Michele Lanzetta

Lo scheduling **per ottimizzare le risorse**

LO SCHEDULING È UN PROBLEMA COMPLESSO CHE SI RISCONTRA IN NUMEROSE APPLICAZIONI REALI E IN PARTICOLAR MODO IN QUELLE INDUSTRIALI. DATA LA SUA IMPORTANZA E COMPLESSITÀ QUESTO PROBLEMA È STATO ED È TUTT'OGGI OGGETTO DI STUDIO E DI RICERCA.

Nella maggior parte dei casi si tratta di allocare delle risorse a delle attività con lo scopo finale dell'ottimizzazione di una determinata funzione obiettivo. Data la massiccia concorrenza a livello industriale l'unica via di salvezza per le aziende moderne è quella di riuscire a presentare sul mercato prodotti efficienti innovativi e di buona qualità nel modo più rapido possibile. Nel corso degli anni diverse sono state le strategie di scheduling studiate e applicate nel settore industriale con risultati più o meno soddisfacenti.

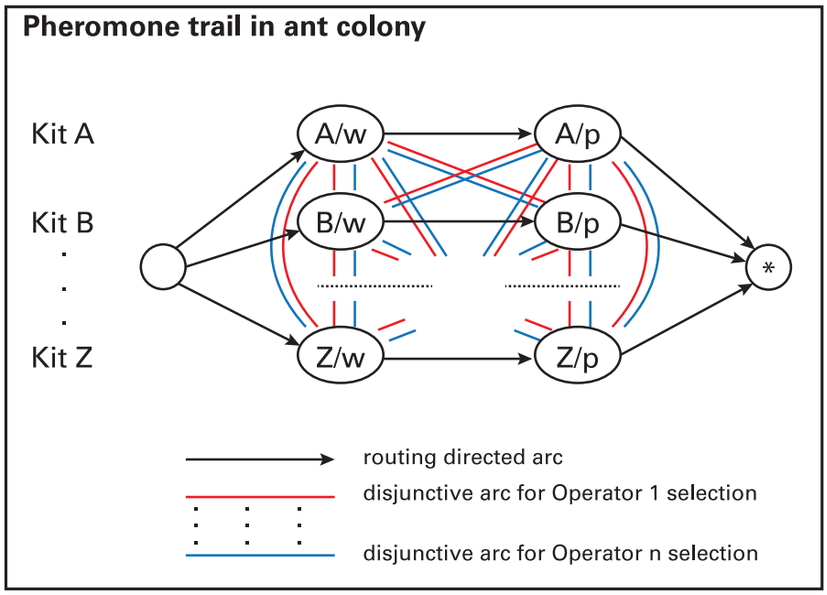
Le diverse tipologie

In generale i principali modelli di scheduling sono:

- Macchina singola: in questo caso tutti i job richiedono l'uso della stessa macchina per poter essere completati.
- Macchine parallele che si dividono ulteriormente in macchine parallele identiche, scorrelate e generiche: Nelle macchine parallele identiche i prodotti possono essere lavorati indifferentemente su m macchine identiche; nelle macchine generiche si hanno sempre m macchine ma ognuna di esse avrà un proprio tempo di lavorazione per cui il tempo finale per l'ottenimento del prodotto sarà diverso a seconda della macchina usata; nelle macchine scorrelate si ha una generalizzazione della situazione precedente: se le velocità delle macchine fossero indipendenti dal prodotto si ricadrebbe nel caso precedente.
- Flow shop: si hanno m macchine in serie e il prodotto deve visitarle tutte in un ordine prefissato (routing fisso).
- Flexible flow shop: è una generalizzazione della situazione precedente: anziché avere m macchine in serie si hanno un determinato numero di stadi di produzione contenenti ognuno un determinato numero di macchine identiche.
- Job shop: In questa organizzazione vi sono sempre m macchine ma ciascun prodotto le attraversa secondo un proprio ordine di lavorazione.
- Flexible job shop: vi sono c centri di lavoro, ognuno dei quali ha m macchine in parallelo. Ogni prodotto visita secondo un proprio ordine i vari centri di lavoro e dentro il centro può essere lavorato su qualsiasi macchina.
- Open shop: in questo caso ogni prodotto può essere processato senza nessun ordine sulle macchine.

