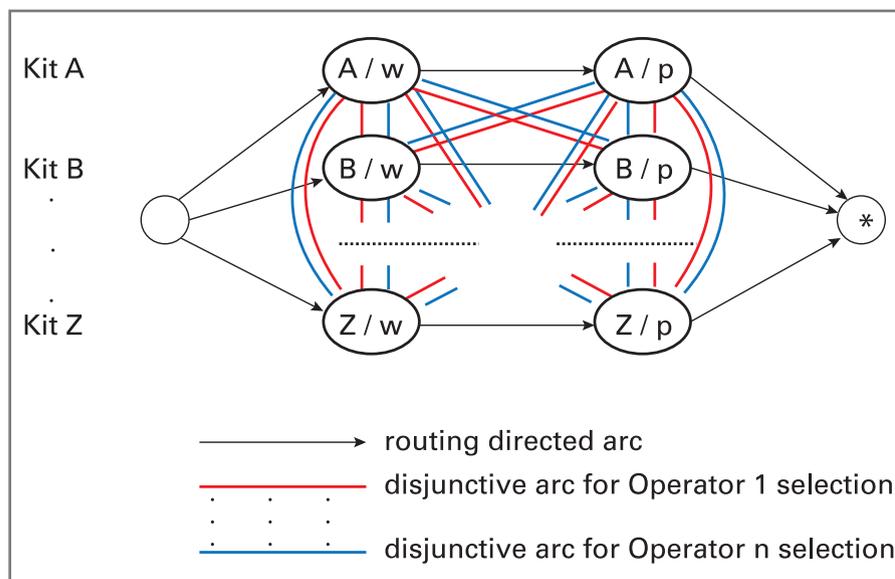


Grafico che mostra il flusso dei prodotti (linee) sulle macchine (nodi) ai fini della creazione del percorso ottimale (scheduling) e deposizione del feromone.

- Ogni formica può nascere con uno stato di partenza  $S_{start}^k$  e avere una o più condizioni di morte  $e^k$ .
- Le formiche esplorano continuamente il grafo costruendo in modo incrementale una soluzione fino a quando per almeno una formica non sono verificate le condizioni di morte.
- Una formica può emettere una scia di feromone mentre esplora il grafo: questa procedura si definisce aggiornamento attivo passo per passo del feromone. In alternativa può esplorare il grafo, trovare una soluzione e poi depositare i feromoni seguendo il percorso inverso. Questa invece è detta procedura di aggiornamento attivo ritardato del feromone.

Le formiche esplorano in modo asincrono il grafo; la scelta del percorso viene fatta sulla base di un processo stocastico che è definito tramite informazioni locali a ogni nodo contenute nella ant routing table. Una buona soluzione si trova in base ai risultati collettivi: quando una



formica trova una soluzione, la valuta e deposita la sua valutazione nella scia di feromone sul percorso che ha seguito. Questa informazione verrà poi ripresa da altre formiche. È previsto inoltre il fenomeno della evaporazione del feromone, un processo che fa decrescere l'intensità  $\tau_{ij}$  con il passare del tempo, impedendo che l'algoritmo converga troppo velocemente verso una soluzione e regione sub-ottima, consentendo così l'esplorazione di nuove regioni. L'ant colony optimization, in conclusione, è un algoritmo il cui punto di forza consiste nell'essere un sistema intelligente distribuito. Ciò significa che le scelte non sono prese da un'unica intelligenza che lavora al

problema ma da una colonia che pensa autonomamente e grazie ai feromoni riesce a condividere le soluzioni appena trovate, adottando dinamicamente il processo di ottimizzazione.

### Descrizione software

Il software realizzato implementa un'ACO riferita a un Flexible Flow Shop. È stato sviluppato per il particolare caso in esame ma è molto flessibile e, tramite minime modifiche del codice, può essere adattato a svariate tipologie di problemi. L'applicazione è costituita da alcune componenti "statiche" (matrici e vettori) utilizzate per la memorizzazione dei dati e da altre "dinamiche" (liste, strutture e puntatori) utilizzate per renderne più agevole e rapida l'esecuzione.

Vengono forniti in ingresso i seguenti dati:

- Numero di job;
- Numero di operazioni per job;
- Numero di macchine;
- Parametro  $\beta$ ;
- Valore iniziale del feromone ( $\tau_0$ );
- Valore dell'evaporazione ( $\rho$ );
- Numero di ants (NA);
- Numero di cicli da effettuare (num\_cicli);
- Tempi di lavorazione delle singole operazioni;
- Tempi di setup relativi alle diverse operazioni;

e si determina il makespan tramite una Ant Colony Optimization.

Nel caso d'esempio del progetto s'impone che ci sia una sola macchina, che i tempi di setup siano nulli per tutte le operazioni e che i tempi di lavorazione



Per capire dove le formiche si trovano in un certo istante, si introduce il concetto di stato mentre per valutare globalmente cosa fanno si parlerà di soluzione ammissibile.