.

Problemas de Otimização Combinatória segundo Garey e Johnson (1979) pertencem a classe NP-difícil, inclusive o PCV, o que significa na prática que dificilmente existirão algoritmos polinomiais que os solucionem, fazendo surgir a necessidade das Metaheurísticas para resolução em tempo hábil.

2.3.1 Formulação

Na formulação original feita por Dantzig, Fulkerson e Johnson (1954), o PCV é apresentado como um problema de programação binária sobre um grafo $G = (N, A)$ da seguinte forma: minimizar a função objetivo. Segue abaixo a formulação:

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

Sujeito a:

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

$$\sum_{j=1}^n x_{ij} = 1 \quad \forall i \in N \quad (2)$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1 \quad \forall S \subset N \quad (3)$$

$$x_{ij} \in \{0,1\} \quad \forall i, j \in N \quad (4)$$

A função objetivo z é um duplo somatório que deve ser minimizado para ir de uma cidade i para uma cidade j , onde c_{ij} é o custo de se transpor uma aresta $(i, j) \in A$ e x_{ij} é variável binária (0-1), que determina se a aresta faz parte do circuito ou não;

quanto as restrições, temos que (1) garante uma única conexão de partida em i , o que quer dizer que ambas as restrições asseguram que cada vértice possui apenas um sucessor e um antecessor. S é um subgrafo de $G = (N, A)$, em que $|S|$ representa, através do símbolo de cardinalidade, o número de vértices desse subgrafo. (2) garante cada posição de chegada j pertencente apenas a uma única cidade; (3) garante descartar subrotas que não formem circuito hamiltoniano.

No próximo capítulo são apresentados três estratégias adicionais adaptadas da Metaheurística ACO, mostrando suas características.

3 METODOLOGIA

Esta pesquisa foi realizada com base em método científico descrito em Wazlawick (2008), fazendo uso de método dedutivo com a natureza descritiva de objetivo exploratório, delineada por pesquisa bibliográfica, fazendo análise quantitativa de dados coletados e experimento computacional para a investigação científica em busca de resposta ao problema tratado.

A implementação de três diferentes estratégias baseada no uso da Metaheurística colônia de formigas foi desenvolvido em linguagem C. Como parâmetros de configuração da heurística foi utilizado o valor de 0,5 para α , β , ρ . A execução dos testes computacionais foram realizados em uma máquina com 16GB de Memória Ram, processador Intel Core i7 e o sistema operacional Windows 10. Foram utilizadas 6 instancias como mostra a tabela abaixo:

Tabela 3.1: Instâncias Utilizadas

Instância	Grafo $G = (V, E)$	Ótimo Global
1. lin105.xml	G(105, 10.920)	14379
2. pr107.xml	G(107, 11.342)	44303
3. pr124.xml	G(124, 15.252)	59030
4. bier127.xml	G(1227, 16.002)	118282
5. pr144.xml	G(144, 20.592)	58537
6. ch150.xml	G(150, 22.350)	6528

Foi revisado a Metaheurística de Otimização por Colônia de Formigas, o Problema do Caixeiro Viajante com resolução de instâncias da biblioteca TSPLib (Library of Traveling Salesman Problems), disponibilizada pela Universitat Heidelberg, e feita a análise quantitativa. A TSPLIB é uma biblioteca de instâncias de exemplo para o TSP (traveling salesman problem) e problemas relacionados, de várias fontes e de vários tipos (GERHARD, 2017).

Foram implementados três estratégias distintas usando a Metaheurística ACO, e aplicadas as mesmas instâncias para ambas. As estratégias são apresentadas a seguir.

3.1 Estratégia 1

Seguindo o modelo da Metaheurística ACO, na primeira estratégia foi adicionado a implementação do uso de Formiga Elite, também chamado de solução elite, ou seja, atualização de feromônio somente da formiga com o melhor caminho e apenas uma aresta (i, j) na iteração 1, duas arestas (i, j) na iteração 2, três arestas (i, j) na iteração 3, até o total de vértices do grafo.

Também foi implementado Busca Aleatória pela atratividade quando a aresta (i, j) não possui feromônio. Ou seja, caso a aresta não possua feromônio, utiliza uma variável randômica gerando o valor para o próximo vértice a ser visitado, caso o vértice já tenha sido visitado, a variável randômica é desligada.

