

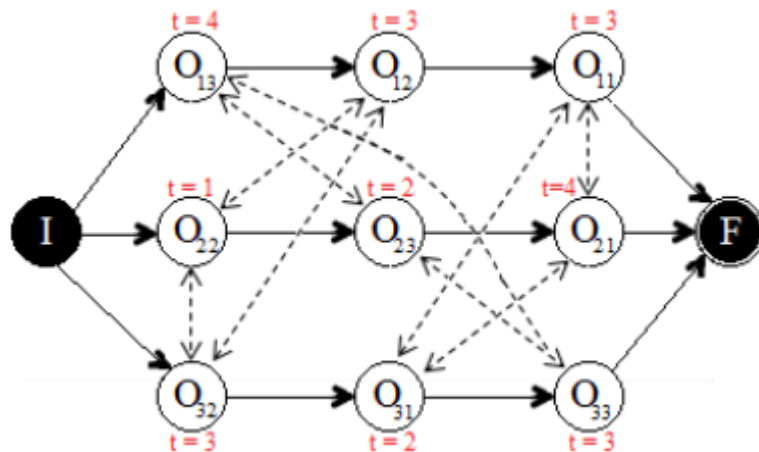
Trabalho Prático 2 - ACO para o Job-Shop Scheduling Problem

Júlia Fonseca de Sena - 2018054508

1. Modelo

A modelagem foi feita com um grafo disjuntivo, ou seja, com ambas arestas direcionadas e não direcionadas. Cada nó representa uma operação, operações de um mesmo job são conectadas por arestas direcionadas na sequência em que devem ser realizadas, e as que serão feitas em uma mesma máquina são ligadas por arestas não direcionadas. Além disso, há dois nós de início e fim que não representam operações. O início possui arestas direcionadas para as primeiras operações de todos os jobs, e as últimas possuem arestas direcionadas para o fim.

Para a construção das soluções, foi criada a classe Ant que simula uma formiga que anda pelo grafo criando uma solução possível. Ela anda entre arestas conectadas, direcionadas ou não direcionadas (desde que estas estejam prontas para serem processadas, quando as operações anteriores do job já tiverem sido feitas) até encontrar uma solução com todas as operações acessadas. Quando essa condição ainda não foi alcançada e a única aresta possível a partir de um nó é a que leva para o fim, é escolhida uma das operações prontas para serem feitas aleatoriamente. A figura abaixo ilustra esse grafo e suas relações, sendo a operação O13 do job 1 feita na máquina 3.



A maior dificuldade de implementação foi fazer uma função makespan correta. Houve dificuldade em abstrair a lógica necessária no código, mas se obteve sucesso com a realização de pesquisas. Quanto a decisões de implementação, foi escolhido o LRT (que dá mais valor às arestas que ligam a nós com jobs que tem maior tempo restante) como heurística por ser mais eficiente para encontrar o makespan mais baixo de acordo com algumas pesquisas. Além disso, a fim de garantir que as soluções passam por todas as operações, na criação foi implementado um array com as operações possíveis de cada job (cada job só possui uma operação pronta para ser utilizada de cada vez, já que devem ser realizadas em sequência), de forma que, cada vez que a formiga chega em um nó que só pode levar ao fim mas ainda não visitou todos os outros, escolhe um nó aleatório desse vetor para ser o próximo.

2. Metodologia

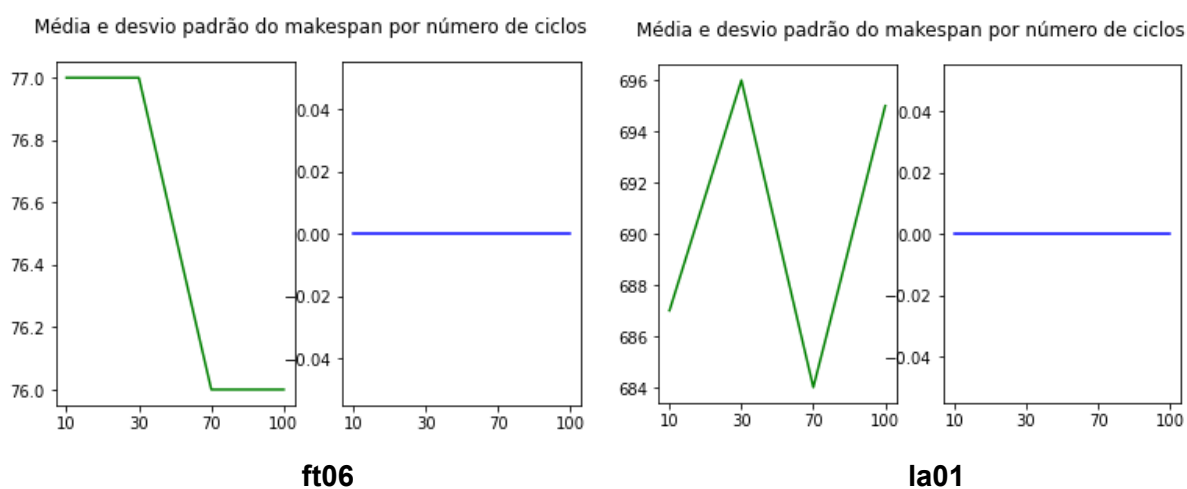
Foram feitas 30 repetições do algoritmo para a escolha dos parâmetros, e plotado gráficos com média e desvio padrão dos melhores resultados (obtidos em cada repetição) para cada parâmetro. A seguir, os números utilizados para cada parâmetro:

