

Trabalho Prático II

Colônia de formigas

Computação Natural

Gabriel Victor Carvalho Rocha

`gabrielcarvalho@dcc.ufmg.br` - 2018054907

1. Modelo

O problema foi modelado em um grafo utilizando um *dict*, onde cada *key* é um dos seus vértices representado em uma tupla (*trabalho*, *maquina*) e mapeia uma lista de vértices que corresponde a suas arestas. Esse grafo possui dois vértices adicionais que são utilizados para iniciar e finalizar o caminho, ou seja, o vértice de início possui arestas para a primeira máquina de cada trabalho e os vértices finais de cada trabalho possuem arestas para o vértice final. Além disso, os vértices que representam uma operação de um trabalho, possuem arestas direcionadas para a máquina subsequente e arestas bidirecionais para cada operação que utiliza a mesma máquina. As arestas bidirecionais são representadas diferente do apresentado anteriormente, para diferenciá-las foi utilizada a representação (*trabalho*, *maquina*, 'bd').

Para mapear os feromônios, foi criada uma matriz de adjacência iniciada com o valor do feromônio inicial (exceto sua diagonal, que possui -1). Um detalhe importante para realizar uma busca nessa matriz, é que dado um vértice (*trab*, *maq*), para achar seu índice basta realizar a operação $trab * qtd_maq + maquinas[trab].index(maq)$, onde o último valor indica em que posição está a máquina no trabalho. Seguindo a mesma ideia, porém sendo apenas uma lista, é mapeado o *desirability* de cada vértice, que representa o tempo restante para finalizar totalmente o trabalho.

Toda essa estrutura está contida dentro da classe *Colonia*, que além do supracitado, possui também os valores da quantidade de máquinas, quantidade de trabalhos e uma lista da classe *Formiga*. Dentro da classe *Formiga* há apenas o caminho gerado pela formiga e o seu *makespan*.

Quando a função *execucao* é chamada, temos o início do caminhamento. Para cada formiga da lista, ela irá decidir um vértice para prosseguir entre os possíveis, essa escolha é feita a partir da probabilidade calculada para cada um desses vértices, que leva em conta seu *desirability* e a quantidade de feromônio entre o vértice atual e ele. Isso é feito até que não tenha mais vértices na lista de próximos, encerrando o loop do caminhamento e entrando para o

cálculo do *makespan*.

Primeiramente é necessária a criação de um grafo dirigido a partir do grafo inicial citado anteriormente. Ou seja, dado um caminho de uma formiga, para cada vértice do grafo, quando houver um vértice *A* que possui aresta para um vértice *B* com a flag '*bd*', e o vértice *A* é precedente em relação ao vértice *B* no caminho, então é retirada a flag '*bd*' dessa aresta para *B* e o vértice *A* é removido das arestas de *B*. Caso *B* seja precedente em relação ao *A*, a operação oposta é realizada. Para calcular o *makespan*, foi utilizada a ideia apresentada no artigo *Ant system for Job-shop scheduling*, encontrando a partir do grafo dirigido (e acíclico) o maior caminho do vértice inicial até o final. Em outras palavras, podemos resumir o objetivo do trabalho em encontrar um caminho gerado por uma formiga que minimize o maior caminho entre o vértice inicial e o vértice final do grafo.

Chegando na função em que se atualiza o feromônio, na primeira parte é evaporado de todas as arestas a taxa definida como parâmetro, após isso é calculado o feromônio da melhor formiga até o momento pela operação $1 / \text{formiga.makespan}$, ou seja, a criação de mais feromônio é feita de forma elitista pela melhor formiga encontrada até o momento, depositando a quantidade calculada anteriormente em todas arestas que representam o seu caminho.

2. Experimentos

A metodologia utilizada na experimentação utilizou as bases de dados *ft06.txt* e *la01.txt*, essas foram escolhidas por terem um tamanho menor, sendo assim mais rápidas para executar as variações dos parâmetros. Os parâmetros foram variados nos seguintes intervalos, sendo executados 30 vezes para cada variação e selecionados os que obtiveram menores médias:

- ❖ Número de formiga: 25, 50 e 100
- ❖ Iterações: 25, 50 e 100
- ❖ Feromônio inicial: 0.001, 0.01 e 0.1
- ❖ Taxa de evaporação: 0.1, 0.25 e 0.5
- ❖ Alfa: 0.5, 1, 2 e 5
- ❖ Beta: 0.5, 1, 2 e 5

a. Arquivo *ft06.txt*

Para *ft06.txt* foi fixado inicialmente o número de iterações em 25, taxa de evaporação em 0.25, feromônio inicial em 0.01, alfa e beta em 1.

