Schedulazione Multi-progetto con Risorse Limitate

Dott.ssa Maria Silvia Pini

Dipartimento di Matematica Pura e Applicata
Email: mpini@math.unipd.it
Resp. accademico: Prof.ssa Francesca Rossi

1 Sommario

La schedulazione multi-progetto in presenza di risorse limitate è un problema che si verifica in vari contesti reali. Vedremo ora una formulazione matematica del problema nell'ambito del job shop scheduling. Questa formulazione permette di modellare varie situazioni del mondo reale come vincoli su più risorse, date di scadenze, risorse che possono essere sostituite, concorrenza e non concorrenza dell'esecuzione di alcuni job. In questo documento considereremo come funzione obiettivo la minimizzazione del tempo totale di throughput (che è il tempo totale trascorso dai progetti da quando arrivano a quando sono completati) e indicheremo anche alcuni possibili risolutori in grado di trovare la soluzione ottima del nostro problema.

2 Introduzione

I problemi di schedulazione che consideremo sono costituiti da un insieme di progetti ciascuno dei quali è costituito da un insieme di job. In questi problemi dobbiamo determinare l'istante temporale in cui un job deve essere eseguito, dato un certo limite sulla disponibilità delle risorse (cioè uomini o attrezzature). La determinazione di questo istante temporale dipende dall'obiettivo che ci prefiggiamo. In questo documento considereremo come obiettivo la minimizzazione del tempo totale di di throughput, che è il tempo totale trascorso dai progetti da quando arrivano a quando sono completati.

Per il problema della schedulazione è possibile fornire varie formulazioni alternative. Comunque una formulazione efficiente del problema dipenderà da una scelta mirata delle variabili del problema, dalla loro sintesi nella funzione obiettivo e nei vincoli. Varie formulazioni sono state date in letteratura per questo problema [2,4,1], comunque nessuna di loro considera vincoli sulle risorse. Proponiamo ora quella presentata in [3] che propone una formulazione di programmazione lineare intera 0-1 che considera sia i vincoli sulle risorse che la schedulazione di più progetti. Mostreremo ora un insieme di equazioni che sono necessarie per assicurare che la nostra schedulazione soddisfi i seguenti vincoli:

- risorse limitate;
- relazioni di precedenza tra job;
- possibilità di suddividere job;
- date di scadenza per job e per progetti;
- sostituzione di risorse per svolgere certi job;
- vincoli di concorrenza e non concorrenza sui job.

Per trovare la soluzione ottima del nostro problema è possibile usare sia risolutori di programmazione lineare intera che risolutori di vincoli. Indicheremo alcuni di questi risolutori nella parte finale di questo documento.

3 Il modello matematico

Per formulare il nostro problema di schedulazione su multi-progetti con risorse vincolate considereremo i seguenti indici:

- i = numero del progetto, $i = 1, \ldots, I$, dove I è il n. di progetti.
- j = numero del job, $j = 1, \ldots, N_i$, dove N_i è il n. di job nel progetto i.
- t = periodo di tempo, $t = 1, \ldots, maxG_i$, dove G_i è la scadenza assoluta. Il progetto i deve essere completato nel periodo G_i o prima. Se una scadenza assoluta non viene specificata, G_i diventa l'ultimo periodo nell'orizzonte della schedulazione.
- g_i = data di scadenza desiderata. Il progetto i non è in ritardo se è completato nel periodo g_i o prima di tale periodo.
- e_i = il primo periodo periodo possibile in cui il progetto i può essere completato.
- a_{ij} = il periodo di arrivo del job j appartenente al progetto i.
- d_{ij} = numero di periodi richiesti per completare il job j nel progetto i.
- l_{ij} = il periodo possibile in cui il job j può essere completato.
- u_{ij} = l'ultimo periodo possibile in cui il job j può essere completato rispetto a una data di scadenza assoluta.