Figura 3.1: Pseudocódigo Estratégia 1

```

Procedimento ACO
  Iniciar estrutura de dados com grafo
  Iniciar vetor de formigas
  Iniciar solução =  $\infty$ 
  Calcular a atratividade para as arestas
  Para i = 0 Até total de Iterações Faça:
    Para formiga[k] = 0 Até total de formigas Faça:
      Enquanto a formiga[k] não construir um circuito completo
        Selecione o próximo vértice pela regra do feromônio ou da atratividade
      Fim do enquanto
      Calcule a distância da solução de cada formiga
      Se Solução da formiga[k] < Solução Então Solução = Solução formiga[k]
    Fim do Para
  Atualizar feromônio pela solução de cada formiga
Fim do Para
Fim ACO

```

Fonte: Adaptado de Dorigo e Stutzle (2004).

3.2 Estratégia 2

Foi implementado nessa estratégia a utilização de:

- (i) Uso de formiga Elite, ou seja, atualização de feromônio somente da formiga com o melhor caminho e apenas uma aresta (i, j) na iteração 1, na iteração 2, duas arestas (i, j) , na iteração 3, três arestas (i, j) , até o total de vértices do grafo.
- (ii) Troca de posição entre vértices vizinhos para o caminho da formiga elite. A troca de posição ocorre com o vizinho mais próximo, tentando evitar caminhos infinitos e buscando melhor solução.
- (iii) Busca Aleatória pela atratividade quando a aresta (i, j) não possui feromônio. Utilizando a variável randômica.

Figura 3.2: Pseudocódigo Estratégia 2

Procedimento ACO

Iniciar estrutura de dados com grafo

Iniciar vetor de formigas

Iniciar solução = ∞

Calcular a atratividade para as arestas

Para $i = 0$ Até total de Iterações Faça:

 Para formiga[k] = 0 Até total de formigas Faça:

 Enquanto a formiga[k] não construir um circuito completo

 Selecione o próximo vértice pela regra do feromônio ou da atratividade

 Fim do enquanto

 Calcule a distância da solução de cada formiga

 Se Solução da formiga[k] < Solução Então Solução = Solução formiga[k]

 Fim do Para

 Atualizar feromônio pela solução de cada formiga

Fim do Para

Fim ACO

Fonte: Adaptado de Dorigo e Stutzle (2004).

3.3 Estratégia 3

Esta estratégia possui todas as características das estratégias anteriores, no entanto, foi adicionado outra característica, a utilização de apenas 1/3 das formigas para caminhar pelo feromônio, embora todas depositem feromônio caso seja escolhida como de melhor caminho na respectiva iteração.

Implementação para Pseudocódigo:

- (i) Uso de formiga Elite, ou seja, atualização de feromônio somente da formiga com o melhor caminho e apenas uma aresta (i, j) na iteração 1, na iteração 2, duas arestas (i, j) , na iteração 3, três arestas (i, j) , até o total de vértices do grafo.
- (ii) Troca de posição entre vértices vizinhos para o caminho da formiga elite buscando melhor solução.
- (iii) Busca Aleatória pela atratividade quando a aresta (i, j) não possui feromônio. Utilizando a variável randômica.
- (iv) Apenas 1/3 das formigas andam pelo feromônio, mas todas depositam feromônio caso seja escolhida como de melhor caminho na respectiva iteração.

Figura 3.3: Pseudocódigo Estratégia 3

```

Procedimento ACO
  Iniciar estrutura de dados com grafo
  Iniciar vetor de formigas
  Iniciar solução = ∞
  Calcular a atratividade para as arestas
  Para i = 0 Até total de Iterações Faça:
    Para formiga[k] = 0 Até total de formigas Faça:
      Enquanto a formiga[k] não construir um circuito completo
        Selecione o próximo vértice pela regra do feromônio ou da atratividade
      Fim do enquanto
      Calcule a distância da solução de cada formiga
      Se Solução da formiga[k] < Solução Então Solução = Solução formiga[k]
    Fim do Para
  Atualizar feromônio pela solução de cada formiga
Fim do Para
Fim ACO

```

Fonte: Adaptado de Dorigo e Stutzle (2004).

4 RESULTADOS

Os resultados obtidos para cada instancia após 30 experimentos realizados para cada uma das diferentes estratégias, pode-se avaliar seu desempenho, observando a média das soluções, o desvio padrão dos resultados e o tempo médio, em segundos, de execução para cada instancia. As tabelas abaixo mostram os resultados após a implementação das três estratégias utilizando as instâncias do TSPLIB já mencionadas.

Tabela 4.1: Resultado Estratégia 1.