- ❖ Variando o número de formigas no intervalo descrito acima, se obteve um melhor makespan utilizando o parâmetro 100 com média 58.7, contra 60.8 e 59.2 dos parâmetros 25 e 50, respectivamente.
- ❖ Variando o número de iterações, se obteve um melhor makespan utilizando o parâmetro 50 com média 58.6, contra 59.2 e 59.4 dos parâmetros 25 e 100, respectivamente.
- ❖ Variando o feromônio inicial, se obteve um melhor makespan utilizando o parâmetro 0.1 com média 57.7, contra 60.0 e 59.3 dos parâmetros 0.001 e 0.01, respectivamente.
- ❖ Variando a taxa de evaporação, se obteve um melhor makespan utilizando o parâmetro 0.1 com média 58.7, contra 59.3 e 58.9 dos parâmetros 0.25 e 0.5, respectivamente.
- ❖ Variando o alfa, se obteve um melhor makespan utilizando o parâmetro 1 com média 58.2, contra 58.9, 58.8 e 58.5 dos parâmetros 0.5, 2 e 5, respectivamente.
- ❖ Variando o beta, se obteve um melhor makespan utilizando o parâmetro 2 com média 55.8, contra 58.0, 58.1 e 56.5 dos parâmetros 0.5, 1 e 5, respectivamente.

Logo, os melhores parâmetros encontrados para *ft06.txt* foram: Número de formigas: 100; Número de iterações: 50; Feromônio inicial: 0.1; Taxa de evaporação: 0.1; Alfa: 1; e Beta: 2. Formando assim o primeiro grupo de parâmetros.

b. Arquivo la01.txt

Para *la01.txt*, foi novamente fixado o número de iterações em 25, a taxa de evaporação em 0.25, feromônio inicial em 0.01, alfa e beta em 1.

- ❖ Variando o número de formigas no intervalo descrito acima, se obteve um melhor makespan utilizando o parâmetro 100 com média 719.2, contra 742.0 e 729.1 dos parâmetros 25 e 50, respectivamente.
- ❖ Variando o número de iterações, se obteve um melhor makespan utilizando o parâmetro 25 com média 719.3, contra 723.0 e 722.4 dos parâmetros 50 e 100, respectivamente.
- ❖ Variando o feromônio inicial, se obteve um melhor makespan utilizando o parâmetro 0.01 com média 709.6, contra 715.5 e 728.2 dos parâmetros 0.001 e 0.1, respectivamente.
- ❖ Variando a taxa de evaporação, se obteve um melhor makespan utilizando o parâmetro 0.25 com média 722.2, contra 732.3 e 734.1 dos parâmetros 0.1 e 0.5, respectivamente.
- ❖ Variando o alfa, se obteve um melhor makespan utilizando o parâmetro 1 com média 713.0, contra 738.9, 729.4 e 735.6 dos parâmetros 0.5, 2 e 5, respectivamente.

- ❖ Variando o beta, se obteve um melhor makespan utilizando o parâmetro 5 com média 673.8, contra 735.6, 725.3 e 685.9 dos parâmetros 0.5, 1 e 2, respectivamente.

Logo, os melhores parâmetros encontrados para *la01.txt* foram: Número de formigas: 100; Número de iterações: 25; Feromônio inicial: 0.01; Taxa de evaporação: 0.25; Alfa: 1; e Beta: 5. Formando assim o segundo grupo de parâmetros.

3. Resultados

Abaixo seguem os resultados obtidos por cada instância utilizando os parâmetros encontrados anteriormente. A média e o desvio padrão foram extraídos de 20 execuções.

a. Arquivo *ft06*

Utilizando os primeiros parâmetros, foram obtidos os seguintes resultados:

- ❖ Média das soluções: 56.7
- ❖ Desvio padrão das soluções: 1.4866
- ❖ Quantidade de ótimos encontrados: 7