- k = tipo di risorsa, $k = 1, \dots, K$, dove K è il numero dei diversi tipi di risorse.
- r_{ijk} = quantità della risorsa di tipo k richiesta dal job j nel progetto i.
- R_{kt} = quantità di risorsa di tipo k disponibile al tempo t.
- x_{ijt} = una variabile che vale 1 se il job j del progetto i è completato nel periodo t e 0 altrimenti. x_{ijt} è uguale a 0 per $t < l_{ij}$ o per $t > u_{ij}$.
- x_{it} = una variabile che vale 1 nel periodo t se tutti i job del progetto i sono stati completati nel periodo t (cioè completato al periodo t-1 o prima). x_{it} è uguale a 0 per $t < e_i$ and $t > G_i$.

3.1 Funzione obiettivo

I job devono essere schedulati in modo da ottimizzare una certa misura della performance o una funzione obiettivo e in modo da rispettare alcuni vincoli. La scelta della funzione obiettivo può essere diversa a seconda dei diversi ambienti di schedulazione. La funzione obiettivo che consideremo in questo documento riguarda la minimizzazione del tempo totale di throughput.

Il tempo di throughput di un progetto singolo è definita come il tempo trascorso tra l'arrivo del progetto e il suo completamento. Un progetto si intende completato quando tutti i suoi job sono stati completati. Minimizzare il tempo di throughput di un progetto singolo è equivalente a massimizzare il numero di periodi che restano dopo che il progetto è stato completato, dove questo numero è $\sum_{t=e_i}^{G_i} x_{it}$. Pertanto la funzione obiettivo per minimizzare la somma dei tempi di throughput per tutti i progetti può essere formulata nel modo seguente:

$$maximize\ z = \sum_{i=1}^{I} \sum_{t=e_i}^{G_i} x_{it}.$$

3.2 Completamento di un job

Ogni job ha esattamente un periodo di completamento.

$$\sum_{t=l_{ij}}^{u_{ij}} x_{ijt} = 1 \quad (i = 1, \dots, I; j = 1, N_i)$$

3.3 Completamento di un progetto

Il progetto i non può essere completato nel periodo t finchè tutti i suoi job non sono completati entro tale periodo, cioè finchè $\sum_{q=l_{ij}}^{t-1} x_{ijq} = 1$ per tutti gli N_i job del progetto i. Questa richiesta può essere scritta come

$$x_{it} \leq \frac{1}{N_i} \sum_{j=1}^{N_i} \sum_{q=l_{ij}}^{t-1} x_{ijq} \quad (i = 1, \ldots, I; t = e_i, e_i + 1, \ldots G_i).$$

3.4 Vincoli di sequenza

Un vincolo di sequenza viene richiesto quando un job non può essere iniziato finchè uno o più di altri job sono stati completati. Per esempio, assumiamo che su un progetto i il job m deve precedere il job n. Se t_{im} e t_{in} denotano i periodi di completamento dei job m e n rispettivamente, allora

$$t_{im} + d_{in} \leq t_{in}.$$

Dal momento che $t_{im} = \sum_{t=l_{im}}^{u_i m} t x_{imt}$ and $t_{in} = \sum_{t=l_{in}}^{u_i n} t x_{int}$, allora il vincolo di sequenza è

$$\sum_{t=l_{im}}^{u_i m} t x_{imt} + d_{in} \leq \sum_{t=l_{in}}^{u_i n} t x_{int}$$

3.5 Vincoli sulle risorse

Il valore r_{ijk} specifica il numero di unità di risorse di tipo k richieste per svolgere il job j nel progetto i. Quindi se $r_{ij1} = 3$ e $r_{ij2} = 2$, allora 3 unità di tipo 1 sono usate insieme a 2 unità di tipo 2 durante quesi periodi in cui viene eseguito il job j.