Lo scheduling tipo flow shop con m macchine parallele per stage viene in genere identificato come flow shop ibrido (HFS). L'HFS è un'area industriale in cui un set di n jobs deve essere processato in una serie di m stages ottimizzando una data funzione obiettivo. Nella forma standard del problema HFS si hanno i seguenti vincoli:

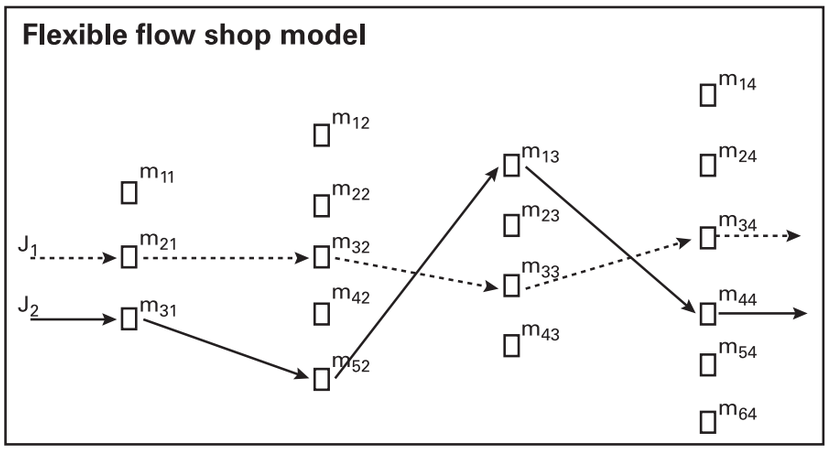
- tutti i jobs e le tutte le macchine sono disponibili al tempo zero.
- le macchine a un dato stage sono identiche e ogni macchina può processare una sola operazione in un dato tempo.
- ogni job può essere processato in una sola macchina in un dato tempo.



Grafo che mostra il flusso dei prodotti (linee) sulle macchine (nodi) ai fini della creazione del percorso ottimale (scheduling) e deposizione del feromone.

AI_2012_006_INT@027-029.indd 27 14/05/12 10.37

ALLOCARE E OTTIMIZZARE



Esempio di sequenziamento di lavorazioni di pezzi (i job J1 e J2) da lavorare su più macchine parallele (cioè funzionalmente equivalenti) suddivise in 4 stadi.

– il setup di un job non dipende da quello lavorato precedentemente le preemption non sono permesse.

Il problema HFS, cosi come la maggior parte dei problemi di scheduling, ha una complessità tale da essere considerato NP-hard (dove NP sta per non deterministic polynomial) poiché è un problema combinatorio, ossia il minimo tempo di completamento di un lotto si trova in corrispondenza di una ben determinata sequenza di job sulle macchine.

Convenzioni, parametri e funzioni obiettivo

La modifica, la rimozione o l'addizione delle assunzioni o dei vincoli del problema standard descritto sopra, porta all'ottenimento di diverse varianti del problema HFS. In generale i problemi sono classificati a seconda dei loro shop (α), a seconda delle assunzioni/vincoli (β) e a seconda della funzione obiettivo prescelta (γ). Quindi il problema è descritto dalla terna $\alpha\beta\gamma$. Il parametro α definisce la struttura dello shop, includendo il numero di stages, il numero e le caratteristiche delle macchine per stages. Il parametro β, indica la lista dei vincoli che devono essere considerati nell'affrontare il problema di ottimizzazione. I più comuni sono:

– rj: indica che il job j, non può essere processato prima della sua release date;
– prmu: indica che tutti i job vengono processati negli stage nello stesso ordine;
– prec: indica che ci sono dei vincoli di precedenza tra le operazioni di jobs differenti;
– Mj: indica che il job j può essere processato solo dalle macchine Mj allo stage k;

Per definire in modo esauriente il problema dello scheduling, devono essere definiti ulteriormente le seguenti quantità:

– Tempi di setup: indica il tempo necessario a riconfigurare la macchina da una lavorazione a un'altra.
– Preemption: se presente, indica la possibilità di interrompere l'esecuzione di un job a favore di un altro a priorità più alta.
– Precedence constraint: indica la presenza di vincoli di precedenza tra i vari job.
– Machine breakdown: indica la presenza di un guasto su una macchina che ne determina il suo inutilizzo.

Inoltre bisogna considerare:

– Tempo di completamento c_jk: indica il tempo necessario affinché la lavorazione sul job j sia portata a termine sulla macchina i.
– Lateness L_j: indica la differenza tra il tempo di completamento e il tempo di consegna del job j. Se tale differenza è positiva, si ha un ritardo e se invece è negativa indica un anticipo.
– Tardiness T_j: è definito come $\max(0;L_j)$;
– Earliness E_j: è definito cone $\min(0;L_j)$.
– In generale è comune associare al job j un peso w_j che può essere a seconda del caso il costo del job oppure il volume.