- Número de ciclos: 10, 30, 70 e 100; mantendo-se o número de formigas como o número de arestas, alfa como 0.4, beta como 0.6 e evaporação como 0.7 e feromônio inicial como 0.2
- Número de formigas: 10, 30, 70 e o número de arestas do grafo; os parâmetros fixos foram 70 ciclos, alfa como 0.4, beta como 0.6 e evaporação como 0.7 e feromônio inicial como 0.2
- Feromônio inicial: 0.02, 0.05, 0.1, 0.2; com 70 números de ciclos, número de formigas como número de arestas, alfa como 0.4, beta como 0.6 e evaporação como 0.7.
- Alfa e beta: 0.2 e 0.8, 0.4 e 0.6, 0.6 e 0.4, 0.8 e 0.2; com 70 ciclos, número de formigas como número de arestas, evaporação como 0.7 e feromônio inicial como 0.02.
- Evaporação: 0.3, 0.5, 0.7, 0.9; com 70 ciclos, número de formigas como o número de arestas, alfa como 0.4, beta como 0.6 e feromônio inicial como 0.02

Essas análises de parâmetros foram feitas nas instâncias ft06 e la01. Ambas são as menores das 4, o que agiliza a experimentação e permite que testes mais profundos sejam feitos (mais repetições, por exemplo). Além disso, la01 não possui o mesmo número de jobs e de máquinas (assim como la29), então é a escolha diferencial de menor escala.

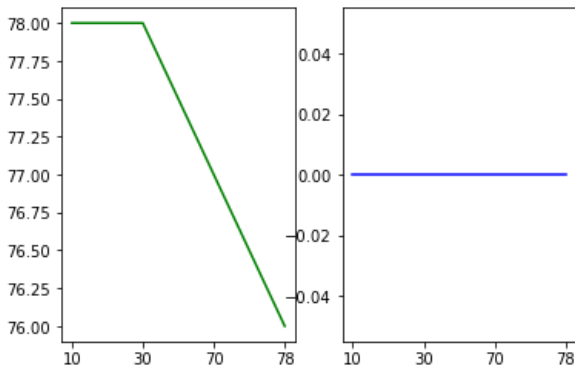
3. Experimentos

Número de ciclos



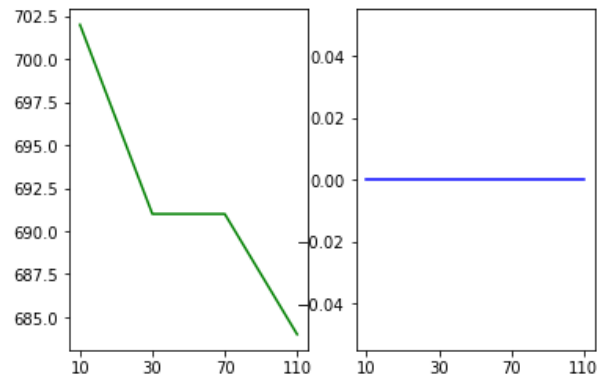
Número de formigas

Média e desvio padrão do makespan por número de formigas



ft06

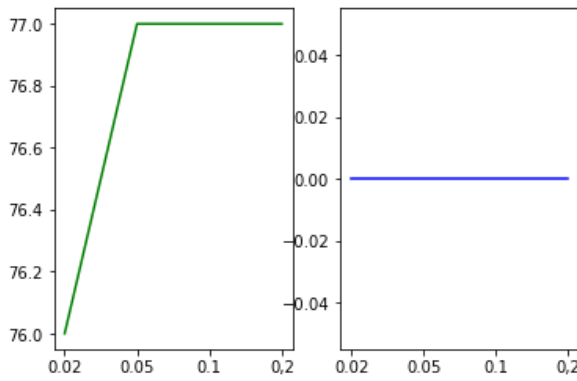
Média e desvio padrão do makespan por número de formigas