Le risorse richieste su un job si assume che siamo utilizzate finchè il job finisce. Se questa assunzione non fosse appropriata si può considerare una leggera modifica. Per esempio, se una certa risorsa viene utilizzata solo durante i primi p periodi del job dove $p < d_{ij}$, allora è possibile trattare il job come due 'subjob' consecutivi che differiscono per le richieste di risorse e con durata rispettivamente p e $d_{ij} - p$. Se i due subjob devono essere fatti in maniera consecutiva allora basta rimpiazzare $\leq$ with $=$ nell'ultima equazione sopra che diventa: $\sum_{t=l_{im}}^{u_i m} t x_{imt} + d_{in} = \sum_{t=l_{in}}^{u_i n} t x_{int}$. Notare che questo approccio si può applicare ad ogni divisione di un job in due o più subjob.

In ogni periodo la quantità di risorsa k usata su tutti i job non può superare la quantità di risorsa k disponbile. Un job sarà eseguito al tempo t se il job è completato nel periodo q dove $t \leq q \leq t + d_{ij} - 1$. Pertanto il vincolo di risorsa può essere scritto nel modo seguente:

$$\sum_{i=1}^{I} \sum_{j=1}^{N_i} \sum_{q=t}^{t+d_{ij}-1} r_{ijk} x_{ijq} \leq R_{kt},$$

dove $t = min a_{ij}, \dots, max G_i$ e $k = 1, \dots K$. Notare che $x_{ijt} = 0$ per $t < l_{ij}$ e per $t > u_{ij}$.

3.6 Sostituzione di risorse

In alcuni scenari reali può essere possibile utilizzare risorse alternative per svolgere certi job. Per esempio un uomo con abilità superiori può essere sostituito con un uomo con capacità inferiori in alcuni job.

Se la sostituzione di risorse è permessa sul job j del progetto i, il vincolo sulle risorse indicato nella sezione precedente deve essere modificato per tenere in considerazione sia la possibilità di sostituire le risorse che le potenziali differenze nelle durate dei job quando queste sono effettuate da risorse diverse. Per gestire questa condizione definiamo un insieme di lavori mutuamente esclusivi tali che solo uno di loro possa essere eseguito. Per esempio, se esistono due metodi alternativi per svolgere il job j', definiamo due alternative come job j_1 e j_2 con durate rispettivamente d_{ij_1} e d_{ij_2} e con $u_{ij_1} = u_{ij_2} = d_{ij'}$. In questo scenario il vincolo di completamento del job diventa:

$$\sum_{q=min\{l_{ij1}, l_{ij2}\}}^{u_{ij'}} (x_{ij_1q} + x_{ij_2q}) = 1.$$

Il vincolo di completamento del progetto diventa:

$$x_{it} \leq \frac{1}{N_i} [\sum_{j=1, j \neq j'}^{N_i} \sum_{q=l_{ij}}^{t-1} x_{ijq} + \sum_{q=min\{l_{ij_1}, l_{ij_2}\}}^{t-1} (x_{ij_1q} + x_{ij_2q})].$$

3.7 Concorrenza e non concorrenza di job

Un vincolo di concorrenza sui job m e n stabilsce che i due job devono essere eseguiti simultaneamente. Questa richiesta può essere formalizzata richiedendo

$$x_{imt} = x_{int}.$$

Un vincolo di non concorrenza sui job m e n stabilisce che i due job non devono essere eseguiti simultaneamente, ma permette che vengano eseguiti in qualche ordine. Il job m sarà eseguito nel periodo t se e solo se

$$\sum_{q=t}^{t+d_{im}-1} x_{imq} = 1$$

Il job n sarà eseguito nel periodo t se e solo se

$$\sum_{q=t}^{t+d_{in}-1} x_{inq} = 1.$$

Pertanto il vincolo desiderato è il seguente:

$$\sum_{q=t}^{t+d_{im}-1} x_{imq} + \sum_{q=t}^{t+d_{in}-1} x_{inq} \leq 1 \quad (t = max\{l_{im}, l_{in}, \ldots min\{u_{im}, u_{in}\}).$$

3.8 Suddivisione di job

Dal punto di vista teorico la suddivisione di job può essere gestita trattando ogni job come d_{ij} subjob ciascuno dei quali dura un periodo e imponenendo appropriati vincoli di sequenza su questi subjob. Comunque, dal punto di vista pratico, la capacità di suddividere un job viene adottata raramente, poichè ci sono dei costi di setup e poichè in genere si preferisce mantenere la continuità del job. Quindi definire meno di d_{ij} subjob per un particolare job può essere utile per fornire una sufficiente flessibilità nella suddivisione senza richiedere troppi subjob.

