

Universidade de Pernambuco (UPE)
Escola Politécnica de Pernambuco (POLI)
Programa de Pós-graduação em Engenharia da Computação (PPGEC)
Programa de Pós-graduação em Engenharia de Sistemas (PPGES)

Relatório da Prática de Inteligência de Enxames
ACS - Ant Colony System

Aluno: Carlos Henrique Maciel Sobral Timoteo

Professor: Dr. Carmelo Bastos Filho

17 de Junho de 2013

Lista de Figuras

4.1	Boxplot para o benchmark Att48	9
4.2	Boxplot para o benchmark Oliver30	10
4.3	Boxplot para o benchmark Eil51	10

Lista de Tabelas

2.1	Tipos de <i>benchmarks</i>	7
-----	--------------------------------------	---

Sumário

1	Introdução	4
1.1	Algoritmo ACS	4
1.2	Objetivo da medição	4
1.3	Objetivo do estudo	4
2	Planejamento	6
2.1	Benchmarks para otimização	6
2.1.1	Att48	6
2.1.2	Oliver30	6
2.1.3	Eil51	6
2.2	Tabela de Experimentos	7
3	Operação	8
4	Resultados	9
4.1	Resultados obtidos	9
4.2	Comparação de execução	10
5	Conclusão	12

Capítulo 1

Introdução

A motivação deste estudo é aprender como desenvolver um algoritmo de otimização por colônia de formigas, mais especificamente o *ACS* (do inglês *Ant Colony System*, Sistema de Colônia de Formigas), e adquirir alguma sensibilidade quanto aos parâmetros da técnica. O objeto de estudo deste trabalho é o algoritmo clássico do ACS e entender quais impactos os parâmetros tem no desempenho da técnica, tais como: influência do feromônio, influência da visibilidade, quantidade de cidades, número de formigas, quantidade de feromônio depositado, número máximo de iterações, quantidade de execuções, entre outros.

Alguns experimentos serão montados e, se necessário, um estudo estatístico será efetuado para melhorar a compreensão do que os parâmetros podem ajudar ou atrapalhar no funcionamento do algoritmo.

1.1 Algoritmo ACS

O ACS é um algoritmo inspirado no comportamento de uma colônia de formigas em busca de alimentos. Foi idealizado por Dorigo e Maniezzo em 1996 [1]. Um conjunto de trilhas marcadas por feromonios são possíveis soluções para o sistema, por meio do processo de deposição de feromônio nas trilhas e da escolha das possíveis trilhas de forma probabilística, é realizada, iterativamente, uma busca para encontrar uma "boa" solução para o problema. A solução é representada pelo melhor caminho encontrado pelas formigas no ambiente, representado por um grafo, evidenciado pela presença marcante de feromônio. Não é possível afirmar que a solução final é a melhor pois seria necessário uma exploração total do espaço de busca e um estudo de convergência para o algoritmo implementado.

1.2 Objetivo da medição

Serão realizadas três análises do desempenho do algoritmo ACS. A primeira análise é avaliar a evolução do comprimento da melhor viagem; a segunda análise é avaliar a evolução do *node branching*; a terceira análise é avaliar a média e desvio padrão do comprimento da melhor viagem encontrada ao final do algoritmo. A métrica de comparação será o melhor *fitness* por iteração (ou conjunto de iterações).

1.3 Objetivo do estudo

- Análise 1: Analisar a evolução do comprimento da melhor viagem

alpha = 1

beta = 5

quantidade de cidades = número de formigas

$Q = 100$

número máximo de iterações = 5000

Quantidade de execuções = 30

- Análise 2: Avaliar a evolução do *node branching*.

