

Trabalho de Implementação – Algoritmos Bioinspirados

Carolina Ribeiro Xavier

1 Introdução

O problema de agendamento de tarefas é um dos mais estudados na pesquisa operacional e na ciência da computação, com vasta aplicação industrial (linhas de produção, agendamento de processos em sistemas computacionais, etc.).

Neste trabalho final, os grupos terão o desafio de resolver o **Problema de Agendamento de Tarefas em Linha (Flow Shop Scheduling Problem - FSSP)**, utilizando um ou mais algoritmos bioinspirados (combinados ou não) estudados na disciplina.

O FSSP consiste em um conjunto de N tarefas que devem ser processadas em um conjunto de M máquinas. Todas as tarefas devem passar por todas as M máquinas na mesma ordem. Por exemplo, uma tarefa j deve ser processada primeiro na máquina 1, depois na máquina 2, e assim por diante, até a máquina M . O objetivo é determinar a ordem das tarefas para todas as máquinas que minimize o *makespan*, ou seja, o tempo total para que todas as tarefas sejam concluídas.

Este trabalho é de implementação e os grupos competirão entre si para encontrar as melhores soluções (menores *makespans*) para um conjunto de instâncias fornecidas.

2 Objetivos

- Implementar um ou mais algoritmos bioinspirados (Algoritmo Genético, Colônia de Formigas, Seleção Clonal, PSO), combinados ou não, para resolver o FSSP.

- Otimizar os parâmetros do(s) algoritmo(s) escolhido(s) para encontrar as melhores soluções possíveis para as instâncias fornecidas.
- Competir com outros grupos para obter o menor *makespan* em cada instância.
- Analisar e discutir o desempenho do algoritmo, suas vantagens e desvantagens, e as estratégias de otimização de parâmetros.
- apresentar os detalhes de implementação e principais decisões de projeto, a fim de fomentar uma discussão entre os grupos.

3 O Problema (FSSP)

Formalmente, o FSSP pode ser definido como:

- N tarefas: $J = \{J_1, J_2, \dots, J_N\}$
- M máquinas: $K = \{M_1, M_2, \dots, M_M\}$
- Cada tarefa J_j consiste em M operações, onde a i -ésima operação de J_j deve ser processada na máquina M_i e tem um tempo de processamento p_{ji} .
- Todas as tarefas seguem a mesma sequência de máquinas: $M_1 \rightarrow M_2 \rightarrow \dots \rightarrow M_M$.
- Uma máquina pode processar apenas uma tarefa por vez.
- Uma tarefa pode ser processada em apenas uma máquina por vez.
- Uma vez que uma operação começa, ela não pode ser interrompida.

O objetivo é encontrar uma permutação $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_N)$ das tarefas que minimize o *makespan* (C_{\max}).

Para calcular o *makespan* para uma dada sequência de tarefas σ :
Seja $C(J_j, M_i)$ o tempo de conclusão da tarefa J_j na máquina M_i . As condições de tempo de conclusão são:

1. $C(J_{\sigma_1}, M_1) = p_{\sigma_1,1}$ (Primeira tarefa na primeira máquina)

2. $C(J_{\sigma_j}, M_1) = C(J_{\sigma_{j-1}}, M_1) + p_{\sigma_j,1}$ para $j > 1$ (Conclusão da tarefa anterior na mesma máquina + tempo de processamento da tarefa atual)
3. $C(J_{\sigma_1}, M_i) = C(J_{\sigma_1}, M_{i-1}) + p_{\sigma_1,i}$ para $i > 1$ (Conclusão da tarefa atual na máquina anterior + tempo de processamento da tarefa atual)
4. $C(J_{\sigma_j}, M_i) = \max(C(J_{\sigma_{j-1}}, M_i), C(J_{\sigma_j}, M_{i-1})) + p_{\sigma_j,i}$ para $j > 1, i > 1$ (Máximo entre a conclusão da tarefa anterior na mesma máquina e a conclusão da tarefa atual na máquina anterior + tempo de processamento da tarefa atual)

O *makespan* é $C_{\max} = C(J_{\sigma_N}, M_M)$.

4 Implementação

Cada grupo deverá implementar:

- **Função de avaliação:** Uma função que recebe uma permutação de tarefas e os tempos de processamento das máquinas, e retorna o *makespan* correspondente.
- **Algoritmo Bioinspirado:** Implementar um dos algoritmos estudados (AG, ACO, CLONALG, PSO) ou uma combinação/variação deles, adaptado para resolver o FSSP.
- **Otimização de Parâmetros:** Ajustar os parâmetros do algoritmo (ex. tamanho da população, número de gerações/iterações, taxas de mutação/cruzamento, parâmetros de feromônio, etc.) para obter o melhor desempenho.