la01

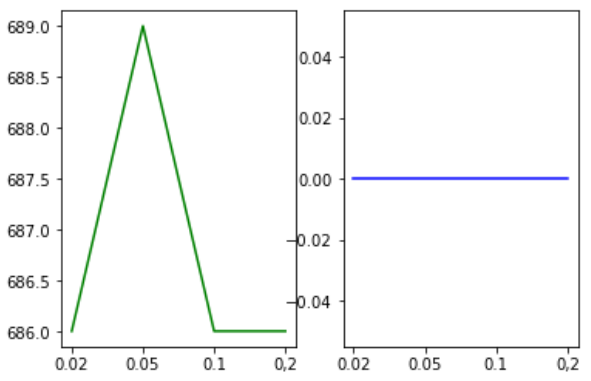
Feromônio inicial

Média e desvio padrão do makespan por feromônio inicial



ft06

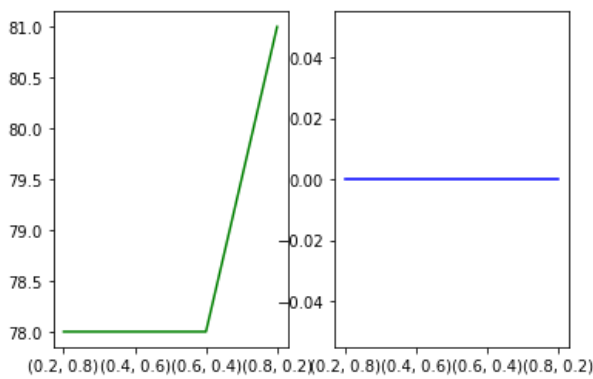
Média e desvio padrão do makespan por feromônio inicial



la01

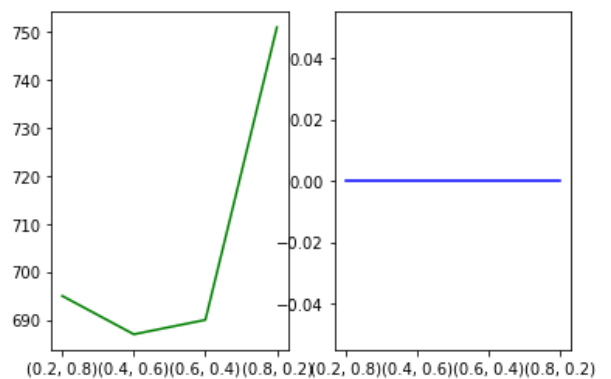
Alfa e beta

Média e desvio padrão do makespan por alfa e beta



ft06

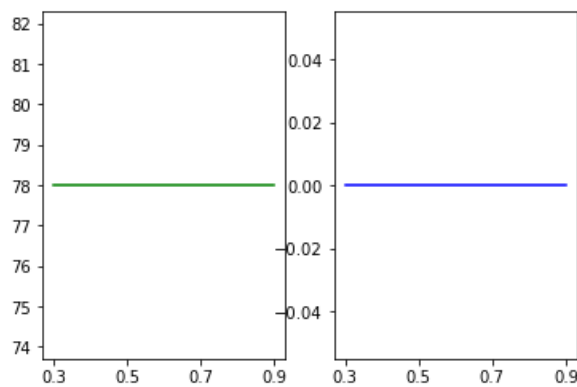
Média e desvio padrão do makespan por alfa e beta



la01

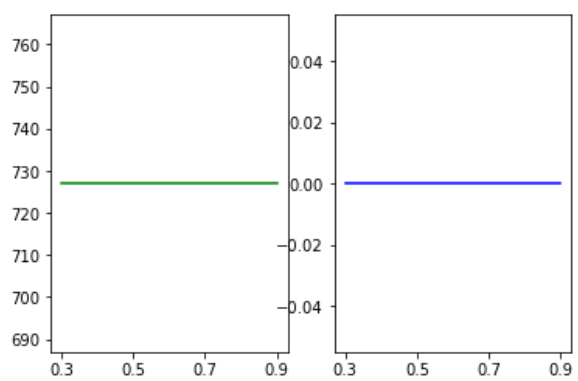
Evaporação

Média e desvio padrão do makespan por taxa de evaporação



ft06

Média e desvio padrão do makespan por taxa de evaporação



la01

Os valores ótimos não foram encontrados em nenhuma das instâncias, com nenhuma variação de parâmetros.