Instancia	Media	Desvio Padrão	Tempo(s)
1. lin105.xml	16387,8	266,69	31
2. pr107.xml	48033,4	284,87	50
3. pr124.xml	66853,5	604,18	63
4. bier127.xml	132007,6	1133,17	104
5. pr144.xml	61497,1	116,45	109
6. ch150.xml	7332,0	88,75	157

Tabela 4.2: Resultado Estratégia 2.

Instancia	Media	Desvio Padrão	Tempo(s)
1. lin105.xml	15679,8	285,54	35
2. pr107.xml	45371,1	671,94	46
3. pr124.xml	64417,7	1077,70	75
4. bier127.xml	128319,6	21185,27	87
5. pr144.xml	60826,5	352,83	110
6. ch150.xml	7099,8	1106,55	142

Tabela 4.3: Resultado Estratégia 3.

Instancia	Media	Desvio Padrão	Tempo(s)
1. lin105.xml	15184,6	259,79	31
2. pr107.xml	45135,2	479,45	47
3. pr124.xml	63311,6	1144,55	81
4. bier127.xml	123651,6	1588,29	98
5. pr144.xml	60881,9	348,15	112
6. ch150.xml	6984,0	119,36	165

4.1 Comparação dos Resultados

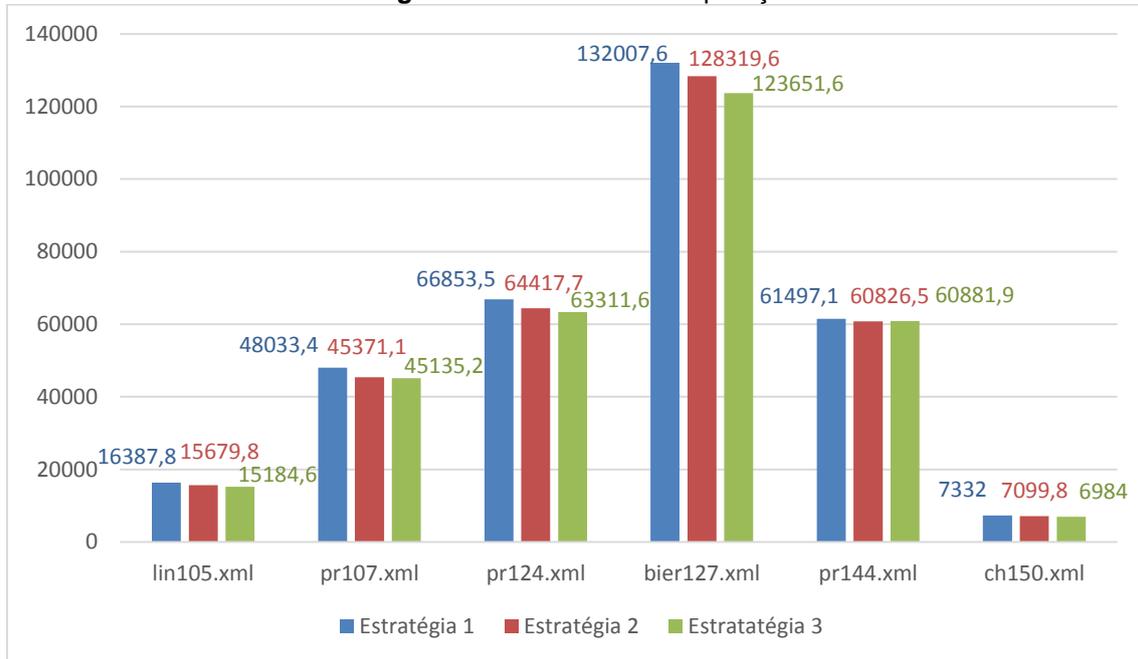
Comparando a média dos resultados das três estratégias para cada uma das instâncias, podemos perceber que a estratégia 3 possui um melhor desempenho em relação as demais, utilizando além da característica das demais estratégias, o procedimento de apenas 1/3 das formigas andarem pelo feromônio, mas todas depositam feromônio caso seja escolhida como de melhor caminho na respectiva iteração.

Tabela 4.4: Comparação da média dos resultados.

Instâncias	Estratégia 1	Estratégia 2	Estratégia 3
1. lin105.xml	16387,8	15679,8	15184,6
2. pr107.xml	48033,4	45371,1	45135,2
3. pr124.xml	66853,5	64417,7	63311,6
4. bier127.xml	132007,6	128319,6	123651,6
5. pr144.xml	61497,1	60826,5	60881,9
6. ch150.xml	7332,0	7099,8	6984,0

O gráfico abaixo apresenta a comparação da média dos resultados das três estratégias.

