

Trabalho Prático
Estruturas de Dados Avançadas (EDA)
ESI/ES IPL/EIM/EEC/EDJD
EST-IPCA

Fase II – Auxiliar de Interpretação

Process Plan	Operation						
	0 1	0 2	0 3	0 4	0 5	0 6	0 7
pr _{1,2}	(1,3) [4,5]	(2,4) [4,5]	(3,5) [5,6]	(4,5,6,7,8) [5,5,4,5,9]			
pr _{2,2}	(1,3,5) [1,5,7]	(4,8) [5,4]	(4,6) [1,6]	(4,7,8) [4,4,7]	(4,6) [1,2]	(1,6,8) [5,6,4]	(4) [4]
pr _{3,3}	(2,3,8) [7,6,8]	(4,8) [7,7]	(3,5,7) [7,8,7]	(4,6) [7,8]	(1,2) [1,4]		
pr _{4,2}	(1,3,5) [4,3,7]	(2,8) [4,4]	(3,4,6,7) [4,5,6,7]	(5,6,8) [3,5,5]			
pr _{5,1}	(1) [3]	(2,4) [4,5]	(3,8) [4,4]	(5,6,8) [3,3,3]	(4,6) [5,4]		
pr _{6,3}	(1,2,3) [3,5,6]	(4,5) [7,8]	(3,6) [9,8]				
pr _{7,2}	(3,5,6) [4,5,4]	(4,7,8) [4,6,4]	(1,3,4,5) [3,3,4,5]	(4,6,8) [4,6,5]	(1,3) [3,3]		
pr _{8,1}	(1,2,6) [3,4,4]	(4,5,8) [6,5,4]	(3,7) [4,5]	(4,6) [4,6]	(7,8) [1,2]		

Tabela 1. Process Plan para um problema de escalonamento com dimensão 8x7 (8 máquinas)

Fase 1

1. Definição de uma estrutura de dados dinâmica para a representação de um *job* com um conjunto finita de n operações;
2. Armazenamento/leitura em ficheiro;
3. Inserção de uma nova operação num *job*;
4. Remoção de uma determinada operação de um *job*;

5. Edição das operações associadas a um *job*;
6. Determinação da quantidade mínima de unidades de tempo necessárias para completar o *job* e listagem das respectivas operações;
7. Determinação da quantidade máxima de unidades de tempo necessárias para completar o *job* e listagem das respectivas operações;
8. Determinação da média de unidades de tempo necessárias para completar uma operação, considerando todas as alternativas possíveis;

Fase 2

1. Definição de uma estrutura de dados dinâmica para representação de um conjunto finito de m *jobs* associando a cada *job* um determinado conjunto finito de operações;
2. Armazenamento/leitura do ficheiro de um *process plan* (representar para efeito de teste o *process plan* da Tabela 1);
3. Inserção de um novo *job*;
4. Remoção de um *job*;
5. Inserção de uma nova operação num *job*;
6. Remoção de uma determinada operação de um *job*;
7. Edição das operações associadas a um *job*;
8. Geração de uma solução para o FJSSP, apresentando a distribuição das operações pelas várias máquinas, e determinando o menor *makespan* (unidades de tempo necessárias para realização de todos os *jobs*);

Gestão de Jobs:

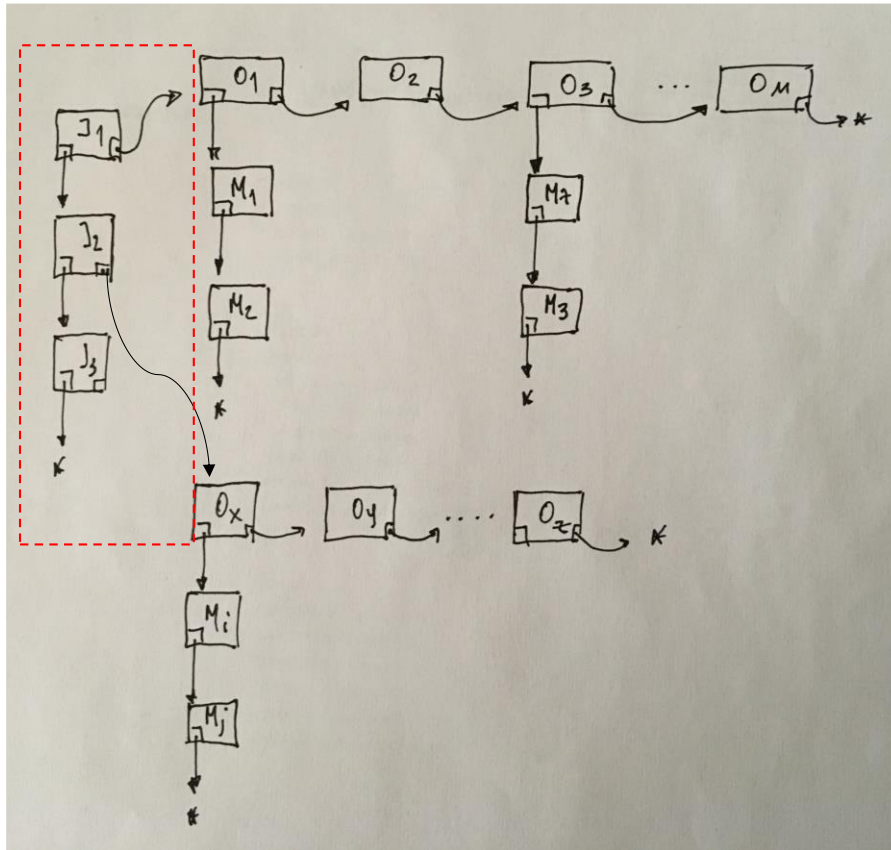
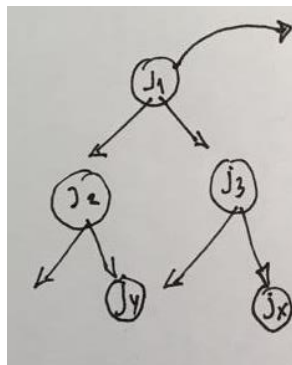


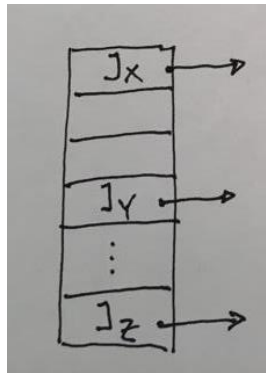
Figura 1 - Gestão de Jobs com Lista Genérica

A gestão do conjunto de Jobs poderá ser suportada por:

- Lista Genérica (esquema na Figura 1)
- Árvore Binária de Procura (ABP)



- Tabela de Hash



Planeamento (FJSSP):

		Tempo							
		t_1	t_2	t_3	t_4				t_T
Máquinas	m_1	O_1J_1	O_1J_1	O_1J_1					
	m_2								
	m_3	O_1J_2	O_1J_2		O_2J_1	O_2J_1	O_2J_1		
	m_4								
				O_2J_2	O_2J_2	O_2J_2	O_2J_2		
								O_3J_2	O_3J_2
	m_n							O_3J_1	O_3J_1

Proposta base de solução:

```

CelulaPlano Plano[M][T];
...
bool Ocupa(int job, int oper, int maq, int t);

```