4. Resultados

Iniciando pelos experimentos com número de ciclos, ambas instâncias tiveram seus menores makespans com 70. Surpreendentemente, na la01 houve um aumento considerável quando os ciclos subiram para 100, mas isso pode ser somente devido ao fato do algoritmo não ser determinístico. Isso também pode explicar o aumento do 10 para o 30. De qualquer modo, em ambos contextos, 70 foi a opção com melhor desempenho, por gastar menos tempo que 100 ciclos e ainda manter os valores mais baixos.

Partindo para a decisão do número de formigas, o último valor adotado foi o número de arestas do grafo (direcionadas e não direcionadas). É um número personalizado para o tamanho de cada instância, então o custo aumenta bastante junto com o tamanho da instância, mas apresentou desempenho melhor em todos os testes, então decidiu-se por adotá-lo de qualquer forma.

O terceiro parâmetro analisado foi o valor do feromônio inicial das arestas. Na instância maior, não houve muita diferença, mas na ft06 somente 0.02 obteve um valor menor, levando a conclusão que o valor inicial pode ter mais influência em instâncias pequenas. De todo modo, tal valor apresentou resultados bons para ambos ft06 e la01, então foi selecionado como feromônio inicial.

Seguindo para a variação dos valores de alfa e beta, sendo que $\alpha + \beta = 1$. O aumento do alfa aumenta a importância dada para o feromônio na escolha de uma aresta a seguir pela formiga, e do beta aumenta do LRT. A escolha de valores que apresentou melhor resultado foi $\alpha = 0.4$ e $\beta = 0.6$ em ambas instâncias, então entende-se que tender a seguir para operações dos trabalhos com maior tempo restante de processamento traz makespans melhores, contudo com algum equilíbrio com o feromônio. Portanto, deve-se levar em conta ambos fatores, mas dando leve maior importância para o LRT.

A última análise foi da taxa de evaporação. A média permaneceu a mesma para todos os valores nas duas instâncias, surpreendentemente, mesmo após re-execuções para ver se surgiam resultados diferentes. Isso sugere que tem baixíssima influência no melhor makespan encontrado. Decidiu-se, então, manter o 0.7 após algumas pesquisas que indicaram que em outros experimentos obteve o melhor desempenho.

O desvio padrão, estranhamente, permaneceu 0 para todos os experimentos. Testando a função utilizada para calculá-lo (numpy.std), é funcional com outros arrays. A única explicação possível é que o desvio padrão entre as melhores soluções obtidas nas 30 repetições foi pequeno a ponto de ser arredondado para 0 pela função. Isso dificultou a utilização dessa estatística na análise.

Quanto aos resultados obtidos na execução de cada instância com os parâmetros selecionados acima (70 ciclos, número de formigas igual ao número de arestas, feromônio inicial 0.02, alfa 0.4, beta 0.6 e taxa de evaporação 0.7), foram os seguintes:

	Melhor makespan	Distância do ótimo	Chegou no ótimo	Média por ciclo	Desvio padrão por ciclo
ft06	76	21	0 vezes	78.31428571428572	1.2368491898236207
la01	684	18	0 vezes	713.7428571428571	10.94490842392769
la29	1404	247	0 vezes	1436.3285714285714	9.935105763433636
la40	2019	797	0 vezes	2066.1857142857143	13.793676457133598

5. Conclusão

O trabalho foi uma ótima forma de abstrair a teoria do ACO para a prática. Isso foi feito bem mais facilmente que no trabalho anterior, acredito que pela maior experiência atual com algoritmos de Computação Natural. A visão de como a escolha de um caminho por formigas, de acordo com feromônios, se aplica a problemas da computação, em particular ao JSSP, foi bem melhor entendida a partir da leitura dos artigos passados como referência e com a implementação de uma versão própria da solução.

6. Referências

- FLÓREZ, Edson and GÓMEZ, Wilfredo and BAUTISTA, Lola. An Ant Colony Optimization Algorithm For Job Shop Scheduling Problem.
- COLORNI, Alberto and DORIGO, Marco and MANIEZZO, Vittorio and TRUBIAN, Marco. Ant system for job-shop scheduling. STATISTICS AND COMPUTER SCIENCE. 1994.