Tra le varie funzioni obiettivo da ottimizzare possiamo avere:

– Minimizzazione del makespan definito come il tempo di completamento dell'ultimo job in lavorazione che consente quindi in modo equivalente di massimizzare il coefficiente di utilizzo delle macchine.
– Minimizzazione della Lateness che significa individuare la massima violazione della data di consegna.
– Minimizzazione della Tardiness.
– Minimizzazione del tempo di setup complessivo.

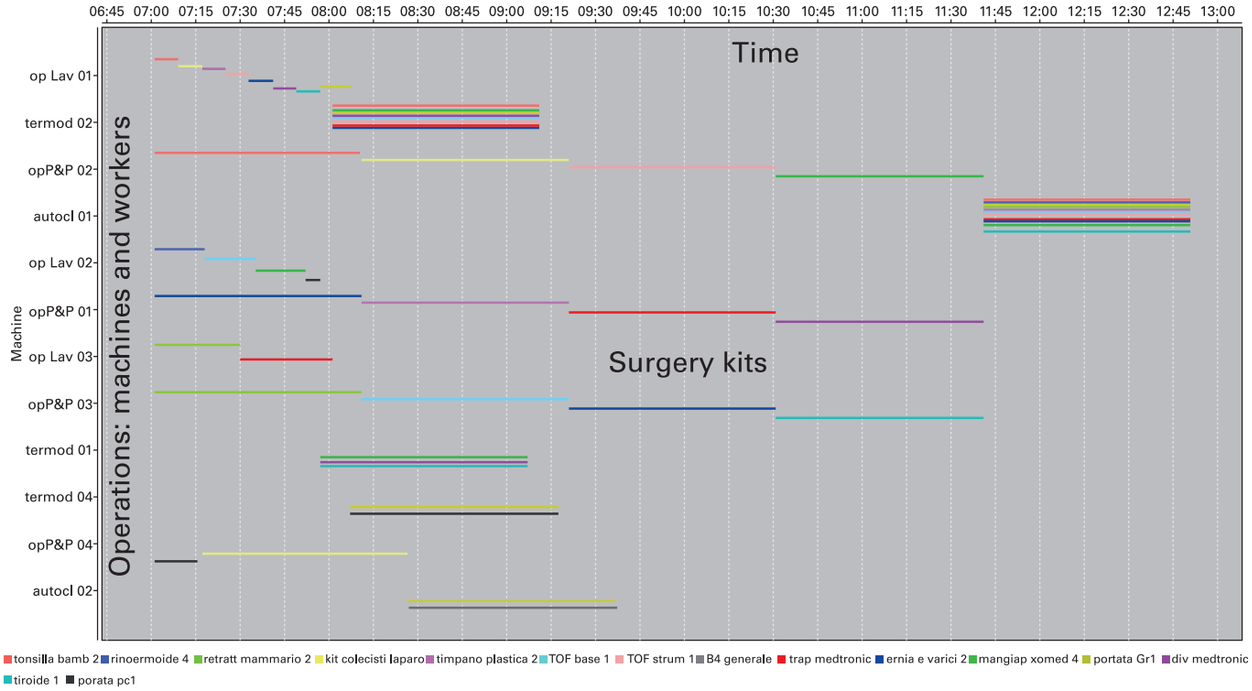
Stato dell'arte

Dato il grande numero di approcci possibili e data la grande varietà del problema si può fare una classificazione in tre grandi classi, cioè quella degli algoritmi esatti quella degli algoritmi euristici e quella degli algoritmi metaeuristici.

Exact algorithm

Senza dubbio la tecnica B&B (branch and bound) è preferita per la risoluzione del problema HFS. Nel caso più semplice, in cui si hanno 2 stage con una sola macchina al primo stage e due al secondo, la tecnica B&B fu proposta per la prima volta da Rao. Successivamente lo stesso problema fu affrontato da Bolat, Harkan e Harby che lo approcciarono con algoritmi genetici euristici e con la tecnica B&B. Il caso opposto fu affrontato da Arthanary, Ramaswamy Mittal e Bagga. Il problema con due stage e qualsiasi numero di macchine identiche al secondo stage fu recentemente studiato e risolto. Inoltre Lee e Kim, proposero un metodo B&B con la minimizzazione della tardiness totale. Invece, il caso in cui il primo stage ha un numero qualsiasi di macchine e il secondo stage ne ha uno solo, è stato studiato da Gupta Hariri e Potts i quali proposero un metodo B&B che è capace di ottenere buone soluzioni in un tempo ragionevole. Successivamente, è stato risolto il problema in cui si hanno due stage senza vincoli sul numero di macchina basato sulla minimizzazione del makespan. In questo caso, è stato implementato un metodo B&B che produce ottime soluzioni per problemi