Figura 4.1: Gráfico de Comparação das médias.



Comparando a média dos resultados das estratégias com o ótimo global de cada instância da TSPLIB, disponibilizada pela Universitat Heidelberg, a estratégia 3 apresentou os resultados mais aproximados, como mostra a tabela abaixo.

Tabela 4.5: Comparação das estratégias com o ótimo global

TSPLIB			Metaheurística ACO Média de 30 Execuções		
Instância	Grafo G(V, E)	Ótimo Global	Estratégia 1	Estratégia 2	Estratégia 3
1. lin105.xml	G(105, 10.920)	14379	16387,8	15679,8	15184,6
2. pr107.xml	G(107, 11.342)	44303	48033,4	45371,1	45135,2
3. pr124.xml	G(124, 15.252)	59030	66853,5	64417,7	63311,6
4. bier127.xml	G(1227, 16.002)	118282	132007,6	128319,6	123651,6
5. pr144.xml	G(144, 20.592)	58537	61497,1	60826,5	60881,9
6. ch150.xml	G(150, 22.350)	6528	7332,0	7099,8	6984,0

A instancia 2. Pr107.xml apresentou uma melhor média para as estratégias 2 e 3 aproximando-se do ótimo global. A instancia 5 também obteve resultados aproximado para todas as estratégias. A tabela abaixo mostra melhor a diferença do Ótimo Global para as estratégias em porcentagem.

Tabela 4.6: Diferença das estratégias para o ótimo global %

TSPLIB			Metaheurística ACO Média de 30 Execuções		
Instância	Grafo G(V, E)	Ótimo Global	Estratégia 1	Estratégia 2	Estratégia 3
1. lin105.xml	G(105, 10.920)	14379	13,97%	9,05%	5,60%
2. pr107.xml	G(107, 11.342)	44303	8,42%	2,42%	1,88%
3. pr124.xml	G(124, 15.252)	59030	13,25%	9,13%	7,25%
4. bier127.xml	G(1227, 16.002)	118282	11,60%	8,49%	4,54%
5. pr144.xml	G(144, 20.592)	58537	5,06%	3,91%	4,01%
6. ch150.xml	G(150, 22.350)	6528	12,32%	8,76%	6,99%

Comparando a média do tempo de execuções das três estratégias para cada instancia, percebe-se que a estratégia 2 mostra-se um pouco mais eficiente na maioria das instancias que as outras duas estratégias.

Tabela 4.7: Comparação do tempo das três estratégias.

TSPLIB		Media de Tempo de execuções em segundos		
Instância	Grafo G(V, E)	Estratégia 1	Estratégia 2	Estratégia 3
1. lin105.xml	G(105, 10.920)	31	35	31
2. pr107.xml	G(107, 11.342)	50	46	47
3. pr124.xml	G(124, 15.252)	63	75	81
4. bier127.xml	G(1227, 16.002)	104	87	98
5. pr144.xml	G(144, 20.592)	109	110	112
6. ch150.xml	G(150, 22.350)	157	142	165

5 CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho apresenta um estudo sobre Otimização por Colônia de Formigas, com foco no estudo do problema do Caixeiro Viajante.

Foi apresentado neste trabalho métodos para solução do problema do caixeiro viajante, por meio de investigações de estratégias desenvolvidas baseadas na Metaheurística Colônia de Formigas. Para isso foi feito um levantamento bibliográfico acerca dos temas e problemas envolvidos.

As três estratégias implementadas foram adaptações realizadas do processo padrão de otimização por colônia de formigas. Foi utilizado formiga elite, ou seja, o feromônio é atualizado somente da formiga com melhor caminho e apenas uma aresta (i, j) na iteração 1, duas arestas (i, j) na iteração 2, até o total de vértices do grafo. Foi adicionado uma busca aleatória pela atratividade quando a aresta (i, j) não possui feromônio. Uma troca de posição entre vértices vizinhos para o caminho da formiga elite buscando uma melhor solução. Além disso o uso de apenas 1/3 das formigas andam pelo feromônio, mas todas depositam feromônio caso sejam escolhidas como de melhor caminho em sua respectiva iteração.

As estratégias foram aplicadas em 6 instâncias da TSPLIB, com 30 experimentos realizados para cada uma, produzindo soluções aproximadas ao ótimo global de cada instância. A associação entre a Metaheurística ACO e as diferentes estratégias foi capaz de fornecer resultados interessantes e deve ser melhor estudada.