$\alpha = 1$

$\beta = 5$

quantidade de cidades = número de formigas

$Q = 100$

número máximo de iterações = 5000

Quantidade de execuções = 30

- Análise 3: Avaliar a média e desvio padrão do comprimento da melhor viagem encontrada ao final do algoritmo

$\alpha = 1$

$\beta = 5$

quantidade de cidades = número de formigas

$Q = 100$

número máximo de iterações = 5000

Quantidade de execuções = 30

- Funções para avaliação do desempenho:

Pseudo-Euclidian Travel Salesman Problem att48

Euclidian Travel Salesman Problem Oliver30

Euclidian Travel Salesman Problem Eil51

No segundo capítulo tem-se a descrição do planejamento dos experimentos assim como o levantamento de hipóteses. No Capítulo três apresenta-se como os testes serão realizados e os resultados comparados. No quarto e último capítulo apresentam-se os resultados obtidos e finaliza-se com as conclusões.

Capítulo 2

Planejamento

Neste capítulo descreve-se o planejamento que foi realizado para a realização dos experimentos.

2.1 Benchmarks para otimização

2.1.1 Att48

O benchmark Att48 é definido como um Pseudo-Euclidian Travel Salesman Problem (PETSP) com o qual constrói-se um grafo com 48 cidades, em que todas são ligadas com todas. O arquivo "att48.tsp" apresenta as coordenadas das cidades. O peso do grafo, representa a distância das cidades vizinhas, e é calculado utilizando a fórmula

$$f(\vec{x}, \vec{y}) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} / 10 \quad (2.1)$$

onde x_i, x_j, y_i, y_j indicam as coordenadas nas dimensões respectivas. Sendo a equação a distância entre cidades, o objetivo de se utilizar a técnica ACS será para minimizar, isto é, encontrar o percurso que apresenta a menor das distâncias entre cidades.

2.1.2 Oliver30

O benchmark Oliver30 é definido como um Euclidian Travel Salesman Problem (ETSP) com o qual constrói-se um grafo com 30 cidades, em que todas são ligadas com todas. O arquivo "oliver30.tsp" apresenta as coordenadas das cidades. O peso do grafo, representa a distância das cidades vizinhas, e é calculado utilizando a fórmula

$$f(\vec{x}, \vec{y}) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (2.2)$$

onde x_i, x_j, y_i, y_j indicam as coordenadas nas dimensões respectivas. Sendo a equação a distância entre cidades, o objetivo de se utilizar a técnica ACS será para minimizar, isto é, encontrar o percurso que apresenta a menor das distâncias entre cidades.

2.1.3 Eil51

O benchmark Eil51 é definido como um Euclidian Travel Salesman Problem (ETSP) com o qual constrói-se um grafo com 48 cidades, em que todas são ligadas com todas. O arquivo "eil51.tsp" apresenta as coordenadas das cidades. O peso do grafo, representa a distância das cidades vizinhas, e é calculado utilizando a fórmula (2.2).

Onde x_i, x_j, y_i, y_j indicam as coordenadas nas dimensões respectivas. Sendo a equação a distância entre cidades, o objetivo de se utilizar a técnica ACS será para minimizar, isto é, encontrar o percurso que apresenta a menor das distâncias entre cidades.

2.2 Tabela de Experimentos

Os parâmetros solicitados para serão variados nos experimentos são:

1. Influência do **feromonio** = 0, 0.5, 1, 2, 5
2. Influência do **visibilidade** = 0, 0.5, 1, 2, 5
3. Quantidade de **idades** = depende do benchmark
4. Quantidade de **idades** = quantidade de cidades
5. Influência da **evaporação** do feromonio = 0.3, 0.5, 0.7, 0.9, 0.999
6. Influência do **quantidade de feromônio** depositada = 1, 100, 1000

O critério de parada adotado para o algoritmo será o número de iterações que será fixado em 5000 (cinco mil).

Os *benchmarks* serão:

Tabela 2.1: Tipos de *benchmarks*

Benchmarks
Att48
Oliver30
Eil51

Temos no total 23 configurações diferentes para realizar experimentos. Cada experimento será realizado trinta vezes e, então, serão comparados.

Capítulo 3

Operação

O experimento consiste de três análises. Também será realizado o teste não paramétrico de Wilcoxon para comparar os casos em que uma simples inspeção visual não for suficiente para saber qual configuração foi melhor que a outra.

