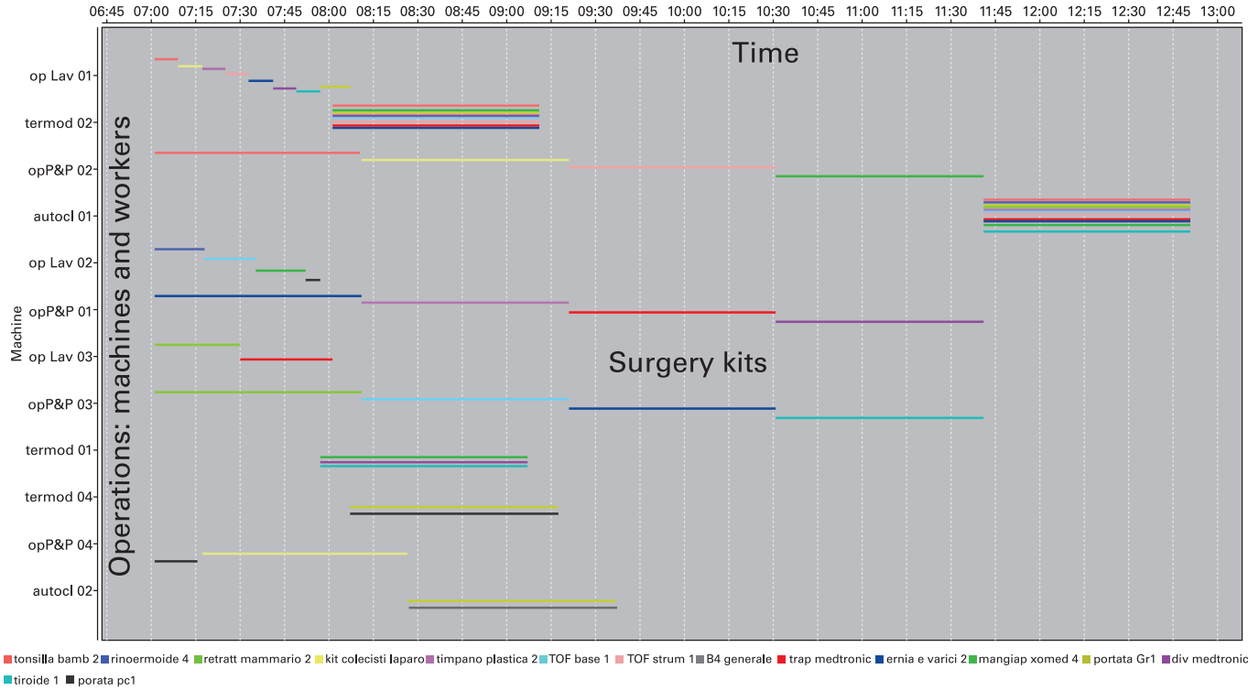

Visualizzazione di un diagramma di gantt che mostra l'impegno temporale delle diverse macchine per le varie lavorazioni rappresentate da tratti di diverse lunghezze e colori.

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