Algumas perspectivas futuras para este trabalho são a utilização de novas instâncias, a implementação de mais estratégias, ou até mesmo uma associação com outras Metaheurística. A utilização das estratégias para solução de outros problemas de otimização.

6 REFERÊNCIAS

- APPLEGATE, D. L.; BIXBY, R. L. & CHVATAL, V. et al. **The Traveling Salesman Problem: A Computational Study**. Princeton University Press, New Jersey, 2007.
- BIANCHI, L. **Ant Colony Optimization and Local Search for the Probabilistic Traveling Salesman Problem: A Case Study in Stochastic Combinatorial Optimization**. Tese de Doutorado, ULB – Université Libre de Bruxelles, 2006.
- BLUM, Christian. **Ant colony optimization: Introduction and recent trends**. Physics of Life Reviews v. 2. 2005.
- BULLNHEIMER, Bernd; Hartl, Richard F. e Strau, Christine. **A new rank based version of the ant system - a computational study**. Central European Journal for Operations Research and Economics, v. 7, p. 25–38. 1997.
- CARABETTI, Eduardo Goecking. **Metaheurística Colônia de Formiga Aplicada ao Problema de Roteamento de Veículos com Coleta e Entrega e Janela de Tempo**. Dissertação de Mestrado, CEFET/MG, Belo Horizonte, 2010.
- Cordenonsi, A. Z. **Ambientes, Objetos e Dialogicidade: Uma Estratégia de Ensino Superior em Heurísticas e Metaheurísticas**. PhD thesis, UFRGS, 2008.
- CORMEN, T. H; Rivest, R. L.; Leiserson, C. E.; Stein, C. **Algoritmos - Teoria e Prática**. Elsevier – Campus - 3ª Ed. 2012.
- DANTZIG, G. B.; FULKERSON, D. R.; JOHNSON, S. M. **Solution of a Large Scale Traveling Salesman**, 1954.
- DIAZ, A. F.; Velarde, J. L. G; Laguna, M.; Moscato, P.; Tseng, T.; Glover, F. e Ghaziri, H. M. **Optimización Heurística y Redes Neuronales en Dirección de Operaciones e Ingeniería**. Paraninfo. 1996.
- DORIGO, Marco. **Optimization, Learning and Natural Algorithms** (in Italian). PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Milão, Itália, 1992.
- DORIGO, M; Maniezzo, V. e Colorni, A. **The ant system: Optimization by a colony of cooperating agents**. IEEE Transactions on Systems, Man, and Cybernetics-Part B, v. 26, n. 1, p. 29–41. 1996.
- DORIGO, Marco; Di Caro, Gianni e Gambardella, Luca M. **Ant algorithms for discrete optimization**. Artificial Life, v. 5, n. 2, p. 137–172. 1999.
- DORIGO, M. e Stützle, T. **Ant Colony Optimization**. MIT Press, Boston, MA. 2004.
- GAREY, M. R.; Johnson, D. S. **Computer and Intractability - a guide to the theory of NP - completeless**. W.H. Freeman and Company, San Francisco, 338 p., 1979.

GUTJAHR, W. J. **First steps to the runtime complexity analysis of ant colony optimization**. Relatório técnico, Department of Statistics and Decision Support Systems, University of Vienna, Áustria. 2007.

KARP, Richard M. **Reducibility Among Combinatorial Problems**. In R. E. Miller and J. W. Thatcher (editors). Complexity of Computer Computations. New York: Plenum. pp. 85–103. 1972.

ORE, Oystein. **Graphs and their Uses**. 2. ed. Washington: The Mathematical Association of America, 1990.

RAGSDALE, C. T. **Modelagem e Análise de Decisão**. Cengage Learning, Sao Paulo, 590 p., 2009.

RIBEIRO, C. C. **Metaheuristics and applications**. Advanced School on Artificial Intelligence (Constraint Programming).Monte Estoril, Portugal. 1996.

SOUZA, M. J. F. **Inteligência computacional para otimização**. Relatório Técnico, Departamento de Computação, Universidade Federal de Ouro Preto. URL <http://www.decom.ufop.br/prof/marcone>. 2009.

TAVORA, R. C. M.; Leoni R. C. **O uso da colônia de formigas no problema de visitação na AMAN**. Simpósio de Excelência em Gestão e tecnologia. 2013.

