

Trabalho Prático 2

Data de Entrega: 10/03/2021

ACO para o problema do *Job-Shop Scheduling*

O principal objetivo deste trabalho é implementar um algoritmo de colônia de formigas (ACO) para resolver o *Job-Shop Scheduling* (JSSP) [1]. O JSSP é um problema de otimização combinatorial NP-hard, e é definido da seguinte forma. Considere um conjunto de n tarefas (*jobs*) e um conjunto de m máquinas, onde cada tarefa é composta por um conjunto de operações cuja ordem de execução é pré-definida. Para cada operação, conhecemos em que máquina ela deve ser executada e o tempo necessário para que ela seja finalizada. Existem algumas restrições em relação as tarefas e máquinas, tais como:

- Uma tarefa não pode ser atribuída a uma mesma máquina duas vezes.
- Não existe uma ordem de execução em relação a operações de tarefas diferentes.
- Uma operação não pode ser interrompida.
- Cada máquina pode processar apenas uma tarefa (operação) de cada vez.

O objetivo do JSSP é distribuir as tarefas entre as máquinas de forma que o tempo de processamento necessário para terminar a execução de todas elas (também conhecido como *makespan*) seja o menor possível. Em outras palavras, queremos determinar a sequência de operações nas máquinas que minimizem o *makespan*.

Considere o exemplo ilustrado na Tabela 1, onde são consideradas 3 tarefas para serem distribuídas em 3 máquinas. Cada tarefa tem 3 operações, e seus tempos de execução e máquinas onde devem ser executadas são descritos na primeira e segunda partes da tabela, respectivamente.

Tabela 1: Exemplo de um problema com 3 tarefas e 3 máquinas.

Tarefa	Operações		
	1	2	3
Tempo de Processamento			
j_1	3	3	2
j_2	1	5	3
j_3	3	2	3
Sequência de máquinas			
j_1	m_1	m_2	m_3
j_2	m_1	m_3	m_2
j_3	m_2	m_1	m_3

A Fig. 1 mostra um exemplo de solução para o problema com *makespan* 12. Sua tarefa é modelar este problema usando um ACO. Na modelagem do ACO deve-se considerar a representação do problema, a regra de transição probabilística, como avaliar as soluções e como atualizar o feromônio.

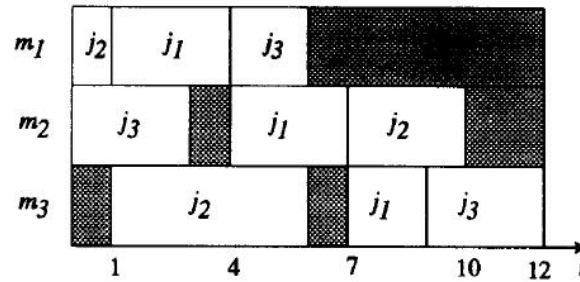


Figura 1: Exemplo de solução para o problema definido na Tabela 1.

Experimentos

O ACO deve ser testado em 4 instâncias do JSSP, descritas na Tabela 2 e disponíveis em <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/jobshop1.txt>

Tabela 2: Instâncias do problema a serem consideradas

Instância	Jobs x Máquinas	<i>Makespan</i> Ótimo
ft06	6x6	55
la01	10x5	666
la29	20x10	1157
la40	15x15	1222

Da mesma forma que no TP1, alguns parâmetros precisam ser calibrados: número de iterações (normalmente menor que o dos algoritmos evolucionários), número de formigas, taxa de feromônio inicial, taxa de evaporação, entre outros. Use seu bom senso e a experiência com o TP1 para fazer um bom planejamento experimental.

Lembrem-se que ao mexer em um dos parâmetros, todos os outros devem ser mantidos constantes, e que a análise dos parâmetros é de certa forma interativa. A configuração de parâmetros raramente vai ser ótima, mas pequenos testes podem melhorar a qualidade das soluções encontradas.

As análises detalhadas dos parâmetros devem ser feitas em apenas 2 das 4 instâncias do problema (justifique a escolha das instâncias). Os parâmetros considerados mais apropriados para cada uma dessas execuções devem ser novamente utilizados para as outras instâncias do problema. Reporte, para cada instância, a média e desvio padrão do *makespan* das soluções encontradas, as diferenças para o ótimo e, caso você encontre o ótimo, em quantas das execuções do seu algoritmo isso aconteceu.

1 O que deve ser entregue

- Código fonte
- Relatório, que deve descrever:
 1. **O modelo:** descreva claramente o seu modelo, as dificuldades e as decisões de projetos adotadas. Explique cada decisão.
 2. **Metodologia:** descreva como foram feitos os experimentos: quantidade de execuções (recomenda-se 30), parâmetros adotados em cada experimento, etc...
 3. **Experimentos:** apresente os experimentos realizados.

4. **Resultados:** explique os resultados obtidos pelos experimentos. Comentários óbvios do tipo “Podemos ver pela figura 2 que a função a é maior que a função b .” são desnecessários. Apresente conclusões obtidas, facilite o trabalho do leitor: “A figura 2 mostra que no contexto X a função a é sempre superior. Isso comprova nossa tese ...”
5. **Conclusão:** O que você aprendeu sobre ACO após fazer este trabalho?

A entrega DEVE ser feita pelo Moodle na forma de um único arquivo zipado, contendo o código e a documentação do trabalho.

Referências

- [1] Christian Blum and Michael Sampels. An ant colony optimization algorithm for shop scheduling problems. *J. Math. Model. Algorithms*, 3:285–308, 09 2004.