O tipo de erro utilizado no teste de hipótese não paramétrico de Wilcoxon é do Tipo II, o que significa que é o caso em que a hipótese nula é aceita, mesmo que realmente ela não seja verificada. Na prática, aceitam-se os falsos-positivos.

O mesmo computador será utilizado na realização de todos os testes, variando apenas o *seed* para o gerador pseudo-aleatório e os parâmetros do ACS. Como se trata do mesmo algoritmo sendo executado antes e depois de algumas mudanças, pode-se realizar o pareamento dos dados sem problemas.

Como saída do programa, tem-se o melhor resultado já visitado pelo enxame para cada 200 iterações. A configuração do computador utilizado para realizar os experimentos é: Intel Core 2 Duo 2,00GHz, 4GB de memória RAM, sistema operacional Windows 7.

Cada configuração foi executada trinta vezes, no mesmo computador. Apesar do computador possuir mais de um núcleo, o programa desenvolvido não utilizou disso para acelerar seu processamento, executando assim o seu código em apenas um núcleo.

Dorigo e Maniezzo[1] realizaram um estudo paramétrico parecido com um dos objetivos desse estudo. Nele foram definidos os seguintes parâmetros ótimos:

1. Influência do **feromonio** = 1
2. Influência do **visibilidade** = 5
3. Quantidade de **idades** = numero de **formigas**
4. Influência da **evaporação** do feromonio = 0.5
5. Influência do **quantidade de feromônio** depositada = 100

Portanto, evitaremos o retrabalho e produziremos somente três experimentos. Um para cada benchmark, utilizando os parâmetros otimizados fornecidos por Dorigo.

Capítulo 4

Resultados

Este capítulo apresenta os resultados obtidos para as 3 configurações investigadas. Nesse caso, utilizamos as primeiras 1000 iterações, porque foi verificado uma variação muito pequena para as demais iterações.

4.1 Resultados obtidos

As figuras a seguir são o *boxplot* de cada configuração.

