

Trabalho: Regras

As regras do trabalho são disponíveis na [página do trabalho](#).

Trabalho: Problemas

WEIGHTED TARDINESS

Entrada Um conjunto de n tarefas com tempo de execução p_j , peso w_j e um prazo “ideal” (due date) d_j , $j \in [n]$.

Solução As tarefas tem que ser processadas numa máquina em alguma ordem. A máquina consegue executar no máximo uma tarefa cada vez, e cada tarefa tem que ser executada sem interrupção. Logo uma solução é uma permutação π das tarefas.

Objetivo Para uma dada permutação o tempo de acabamento da tarefa j é $C_j = \sum_{i|\pi_i \leq \pi_j} p_i$. O atraso da tarefa relativo ao prazo é definido por $T_j = \max\{C_j - d_j, 0\}$. O objetivo é minimizar o atraso ponderado total é $\sum_{j \in [n]} w_j T_j$.

Informações adicionais Instâncias disponíveis em <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/wtinfo.html>. Testes devem ser feitos as instâncias 3, 14, 15, 92 com $n = 40, 50, 100$ tarefas, para um total de 12 instâncias.

OPEN SHOP SCHEDULING

Entrada Um conjunto de tarefas $J = [n]$ e um conjunto de máquinas $M = [p]$. Cada tarefa consiste em p operações com tempo de execução t_{jm} para $j \in J, m \in M$. A m -ésima operação da i -ésima tarefa tem que ser executada na máquina m . Cada máquina consegue executar no máximo uma operação cada vez, e uma operação tem que ser executada sem interrupção.

Solução Um tempo de início i_{jm} de cada operação de cada tarefa, tal que as operações de cada tarefa são executadas em ordem, i.e., $i_{jm} + t_{jm} \leq i_{j,m+1}$ para todos $j \in J$ e $m \in [p-1]$, e tal que não tem sobreposição de tarefas na mesma máquina, i.e., para um dado $m \in M$ todos intervalos $(i_{jm}, i_{jm} + t_{jm})$ para $j \in J$ são mutualmente disjuntos.

Objetivo Minimizar o término da última operação executada (inglês: *makespan*), i.e. minimizar $\max_{j \in J} i_{jm} + t_{jm}$.

Informações adicionais Instâncias disponíveis em <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/openshopinfo.html>. Testes devem ser feitos com instâncias 3, 5 com $n = 4, 5, 7, 10, 15, 20$ tarefas, para um total de 12 instâncias.

CLIQUE MÁXIMO

Entrada Um grafo não-direcionado $G = (V, A)$.

Solução Uma clique $C \subseteq V$, i.e., um conjunto C de vertices tal que todo par de vértices em C é conectado por uma aresta.

Objetivo Maximizar $|C|$.

Informações adicionais Instâncias disponíveis em <http://cs.hbg.psu.edu/txn131/clique.html>. Testes devem ser feitos com os grafos brock800_1, brock800_2, p_hat700-1, p_hat700-2, p_hat1000-1, p_hat1000-2, p_hat1500-1, p_hat1500-2, MANN_a27, MANN_a45, MANN_a81, san400_0.9-1.