**ESAME BA 16/01/2025 CLOSED BOOK – 90min**

**Domanda 1**
Consideriamo il problema di assegnamento generalizzato, in cui sono dati un set di job e un set di macchine su cui eseguirli. Per ogni coppia job-macchina abbiamo un costo e un tempo di esecuzione. Per ogni macchina abbiamo il tempo disponibile (capacità). Si tratta di assegnare ogni job a una macchina, in modo da minimizzare il costo complessivo, rispettando i vincoli di capacità.

1. Formulare il problema come modello lineare intero.
2. Consideriamo il rilassamento lagrangiano (dualizzazione) dei vincoli di capacità, e analizzare il problema rilassato risultante e la sua soluzione. Si tratta di un problema facile o difficile?
3. Come si possono ottimizzare i moltiplicatori di Lagrange (variabili duali)?

**Soluzione**
Introducendo variabili binarie $x_{im}$, che indicano l'assegnamento del job $i$ alla macchina $m$, e introducendo i dati necessari (tempo di lavorazione e costo per ogni coppia job/macchina e la disponibilità di ogni macchina), otteniamo il modello:

$$\begin{aligned} \min & \sum_i \sum_m c_{im} x_{im} \\ st & \sum_m x_{im} = 1 \quad \forall i \\ & \sum_i p_{im} x_{im} \leq R_m \quad \forall m \end{aligned}$$

Introducendo moltiplicatori $\mu_m \geq 0$ per ogni macchina e dualizzando il secondo vincolo otteniamo il problema rilassato

$$\begin{aligned} \min & \sum_i \sum_m c_{im} x_{im} + \sum_m \mu_m \left( \sum_i p_{im} x_{im} - R_m \right) \\ st & \sum_m x_{im} = 1 \quad \forall i \end{aligned}$$

Riscrivendo la lagrangiana come

$$\sum_i \sum_m (c_{im} + \mu_m p_{im}) x_{im}$$

si vede che il problema diventa banale. Basta assegnare ogni job alla macchina più economica, dove al costo vero si aggiunge il prezzo ombra della risorsa moltiplicato per il tempo di lavorazione.

Tale scelta può comportare il non soddisfacimento del vincolo di capacità di una macchina, nel qual caso occorre incrementare il prezzo ombra. Al contrario, se la risorsa non è pienamente utilizzata. Quindi applicando il metodo del subgradiente, otteniamo lo schema iterativo

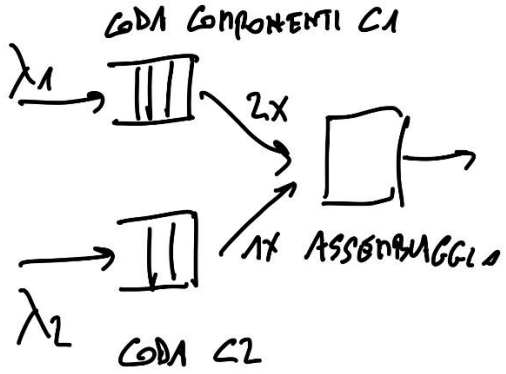
$$\mu_m^{(k+1)} = \max\left\{0, \mu_m^{(k)} + \alpha \left( \sum_i p_{im} x_{im}^{(k)} - R_m \right)\right\}$$

per un passo $\alpha > 0$, dove $x_{im}^{(k)}$ è la soluzione del problema rilassato con prezzi ombra $\mu_m^{(k)}$.

**Domanda 2**
Spiegare (sinteticamente) il modello di base di Hotelling (strada lineare).

**Soluzione**
Vedere slide di teoria.

**Domanda 3**




Il disegno schematizza un sistema di assemblaggio. Ogni finito richiede 2 componenti di tipo C1 e 1 componente di tipo C2. Quando i componenti necessari sono disponibili nelle code (buffer) davanti alla stazione di assemblaggio, viene assemblato un pezzo finito, cosa che richiede un tempo uniformemente distribuito tra limiti noti. Se non sono disponibili i componenti, la stazione finale si ferma in attesa.

I componenti arrivano ai buffer secondo due processi di Poisson (indipendenti), con tassi noti.

Scrivere un simulatore che permetta di valutare, su un orizzonte temporale dato, il numero di pezzi mediamente prodotti nell'unità di tempo (scegliete voi quale) e, per ognuno dei due buffer da, il numero massimo di pezzi in coda osservato durante la simulazione (partiamo da code vuote).

Se può essere utile, qui un memo su funzioni e costrutti MATLAB (NB: **NON** siete tenuti a usare queste funzioni o modi di programmare).

```
>> help min
 min    Minimum elements of an array.
    M = min(X) is the smallest element in the vector X.
    [M,I] = min(X) also returns the indices corresponding to the
minimum values.

>> help case
 case SWITCH statement case.
    case is part of the SWITCH statement syntax, whose general form
is:

        SWITCH switch_expr
          case case_expr,
            statement, ..., statement
          case {case_expr1, case_expr2, case_expr3,...}
            statement, ..., statement
         ...
          OTHERWISE,
            statement, ..., statement
        END

    See also switch, if, else, elseif, while, end.
```

**Soluzione**
Possiamo considerare i seguenti eventi:

1. Arrivo di un componente C1
2. Arrivo di un componente C2
3. Completamento di un assemblaggio

Lo stato contiene:

- Il clock di simulazione.
- Un vettore di tre componenti che contiene per ogni tipo di evento il prossimo (eventualmente infinito se l'evento non è schedulato, nel caso del completamento)
- Lo stato della macchina di assemblaggio (busy/idle)
- La lunghezza delle due code di componenti

Si introducono anche variabili che contengono le statistiche (max lunghezza per le due code, numero di finiti assemblati).

La gestione degli eventi è, nei tre casi di cui sopra:

1. Schedulo il prossimo arrivo C1. Se il server è busy o manca il componente C2, incremento la coda C1 (e verifico se devo aggiornare il massimo della lunghezza di coda). Se il server è idle e l'altro componente è disponibile, decremento coda C2 e schedulo completamento assemblaggio.
2. Come per evento 1, scambiando i componenti.
3. Incremento contatore pezzi prodotti. Se ci sono componenti disponibili nelle due code, decremento le lunghezze e schedulo il prossimo completamento. Altrimenti passo server a stato idle, e metto infinito nel vettore degli eventi (evento deschedulato).

Vedere una possibile bozza di codice MATLAB nel file `assembly.m`