Supponiamo che il job j possa essere suddiviso e che i suoi subjob siano posti in sequenza come richiesto dal vincolo di sequenza presentato nella Sezione 3.4. Quando due dei suoi job in sequenza, chiamiamoli m e n, non sono fatti in maniera contigua allora

$$\tau_{mn} = \sum_{t=l_{in}}^{u_{in}} t x_{int} - \sum_{t=l_{im}}^{u_{im}} t x_{imt} - d_{in}$$

rappresenta la durata dello split. Possiamo associare a τ_{mn} un costo di penalità c_n che dipende dalla durata dello split oppure possiamo associare un costo di penalità C_n allo split, che è indipendente dalla sua durata. In questo secondo caso $C_n\tau_n$ è il costo di penalità della suddivisione del job dove τ_n è una variabile Booleana $0-1$ definita nel modo seguente: $\tau_n = 1$ se c'è uno split (cioè $\tau_{mn} > 0$) e $\tau_n = 0$ altrimenti.

Nella nostra formulazione possiamo richiedere che $C_n\tau_n$ sia inferiore a un certo valore.

4 Alcuni risolutori

Nella sezione precedente abbiamo visto come modellare i problema di schedulazione multi-progetto utilizzando la programmazione lineare intera. Ora indicheremo alcuni risolutori che restituiscono la soluzione ottima di questi problemi.

Per risolvere il nostro problema è possibile utilizzare vari risolutori di programmazione lineare intera:

- Tra i risolutori a pagamento si segnala ILOG CPLEX.
- Tra i risolutori che possono essere scaricati gratuitamente da Internet, si segnala GLPK. La versione di GLPK per Windows si può scaricare al seguente URL:
  http://gnuwin32.sourceforge.net/packages/glpk.htm.
  Nel pacchetto di installazione di GLPK è documentato (nel file lang.pdf) anche MathProg, il linguaggio che serve per descrivere il modello matematico da dare in ingresso a GLPK. Tutto ciò che serve per scrivere il modello nel linguaggio MathProg è un editor di file di testo, come il Notepad di Windows.
- Altri risolutori gratuiti di modelli di programmazione lineare intera sono reperibili sul sito di COIN-OR http://www.coin-or.org/index.html.

Per risolvere il problema di scheduling esaminato è anche possibile utilizzare dei risolutori di vincoli:

- Tra i risolutori a pagamento si segnala ILOG SOLVER.
- Tra i risolutori che possono essere scaricati gratuitamente da Internet, si segnalano GECODE (http://www.gecode.org/) e CHOCO (http://www.emn.fr/z-info/choco-solver/).

5 Conclusioni

In questo documento abbiamo considerato problemi di scheduling multi-progetto in presenza di risorse limitate in cui l'obiettivo è quello di minimizzare il tempo totale di throughput. Abbiamo presentato un modello matematico capace di modellare fedelmente questi problemi e abbiamo indicato alcuni risolutori in grado di trovare una soluzione ottima di questi problemi.

Riferimenti bibliografici

1. E. H. Bowman. The schedule-sequencing problem. *Operations Research*, 7(5):621–624, 1959.
2. A. A. B. Pritsker and L. J. Watters. On the job shop scheduling problem. *Operations Research*, 8(2):219–223, 1960.
3. A. A. B. Pritsker, L. J. Watters, and P. M. Wolfe. Multiproject scheduling with limited resources: A zero-one programming approach. *Managment Science*, 16(1):93–108, 1969.
4. H. Wagner. An integer linear programming model for machine scheduling. *Naval Research Logistics Quarterly*, 6(2):131–140, 1959.