GERHARD, Reinelt. **Universität Heidelberg. Institut für Informatik Im Neuenheimer Feld 368 D - 69120 Heidelberg. Germany**. <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95>. Acessado em janeiro de 2017.

WAZLAWICK, Raul Sidnei. **Metodologia de Pesquisa para Ciência da Computação**. Editora: Elsevier. 2008.

ANEXOS

Resultado de 30 experimentos das três estratégias para a instancia 1. lin105.xml

Lin105.xml					
Estratégia 1		Estratégia 2		Estratégia 3	
Solução	Tempo (s)	Solução	Tempo (s)	Solução	Tempo (s)
16819.2	48.927	15260.6	46.335	14547	58.087
16270.9	41.566	15798.3	64.052	14901.7	43.886
16229.2	49.84	15861.3	49.706	15373.6	47.788
16388.2	55.824	15413.2	30.995	14979.3	42.129
16701.8	29.46	15692.9	61.209	14867.5	55.814
16556.3	56.174	15627.2	64.067	15053.6	53.015
16529.6	33.946	15821.3	30.469	15455.7	63.671
16336	66.907	15174.9	31.69	14911	37.1
16074	36.868	16056.5	38.336	15215.8	45.322
16064.9	67.201	15773	50.827	15577.3	49.665
16485.4	65.498	15713.6	50.198	15089.2	44.194
15991.1	42.929	16284.8	59.729	15659.2	44.132
16593.5	49.887	15448.7	43.507	15191.2	35.337
16228.2	27.416	15821.4	27.778	15347.1	39.479
16383.7	67.658	16096.5	37.448	15389	49.296
16410.2	66.254	15900.4	23.512	15413.7	56.061
16123.4	59.133	15591.4	50.673	14964.9	54.032
16840.9	62.589	15328.7	57.843	15085.8	35.07
16774.7	46.23	15482.4	29.987	15450	66.358
16623.4	64.813	15506.4	32.734	14659.9	35.507
15968.2	64.7	15777.9	62.696	15302.5	42.944
16291.4	35.805	15457.5	62.191	15229.4	53.089
16514.1	16.711	16035.4	19.54	15232.9	45.762
16709.1	60.511	16098.7	54.628	15587.6	47.085
16332.6	52.64	15922.9	72.697	15170.5	61.747
16377.9	57.511	15753.9	49.826	15309.2	46.983
16246.6	67.242	15652.8	43.204	15257.1	43.058
16667	47.033	15585.2	23.495	15302.4	59.039
15728.3	28.159	15136	63.67	14980.7	32.222
16373.1	34.564	15319.6	54.607	15032.5	50.187

Resultado de 30 experimentos das três estratégias para a instancia 2. pr107.xml

Pr107.xml					
Estratégia 1		Estratégia 2		Estratégia 3	
Solução	Tempo (s)	Solução	Tempo (s)	Solução	Tempo (s)
47853.8	24.777	44827	31.341	46206.4	39.833
47983.8	14.643	45443.8	39.815	45178.9	44.62
47901.9	42.551	45585.7	41.754	44740.7	25.963
47822	35.909	44589.1	16.78	45184	15.679
47979.9	29.203	45288.1	6.938	43778.1	35.124
48506.9	44.356	44416.5	25.487	45343.9	44.467
48099.4	43.313	45610	16.97	45316.4	31.794
47507.6	29.361	44453.5	32.489	44527.9	47.373
48154.6	34.548	45973.5	7.065	46071.7	7.907
47458.1	24.565	45456.3	42.218	44895.7	15.652
47624.8	14.19	44914	10.227	45371.7	47.129
48181.7	17.261	44441.5	35.299	45723.1	38.751
47624.8	7.108	45832.6	5.316	44906.6	26.558
48613.5	40.996	46437.9	35.936	43821.6	50.803
47993.3	35.566	46375.9	67.664	44001.8	43.047
48346.1	26.142	46971.3	60.814	45538	45.525
47857.6	35.609	45592.6	40.48	44315.8	11.765
47857.6	18.535	45687.8	40.123	45160.7	44.055
48100.2	46.595	45885.8	64.219	45761.6	26.892
48219.8	51.334	44493	36.283	45409.1	32.76
47704.6	35.723	44333.3	34.083	44856.9	42.002
48001.8	28.192	46049.3	27.198	44210.2	16.952
48115.7	10.319	45931.4	42.114	46631.6	3.146
48206.9	11.477	45445.8	42.549	43102.5	28.904
48491.8	70.11	44708.8	10.189	44316.1	31.256
47938.9	37.838	45362.8	42.215	44220.1	23.417
48096.5	14.3	45535	39.032	45806	37.921
48363.3	51.652	45852.3	39.74	43416.2	58.679
48024.8	21.223	44987.2	63.302	45430.1	17.48
48370.3	21.015	44651.2	31.367	44175.2	17.261

