

Ricerca Operativa: AMPL



Scheduling dei processi

Stefano Cavalli

Anno accademico 2018/2019

Ricerca Operativa: AMPL

Il problema

Un insieme T di processi deve essere eseguito su un numero r di processori. Ogni processo deve eseguire prima un'operazione sul processore 1, poi sul processore 2, fino al processore r .

Per ogni processo i e ogni processore j è indicato un tempo p_{ij} di esecuzione dell'operazione relativa al processo i sul processore j .

Si vuole pianificare in quale istante far partire l'esecuzione di ogni operazione in modo tale che l'istante in cui tutti i processi risultano terminati sia minimizzato.

Si ipotizzi che le operazioni non siano interrompibili (ovvero, una volta partita la loro esecuzione non possono essere interrotte).

Si formuli il modello matematico per questo problema, lo si scriva in AMPL e si definiscano i dati di una particolare istanza, risolvendola. Provate a fare test con $r = 2, 3, 4$. Si faccia inoltre un'analisi di cosa succede se si modificano alcuni dei dati dell'istanza.

Ricerca Operativa: AMPL

Il modello matematico

Per risolvere questo problema è stato creato un modello che rispetta i seguenti parametri, insiemi, variabili e vincoli.

PARAMETRI:

```
param tot_processes >= 0 integer ; # number of processes must be >= 0
param r >= 0 integer ;           # number of processors must be >= 0
param Z >= 0 integer ;           # set to 40 but it could have been:
                                #max(sum(total execution time)),for every
                                #processor j.
```

INSIEMI:

```
set PROCESSES := 1..tot_processes ;
set PROCESSORS := 1..r ;
set TIME := 0..Z-1 ;
```

VARIABILI:

```
var end >= 0 ;    #end that must be minimized (every process has terminated)
var busy {PROCESSES, PROCESSORS, TIME} binary ; #processor j is busy
                                                    #(1) or not (0)
```

OBIETTIVO:

“Si vuole pianificare in quale istante far partire l’esecuzione di ogni operazione in modo tale che l’istante in cui tutti i processi risultano terminati sia minimizzato.” Questo è un problema di minimizzazione, di conseguenza la funzione obiettivo sarà

min execution_time: end;

VINCOLI:

#1 every process must terminate before the param Z

subject to vincolo_end {i in PROCESSES}:

$$\sum\{t \text{ in TIME}\} (t + p[i,r])*busy[i,r,t] \leq \text{end} ;$$

#2 each process assigned to a single processor for r times (r= processor's number)

subject to vincolo_process_to_processor {i in PROCESSES} :

$$\sum\{t \text{ in TIME}, j \text{ in PROCESSORS}\} busy[i,j,t] = r ;$$

#3 each activity must run just one time in t(0..Z-1)

subject to vincolo_activity_one_time {i in PROCESSES, j in PROCESSORS} :

$$\sum\{t \text{ in TIME}\} busy[i,j,t] = 1 ;$$

#4 each processor must run one process in t(0...Z-1)

subject to vincolo_processor_one_process {j in PROCESSORS, t in TIME} :

$$\sum\{i \text{ in PROCESSES}, k \text{ in max}(t-p[i,j]+1,0)..(t)\} busy[i,j,k] \leq 1 ;$$

#5 each process must have just one activity running at the same time

subject to vincolo_limite2 {i in PROCESSES, t in TIME} :

$$\sum\{j \text{ in PROCESSORS}, k \text{ in max}(t-p[i,j]+1,0)..(t)\} busy[i,j,k] \leq 1 ;$$

#6 required some priority between activities

subject to vincolo_priority_activity {i in PROCESSES, j in PROCESSORS, t in TIME} :

$$\sum\{y \text{ in } 0..t\} busy[i,\max(j-1,1),y] \geq busy[i,j,t] ;$$

Ricerca Operativa: AMPL

Implementazione del modello matematico

L'implementazione del modello è stata inserita nel file `process_scheduling.MOD`, in cui sono descritti matematicamente tutti i vincoli, le variabili, gli insiemi e la funzione obiettivo del problema che si sta cercando di risolvere.

Nel file `process_scheduling.DAT`, invece, vengono inseriti i dati del problema (gli input) che, a seconda della casistica, produrranno dei risultati differenti.

`process_scheduling.RUN` contiene invece le istruzioni per eseguire il modello e mostrare in output i risultati ottenuti.

Ricerca Operativa: AMPL

Le istanze del problema

Le tre casistiche evidenziate dal problema sono:

- 1 caso: analisi con 2 processori;
- 2 caso: analisi con 3 processori;
- 3 caso: analisi con 4 processori;

Ho scelto un numero di processi sempre pari a 8 (valore oltre la quale l'esecuzione del programma richiede un numero elevato di secondi) insieme ad un parametro temporale massimo entro la quale tutti i processi devono essere terminati e che mi servirà per la creazione dei vincoli ($=40$. Volendo avrei potuto utilizzare il valore massimo tra le somme di tutti i tempi di esecuzione di ogni sottoprocesso, per ogni diverso processore).

ANALISI CON 2 PROCESSORI:

$r = 2$, $\text{tot_processes} = 8$

	Processore 1	Processore 2	Processore 3	Processore 4
Processo 1	1	4	-	-
Processo 2	2	2	-	-
Processo 3	3	3	-	-
Processo 4	4	1	-	-
Processo 5	3	2	-	-
Processo 6	4	2	-	-
Processo 7	2	3	-	-
Processo 8	1	2	-	-

Il tempo necessario a svolgere ogni attività di ogni processo in modo da minimizzare la deadline è di 21 secondi.

ANALISI CON 3 PROCESSORI:

$r = 3$, $\text{tot_processes} = 8$

	Processore 1	Processore 2	Processore 3	Processore 4
Processo 1	1	4	2	-
Processo 2	2	2	1	-
Processo 3	3	3	2	-
Processo 4	4	1	3	-
Processo 5	3	2	3	-
Processo 6	4	2	4	-
Processo 7	2	3	3	-
Processo 8	1	2	4	-

Il tempo necessario a svolgere ogni attività di ogni processo in modo da minimizzare la deadline è di 26 secondi.

ANALISI CON 4 PROCESSORI:

$r = 4$, $\text{tot_processes} = 8$

	Processore 1	Processore 2	Processore 3	Processore 4
Processo 1	1	4	2	2
Processo 2	2	2	1	3
Processo 3	3	3	2	1
Processo 4	4	1	3	3
Processo 5	3	2	3	1
Processo 6	4	2	4	2
Processo 7	2	3	3	1
Processo 8	1	2	4	3

Il tempo necessario a svolgere ogni attività di ogni processo in modo da minimizzare la deadline è di 27 secondi.

Ricerca Operativa: AMPL

Analisi della sensitività

Cambiando i valori che rappresentano la durata di un sottoprocesso, non sempre viene modificato il valore ottimo. Se si scelgono dei valori interi molto più grandi, è ovvio che il valore ottimo sarà maggiore perché cambierà il tempo di esecuzione, che sarà più grande.

Tuttavia, ci sono anche alcuni in casi in cui un incremento del tempo di esecuzione di un sottoprocesso non implica per forza un incremento del valore obiettivo, per esempio nella situazione in cui ci fosse un processore IDLE, verrebbe utilizzato anziché rimanere inattivo, quindi il tempo finale non aumenterebbe.