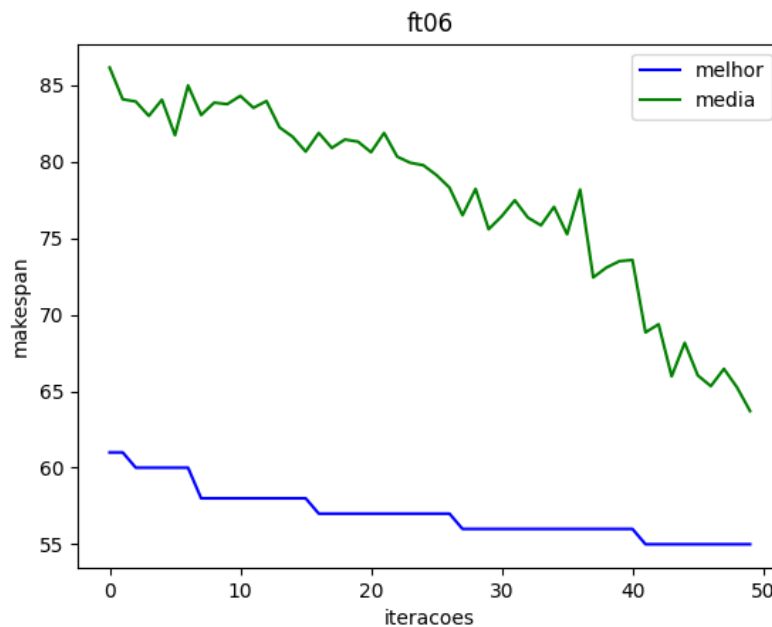


Gráfico 1: Resultado de uma execução do *ft06* utilizando os primeiros parâmetros.

Utilizando os segundos parâmetros, foram obtidos os seguintes resultados:

- ❖ Média das soluções: 57.1
- ❖ Desvio padrão das soluções: 1.04403
- ❖ Quantidade de ótimos encontrados: 2

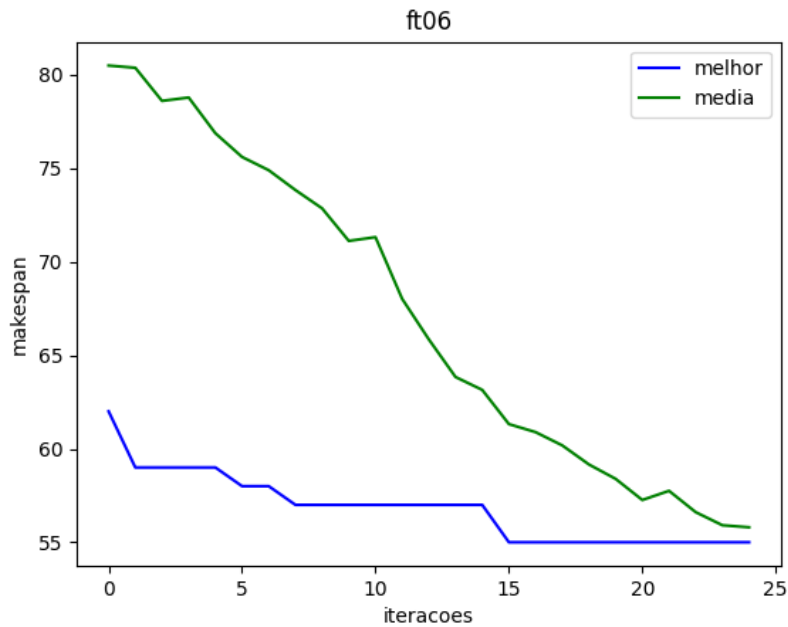


Gráfico 2: Resultado de uma execução do *ft06* utilizando os segundos parâmetros.

Como o *ft06* possui um conjunto de menor, é possível observar sua convergência para o valor ótimo, tanto pelos parâmetros obtidos por *la01*, quanto para o encontrado para ele mesmo, possuindo maior número de soluções ótimas.

b. Arquivo *la01*

Utilizando os primeiros parâmetros, foram obtidos os seguintes resultados:

- ❖ Média das soluções: 706.45
- ❖ Desvio padrão das soluções: 17.1682
- ❖ Diferença do melhor para o ótimo: 2