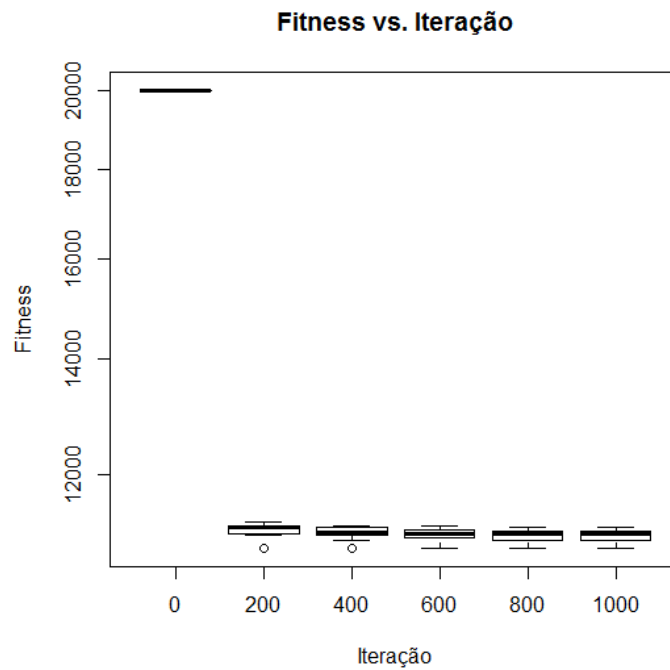


Figura 4.1: Boxplot para o benchmark Att48

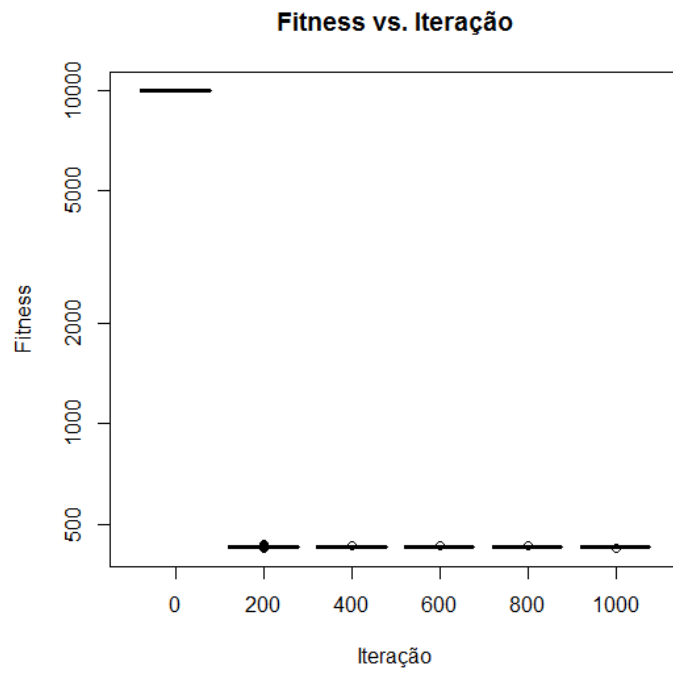


Figura 4.2: Boxplot para o benchmark Oliver30

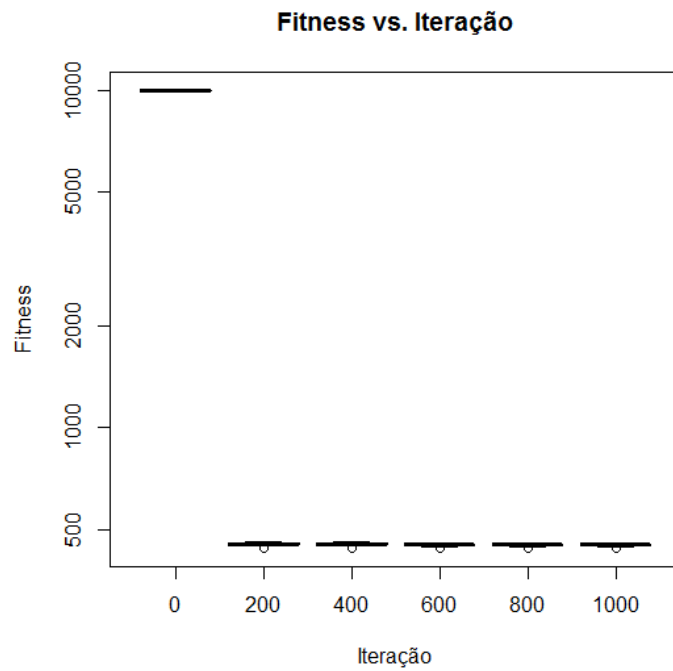


Figura 4.3: Boxplot para o benchmark Eil51

4.2 Comparação de execução

Como foram selecionadas as melhores configurações de acordo com Dorigo e Maniezzo[1], não exibimos o Teste de Wilcoxon Pareado. Utilizando a linguagem R teríamos um código assim:

```
oliver30_i <- read.table (file = 'C:\\Users\\carlos\\Desktop\\ACO\\oliver30_i.  
eil51_i <- read.table (file = 'C:\\Users\\carlos\\Desktop\\ACO\\eil51_i.txt',  
att48_i <- read.table (file = 'C:\\Users\\carlos\\Desktop\\ACO\\att48_i.txt',  
wilcox.test(oliver30_i$FITNESS, oliver30_j$FITNESS, alternative = "greater", p
```

Capítulo 5

Conclusão

O algoritmo ACS é bastante utilizado para otimizar problemas combinatoriais discretos. Além do algoritmo clássico há variações da técnica. Para este trabalho realizou-se um estudo de parametrização a partir das informações encontradas em Dorigo e Maniezzo[1]. Observou-se que todas as configurações foram capazes de convergir apesar de fazê-lo de modo diferente.

De modo geral, é possível utilizá-lo para resolver problemas que podem ser representados por um grafo e cujo objetivos podem ser a minimização de custos de trajetos, distribuição de tráfego.

Referências Bibliográficas

- [1] V. Dorigo, M.; Maniezzo. Ant system: optimization by a colony of cooperating agents. *IEEE transaction on systems man and cybernetics Part B Cybernetics a publication of the IEEE System Man and Cybernetics Society*, 1996.