Resultado de 30 experimentos das três estratégias para a instancia 3. pr124.xml

Pr124.xml					
Estratégia 1		Estratégia 2		Estratégia 3	
Solução	Tempo (s)	Solução	Tempo (s)	Solução	Tempo (s)
66901.7	74.879	64904.1	104.67	62223	93.424
65189.6	77.732	65758.3	74.032	63007.9	31.979
66458.8	59.217	64125	82.794	60520.1	93.379
66764.6	82.819	63621.9	90.309	64098.3	94.71
67514.2	61.796	62350.9	68.705	61862.5	91.603
66667.6	24.581	65522.1	22.221	61816.8	88.9
66987.4	92.754	64203.9	120.062	63968.1	106.748
66551.5	51.462	63199.8	31.674	62439.6	94.735
67061.4	88.731	64718.4	38.396	62770.2	91.888
66738.3	70.527	61732.6	85.07	63266.2	93.461
67297.6	26.367	64385.7	118.685	63093.4	88.018
66605.3	53.789	65229.5	25.481	63141.6	88.316
68037.5	103.562	63986.9	114.654	63866.2	84.968
66066.4	23.699	64145.4	71.888	64621.2	94.578
67381	80.38	65433.9	116.507	63633	94.189
68070.9	39.16	65721.4	78.781	65640.4	91.875
67061.4	36.073	62515.8	84.003	63299.5	100.572
66507.6	23.92	64564.7	89.751	62226.5	88.315
66792	112.568	63593.2	24.843	63662.1	84.601
67158.3	70.197	64545.7	22.479	62726.3	40.757
66901.7	74.101	64623.7	78.468	64974.7	38.675
65833.9	112.697	64672.7	43.449	61926.2	38.117
66907.1	30.86	66344.2	87.538	62386.1	80.655
66735.6	112.382	65192	120.09	63926.3	36.948
67690.1	108.406	64257.3	47.246	64877.8	95.216
66003.3	23.63	65576.9	115.292	63804.6	100.389
67368	27.483	63083	78.323	65206.8	30.865
67228.6	52.485	65689.1	81.825	63339.8	96.89
66455.5	40.163	63870.3	68.594	64567.8	85.586
66667.6	79.671	64962.8	76.886	62454.2	93.619

Resultado de 30 experimentos das três estratégias para a instancia 4. bier127.xml

Bier27.xml					
Estratégia 1		Estratégia 2		Estratégia 3	
Solução	Tempo (s)	Solução	Tempo (s)	Solução	Tempo (s)
132069	97.669	129061	118.037	124511	83.897
132648	52.937	129475	103.107	121930	115.471
133832	122.585	130139	92.967	123352	112.188
131037	106.089	129202	124.206	121905	112.508
133109	120.286	131777	83.881	125965	107.544
131470	122.937	129544	51.095	125609	132.91
132095	86.087	129756	92.569	123437	127.066
130306	112.267	129934	123.961	124389	99.045
131552	93.027	125900	124.843	121894	118.27
133330	103.666	126998	91.701	123239	127.246
132106	122.271	124869	109.635	125478	121.526
132499	99.145	125668	95.921	126837	120.79
132240	54.494	129390	118.386	123993	101.735
132375	104.473	125103	120.067	124809	115.764
133142	112.606	127536	131.632	123667	118.596
131714	105.049	131355	101.124	123359	130.294
132687	127.069	130713	135.039	121749	91.374
132699	131.23	127734	122.263	119782	120.256
133006	114.43	128622	112.848	123327	136.898
132944	92.006	126645	123.108	121053	116.873
129678	111.173	126167	122.258	124925	114.756
130473	98.236	132389	90.153	121049	134.061
131819	126.487	130067	102.008	123511	113.276
132467	82.607	123998	137.464	123016	120.716
131280	103.153	130009	131.922	123459	124.388
128535	123.551	125618	91.914	124009	97.479
131672	86.715	130520	126.416	124716	103.534
132544	118.515	127157	127.798	125332	100.617
133299	87.052	126806	122.74	124121	94.039
131602	118.187	127436	88.292	125124	62.105

Resultado de 30 experimentos das três estratégias para a instancia 5. pr144.xml.