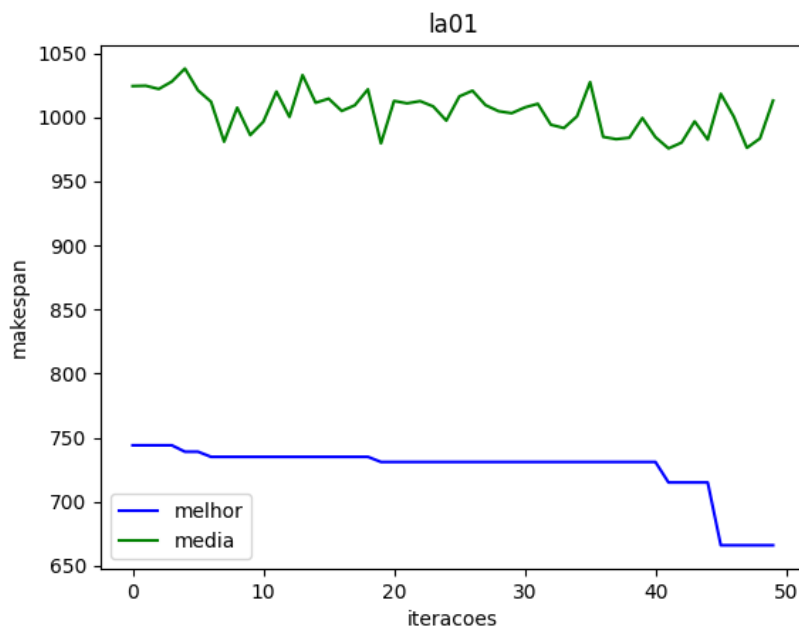


Gráfico 3: Resultado de uma execução do *la01* utilizando os primeiros parâmetros.

Utilizando os segundos parâmetros, foram obtidos os seguintes resultados:

- ❖ Média das soluções: 676.6
- ❖ Desvio padrão das soluções: 10.6634
- ❖ Quantidade de ótimos encontrados: 7

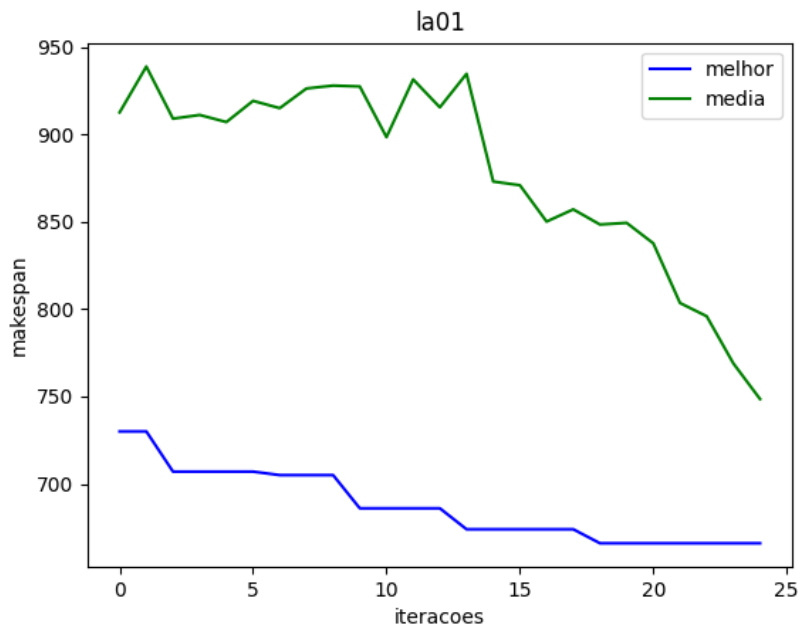


Gráfico 4: Resultado de uma execução do *la01* utilizando os segundos parâmetros.

O mesmo que foi concluído anteriormente, pode-se afirmar aqui. É possível ver a convergência em ambos conjuntos de parâmetros, porém o que foi encontrado pelo próprio *la01* obteve um número maior de soluções ótimas.

c. Arquivo *la29*

Utilizando os primeiros parâmetros, foram obtidos os seguintes resultados:

- ❖ Média das soluções: 1706.9
- ❖ Desvio padrão das soluções: 30.5448
- ❖ Diferença do melhor para o ótimo: 509