5 Instâncias do Problema

Serão fornecidos arquivos de instância. Cada arquivo representa um problema FSSP diferente. O formato de cada arquivo será:

```
N M
p_11 p_12 ... p_1M
p_21 p_22 ... p_2M
...
p_N1 p_N2 ... p_NM
```

Onde:

- N é o número de tarefas.
- M é o número de máquinas.
- p_{ji} é o tempo de processamento da tarefa j na máquina i .

Exemplo de Arquivo de Instância (fssp_instance_01.txt):

```
3 3
10 4 8
5 6 7
3 2 12
```

Isso significa: 3 tarefas, 3 máquinas.

- Tarefa 1: 10 na M1, 4 na M2, 8 na M3
- Tarefa 2: 5 na M1, 6 na M2, 7 na M3
- Tarefa 3: 3 na M1, 2 na M2, 12 na M3

Instâncias para o FSSP

As instâncias serão fornecidas nos seguintes arquivos:

- fssp_instance_01.txt: 3 tarefas, 3 máquinas
- fssp_instance_02.txt: 5 tarefas, 4 máquinas
- fssp_instance_03.txt: 7 tarefas, 5 máquinas
- fssp_instance_04.txt: 10 tarefas, 6 máquinas

6 Submissão e Competição

Cada grupo deverá submeter:

1. **Código-Fonte:** Todo o código-fonte da solução, com comentários claros.

2. **Relatório (PDF):** Um documento descrevendo a abordagem, o algoritmo escolhido, a justificativa para as decisões detalhadas de projeto (representação, operadores, etc.), os parâmetros utilizados, e uma análise detalhada dos resultados obtidos para cada instância. Inclua gráficos de convergência.
3. **Arquivo de Resultados:** Um arquivo de texto para cada instância no formato `resultado_instancia_XX.txt` contendo:

Melhor Makespan: [valor do makespan]

Melhor Sequencia: [sequencia de tarefas separadas por espaço]

Tempo de Execucao (segundos): [tempo total de execucao]

Regras de Competição:

- A avaliação será feita usando um conjunto de instâncias fornecidas (as mesmas para todos os grupos) e outras instâncias que todos os grupos desconhecem.
- Será imposto um **limite de tempo de execução** para cada instância (e.g., 5 minutos por instância). Soluções que excederem o tempo limite serão penalizadas.
- A métrica principal de competição será o **menor *makespan* obtido**. Em caso de empate no *makespan*, o menor tempo de execução será o critério de desempate.
- O uso de bibliotecas de otimização prontas ou de implementações de algoritmos bioinspirados diretamente de bibliotecas é **PROIBIDO**. O objetivo é que vocês implementem os algoritmos do zero.
- É permitido o uso de bibliotecas numéricas como NumPy e de visualização de dados.

7 Critérios de Avaliação

- **Corretude da Implementação: (20%)** O algoritmo funciona como esperado, calcula o *makespan* e implementa as operações do algoritmo corretamente. O código deve ser legível e comentado.

- **Qualidade da Solução: (20% será proporcional ao rank do grupo na solução)** Quão perto das melhores soluções conhecidas (ou do ótimo) o algoritmo do grupo chega.
- **Eficiência Computacional: (10% será proporcional ao rank do grupo na solução)** Tempo de execução para encontrar a solução.
- **Análise e Discussão: (20%)** Qualidade do relatório, clareza das explicações, profundidade da análise dos resultados e da otimização dos parâmetros.
- **Originalidade/Criatividade (10 %):** Se o grupo implementar uma abordagem inovadora ou uma combinação interessante de algoritmos.
- **Apresentação: (10%)** Qualidade da apresentação, adequação ao tempo, clareza e profundidade das explicações.
- **Qualidade da Apresentação comparativa: (10% será proporcional ao rank do grupo na apresentação oral e do documento)** Quão boa e criativa são as apresentações dos resultados no texto e oral.

8 Considerações Finais

Os documentos devem ser entregues até o dia 07/07 e serão apresentados nos dias 07 e 09 de julho.

A apresentação não deve ser menor do que 10 minutos e nem maior do que 20 minutos, sob pena de redução da nota nesse quesito.

O membro do grupo que não comparecer na apresentação não terá sua participação no grupo considerada para efeito algum.

Todos os membros do grupo devem ter domínio sobre todos os detalhes do trabalho, será aplicado um questionário sobre o trabalho ao final das apresentações.