

Scheduling: Cotas y Redondeos

Dpto. Ingeniería Industrial, Universidad de Chile

IN47B, Ingeniería de Operaciones

Contenidos

- 1 Notación
- 2 Relajaciones

Desc. problemas Scheduling

Hacer $\{1, \dots, n\}$ trabajos en $\{1, \dots, m\}$ máquinas

Maquinas | r , d , p_{mtn} | func. objetivo

Graham, Lawler, Lenstra y Rinnooy Kan (1979)

Numero de maquinas

Letra Numero

- ① P : maquinas identicas paralelas (trabajo j se demora p_j en todo i)
- ② Q : maquinas uniformes (maquina i demora s_i , $p_{ij} = p_j / s_i$)
- ③ R : maquinas no relacionadas (s_{ij} , $p_{ij} = p_j / s_{ij}$)
- ④ O : open shop, cada trabajo j requiere de m procesos en cualquier orden
- ⑤ F : flow shop, cada trabajo j requiere de m procesos en el mismo orden
- ⑥ J : job shop, cada trabajo j requiere una serie de procesos con alguna precedencia.

Numero: indica cuantas maquinas, si no esta son m .

Propiedades del problema

Propiedades (r , d , $pmtn$)

- 1 r : release times de los trabajos. tiempo de comienzo
- 2 d : due date. tiempo en que termina cada trabajo.
- 3 $pmtn$: preemption (interrupcion en un trabajo para comenzar otro)

Funcion objetivo

- 1 $\sum_{j=1}^n C_j$
- 2 $\sum_{j=1}^n L_j = \sum_{j=1}^n \max\{C_j - d_j, 0\}$
- 3 $\sum_{j=1}^n U_j$ donde $U_j = 1$ cuando $C_j - d_j > 0$

$R||C_{\text{máx}}$ $R||C_{\text{máx}}$

Tenemos

- 1 $C_{\text{máx}}$ maximo completion time
- 2 R maquinas no relacionadas $p_{ij} = p_j/s_{ij}$
- 3 m maquinas
- 4 n trabajos.

Problema es NP-hard

- relajación
- redondeo para encontrar solución con garantia

$R||C_{\text{máx}}$ $R||C_{\text{máx}}$

Un problema relacionado facil es $R|_{\text{pmtn}} | C_{\text{máx}}$.
 Considere x_{ij} fracción de trabajo j que se hace en i .

$$\begin{array}{ll}
 \text{mín} & D \\
 \text{s.t.} & \sum_{i=1}^m x_{ij} = 1 \quad j \in \{1, \dots, n\} \\
 & \sum_{j=1}^n p_{ij} x_{ij} \leq D \quad i \in \{1, \dots, m\} \\
 & \sum_{i=1}^m p_{ij} x_{ij} \leq D \quad j \in \{1, \dots, n\} \\
 & x_{ij} \geq 0
 \end{array}$$

R|pmtn|C_{máx}

- 1 Solución del PL entrega un plan de trabajo.
- 2 Issue:
Si se puede programar x_{ij} sin repartir el mismo trabajo j a dos maquinas i al mismo tiempo.
- 3 Matching.

El problema sin preemption se puede escribir

$$\begin{array}{ll}
 \text{mín} & D \\
 \text{s.t.} & \sum_{i=1}^m x_{ij} = 1 \quad j \in \{1, \dots, n\} \\
 & \sum_{j=1}^n p_{ij} x_{ij} \leq D \quad i \in \{1, \dots, m\} \\
 & x_{ij} \in \{0, 1\}
 \end{array}$$

Relajación

Un problema de decision:

$$\begin{aligned} \sum_{i=1}^m x_{ij} &= 1 & j \in \{1, \dots, n\} \\ \sum_{j=1}^n p_{ij} x_{ij} &\leq D & i \in \{1, \dots, m\} \\ x_{ij} &\geq 0 \\ x_{ij} &= 0 & \text{si } p_{ij} \geq D \end{aligned}$$

Esto ya no es un problema lineal si D es variable.

Si existe plan con lifespan D entonces hay una solución a lo anterior

Si no existe solución entonces no existe plan con lifespan D

Relajación

- Si existe sol factible al problema de decision, encontremos un plan
- Usamos un redondeo y propiedades de PL.
- Solucion tiene a lo mas $m + n x_{ij} > 0$
- Solucion tiene a lo mas m trabajos j fracionados.
- En le grafo $(i, j) \in A$ ssi $0 < x_{ij} < 1$ se puede ordenar trabajos a maquinas (bosque con pocos ciclos).
- Redondeo tiene lifespan $\leq 2D$.

Problema

- 1 maquina
- con release times
- minimizar suma de tiempos de termino

- Problema NP-hard
- Pero $1|r|pmtn|\sum C$ es facil
 - SRPT encuentra el optimo
 - Sup que no, y cambiar trabajo entre el que tiene el shortest remaining processing time y el que es ejecutado primero

Relajación

- Resolver $1|r \text{ pmtn}|\sum C$ encontrar C^P
- ordenar trabajos segun $C_1^P \leq C_2^P \leq \dots \leq C_n^P$
- ejecutar trabajos sin preemption en este orden, C^N
- $\sum C^P \leq \sum C^N \leq 2 \sum C^P$