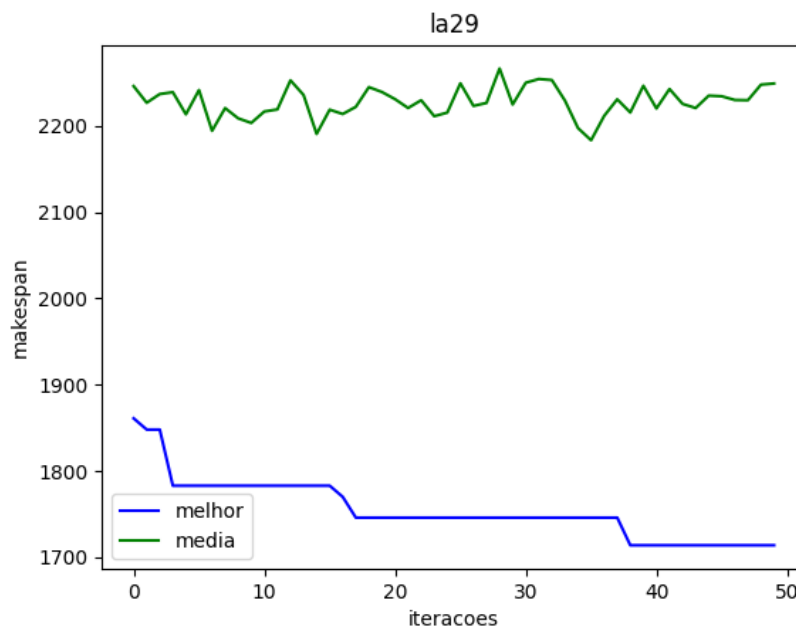


Gráfico 5: Resultado de uma execução do *la29* utilizando os primeiros parâmetros.

Utilizando os segundos parâmetros, foram obtidos os seguintes resultados:

- ❖ Média das soluções: 1659.25
- ❖ Desvio padrão das soluções: 17.0348
- ❖ Diferença do melhor para o ótimo: 471

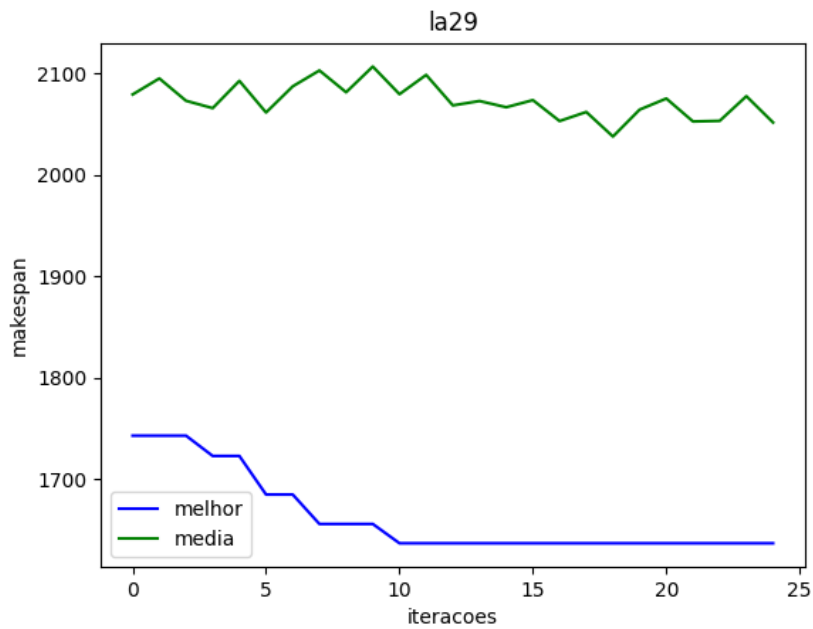


Gráfico 5: Resultado de uma execução do *la29* utilizando os segundos parâmetros.

Já a instância *la29*, por possuir um tamanho maior e como os parâmetros foram “calibrados” em instâncias diferentes, obteve um valor não muito bom. O conjunto de parâmetros encontrados na instância *la01* obteve os melhores resultados, diferenciando em 471 do valor ótimo.

d. Arquivo *la40*

Utilizando os primeiros parâmetros, foram obtidos os seguintes resultados:

- ❖ Média das soluções: 1825.95
- ❖ Desvio padrão das soluções: 40.3589
- ❖ Diferença do melhor para o ótimo: 529