Pr144.xml					
Estratégia 1		Estratégia 2		Estratégia 3	
Solução	Tempo (s)	Solução	Tempo (s)	Solução	Tempo (s)
61495.3	78.865	61189.1	110.772	60368.8	142.723
61495.3	86.631	60121.7	109.407	61189.1	34.783
61495.3	123.815	61032.5	101.901	60904	126.814
61488.7	64.965	61013.1	92.441	60835.5	18.807
61495.3	105.154	60760.1	106.221	61263.6	82.793
61495.3	106.818	60428.1	102.628	60722.2	132.97
61495.3	108.607	60084	104.069	60859.5	125.533
61495.3	122.237	60847.1	95.399	61013.1	133.599
61495.3	138.704	61002.5	60.46	61176.9	138.837
61495.3	91.467	60869.5	129.646	60977.9	103.289
61495.3	94.1	61219.5	82.182	60584.2	85.883
61495.3	72.563	60635.8	22.393	60905.1	82.819
61495.3	108.8	61081.1	67.943	60538.5	78.456
61495.3	103.746	60834.7	131.013	61189.1	119.441
61495.3	15.529	61005.1	132.737	61189.1	114.558
61495.3	49.453	60834.7	68.455	60994.9	119.064
61469.1	87.699	59773.3	54.005	61359.2	91.919
61495.3	75.257	61033.7	100.801	61219.5	61.661
61495.3	98.544	61033.7	119.731	60865.9	40.272
61495.3	123.367	61033.7	94.699	60251.3	139.605
61495.3	114.706	61033.7	77.244	61219.5	93.444
61495.3	30.246	60449	109.169	61219.5	134.762
61495.3	100.844	61189.1	99.756	60592.4	91.389
61495.3	101.699	60512.2	50.152	61020.6	106.746
61495.3	45.224	60796	15.719	61023.2	46.262
61495.3	92.95	61033.7	113.558	60557.4	127.333
61495.3	105.087	61033.7	25.656	61189.1	67.281
61495.3	102.387	61033.7	29.25	61014.5	110.641
61495.3	118.644	60635.8	145.076	59915.6	135.089
61581.8	99.234	61243.7	78.829	60298.3	77.566

Resultado de 30 experimentos das três estratégias para a instancia 6. ch150.xml.

Ch150.xml					
Estratégia 1		Estratégia 2		Estratégia 3	
Solução	Tempo (s)	Solução	Tempo (s)	Solução	Tempo (s)
7424.49	153.217	7126.24	149.75	7117.02	166.607
7389	83.87	7022	158.462	6865.97	190.798
7316.37	184.423	7152.89	192.681	6892.44	181.68
7334.78	187.123	7142.55	157.206	6767.26	167.843
7389	189.358	7203.28	185.434	6927.74	166.021
7384.77	108.274	6888.79	187.272	6822.75	177.827
6987.76	143.511	7266.15	144.203	6997.26	113.923
7407.35	164.97	7138.41	184.469	7050.81	165.834
7308.19	132.465	7198.23	79.451	7056.3	161.522
7308.19	171.995	7176.96	73.953	7128.11	94.487
7302.21	151.232	7158.28	164.641	7218.48	180.76
7327.29	157.712	7216.41	75.033	6833.44	161.665
7355.65	151.83	6886.32	144	7214.49	167.374
7264.58	176.345	7138.43	90.264	6997.76	172.38
7405.33	144.852	7138.41	190.733	7108.03	172.662
7449.17	144.246	7009.02	93.233	7044.17	179.648
7377.43	152.176	7063.79	86.456	7069.63	177.054
7172.33	176.177	7117.91	78.998	7069.6	135.169
7277.71	184.859	7095.48	193.51	6879.9	178.715
7308.19	187.704	7028.5	180.858	6931.11	136.517
7377.43	169.29	6927.02	152.311	7034.81	161.352
7416.23	195.917	6883.11	186.351	6755.32	183.435
7308.19	184.39	7147.48	98.962	7071.9	174.886
7270.39	184.212	7142.9	160.925	7004.29	185.07
7283.96	146.184	7022	95.959	6901.59	184.011
7270.06	148.133	7084.2	172.479	7095.51	172.977
7325.23	75.71	7250.21	102.293	6949.86	136.051
7404.79	187.123	6971.91	147.06	6894.86	173.146
7389.8	183.799	7163.12	189.494	6917.82	169.786
7423.34	100.638	7235.32	143.88	6902.57	169.951