AI_2012_006_INT@027-029.indd 28 14/05/12 15.22



Visualizzazione di un diagramma di gantt che mostra l'impegno temporale delle diverse macchine per le varie lavorazioni rappresentate da tratti di diverse lunghezze e colori.

in cui il numero di job può arrivare a 1000. La prima versione del metodo B&B per il problema generale di HFS, con qualsiasi numero di macchine e stage, è dovuto a Brah e Hunsucker. Inizialmente, questa tecnica, fu proposta per un problema in cui era presente un unico stage con macchine parallele e poi conseguentemente fu adattato al problema di cui sopra.
Nella maggior parte degli algoritmi B&B, la costruzione della soluzione parte nel primo stage, poi passa il secondo stage fino a attraversare tutti gli stage. Ad ogni punto di decisione, lo stage critico per esempio il bottleneck e un job vengono selezionati. Il problema, poi viene riavviato per decidere se c'è uno schedule con un valore di makespan piccolo rispetto a un certo limite detto Upper Bound o UB.

Heuristics

Il metodo euristico più semplice, è il dispatching rules conosciuto anche come list scheduling algorithms. Si tratta, di semplici regole per la classificazione e l'assegnamento dei jobs sulle macchine. Questi algoritmi, sono stati usati come strumento per la risoluzione di problemi di rilevanza sia teorica che pratica. Per esempio Salomon e Guinet, usarono tali metodi per la risoluzione di un problema HFS. Successivamente Kochar e Morris, proposero una alternativa del metodo e un algoritmo di local search rispettivamente. Gupta e Tunk, proposero inoltre una serie di regole basate sempre su metodi euristici per la soluzione di un problema a due stadi. Invece, Hunksucker e Shah studiarono un problema vincolato di HFS con la presenza di un numero vincolato di jobs. Il list scheduling, quindi, è particolarmente adatto a far fronte a ambienti industriali complessi e dinamici. Un dato numero di metodi euristici sofisticati, usa la strategia del dividi e conquista (divide and conquer) in cui il problema originale è diviso in tanti piccoli sottosistemi che vengono poi risolti uno alla volta e le loro soluzioni sono integrate in una unica soluzione per il problema originale.

Metaheuristics

Negli ultimi venti anni, i ricercatori che lavorano nel campo dell'ottimizzazione, hanno sviluppato tutta una serie di strategie che vanno a migliorare i metodi euristici visti in precedenza. Molti di questi metodi, conosciuti come metaeuristici, aggiungono un elemento di randomizzazione al metodo euristico con l'idea che il suo ripetuto uso possa condurre a una soluzione migliore. Nel simulted annealing (SA) per esempio una local search euristica viene chiamata molte volte. L'elemento randomness, è integrato con l'idea di prevenire la stagnazione su un ottimo, il quale rappresenta a volte il punto debole di una procedura di local search. La Tabù search (TS), ottiene lo stesso obiettivo permettendo movimenti non migliorativi e evitando certi movimenti migliorativi che possono portare la search a punti già visitati. Gli algoritmi genetici (GA), invece mantengono una popolazione di soluzioni e conducono una esplorazione simultanea di differenti parti di uno search space. Queste tre tipologie, sono quelle che sono più usate nella risoluzione del problema di scheduling. Altre tecniche metaeuristiche sono la ant colony optimizzation (ACO) rappresentabile con il grafo nella prima figura, la artificial immune system (AIS) e le neural networks (NN). Molti dei metodi proposti per la risoluzione dei problemi HFS sfruttano una semplice strategia che è la restrizione della ricerca allo spazio delle permutazione dei jobs. L'idea è quella di trovare una permutazione di n jobs e costruire uno schedule, come quello mostrato nella figura in alto, assegnando i jobs sulle macchine secondo questo ordine. Uno dei primi articoli che ben spiegano questa idea, è dovuto a Voss dove è usato un dispatching rule per generare un algoritmo TS per risolvere un problema HFS con una sola macchina nel secondo stage.

© RIPRODUZIONE RISERVATA

AI_2012_006_INT@027-029.indd 29
14/05/12 10.37