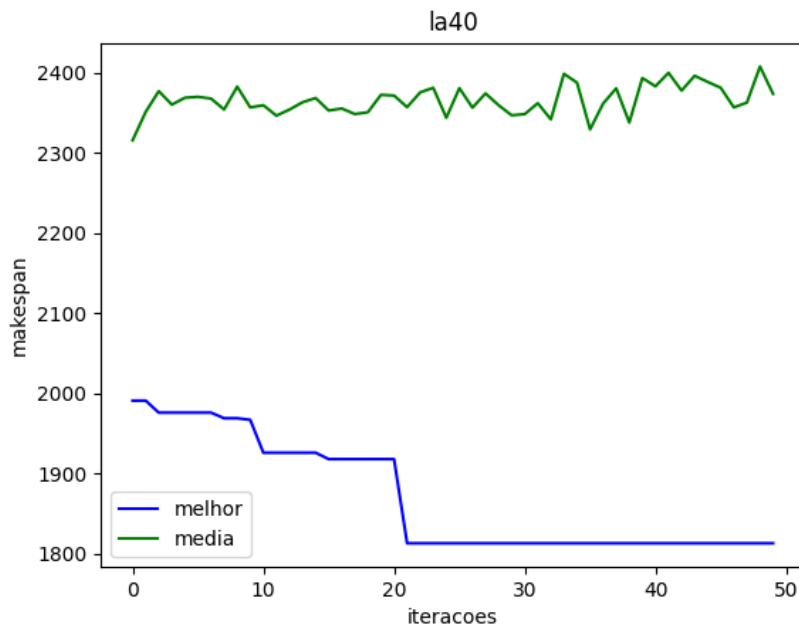


Gráfico 6: Resultado de uma execução do *la40* utilizando os primeiros parâmetros.

Utilizando os segundos parâmetros, foram obtidos os seguintes resultados:

- ❖ Média das soluções: 1809.95
- ❖ Desvio padrão das soluções: 26.72728
- ❖ Diferença do melhor para o ótimo: 523

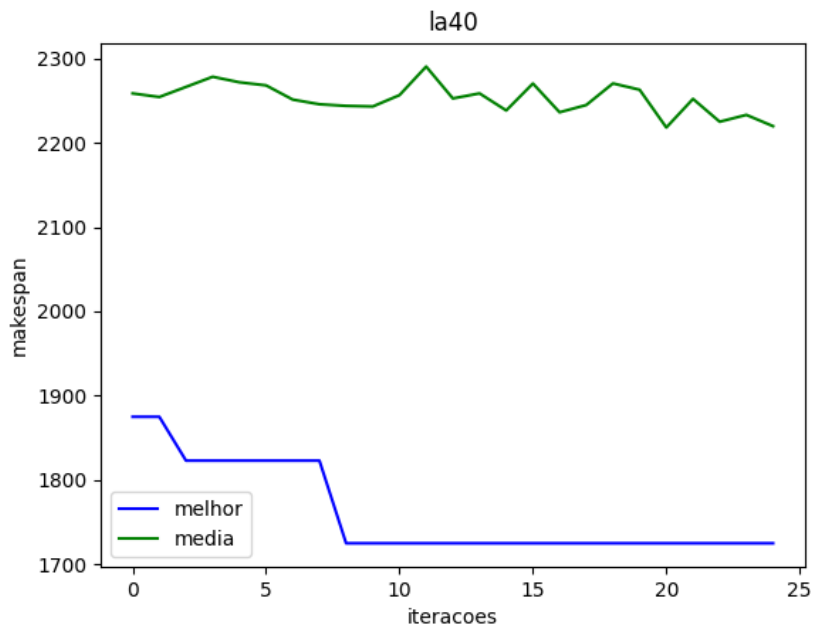


Gráfico 7: Resultado de uma execução do *la40* utilizando os segundos parâmetros.

Como no caso anterior, a instância *la40* também não obteve um resultado tão bom quanto os outros. Obteve melhores resultados com o segundo conjunto de parâmetros, se diferenciando em 523 do valor ótimo.

4. Conclusão

Dessa forma, após concluir o trabalho ficou claro o funcionamento do algoritmo ACO, possibilitando a visualização da convergência dos valores de *makespan* após cada nova iteração do caminharmento das formigas sobre o grafo.

Porém, foi necessário o uso do elitismo, onde somente a melhor formiga deposita o feromônio, pois sem o uso do mesmo não estava obtendo uma convergência adequada. Com isso, acabou-se perdendo bastante *exploration*, causando convergências mais rápidas.

5. Bibliografia

PAPPA, Gisele. Slides - Introdução à Inteligência Coletiva (de Enxames). PDF disponibilizado via Moodle UFMG.

PAPPA, Gisele. Slides - Colônia de Formigas. PDF disponibilizado via Moodle UFMG.

PAPPA, Gisele. Slides - Aplicações de ACO. PDF disponibilizado via Moodle UFMG.

COLORNI, Alberto; DORIGO, Marco; MANIEZZO, Vittorio; TRUBIAN, Marco. **Ant system for Job-shop scheduling**. Belgian Journal of Operations Research, Statistics and Computer Science. Disponível em: <https://www.orbel.be/jorbel/index.php/jorbel/article/view/169/125>.