

Domanda 1

Consideriamo un problema di programmazione della produzione con incertezza sulla qualità dei prodotti. Per esempio, se si pianifica un lotto di 100 pezzi, quelli davvero utilizzabili per soddisfare la domanda dei clienti potrebbero essere un numero tra 75 e 95. Supponiamo di avere degli scenari discreti, con probabilità data, che ci danno la percentuale di pezzi buoni per ogni tipo di prodotto.

I diversi prodotti richiedono un certo tempo di lavorazione su un insieme di macchine. Non consideriamo tempi di setup e facciamo finta che la quantità prodotta possa essere rappresentata da una variabile continua (non ristretta quindi ai soli valori interi). Per ogni prodotto abbiamo anche il costo di produzione e il prezzo di vendita. Consideriamo un solo periodo di produzione a cui segue il soddisfacimento della domanda su un solo periodo.

La domanda è assunta deterministica e rappresentata da un insieme di ordini cliente. Ciascun ordine elenca i prodotti richiesti e la loro quantità (un ordine potrebbe richiedere 10 pezzi A, 20 pezzi B e 3 pezzi C). Gli ordini non possono essere soddisfatti parzialmente. O si soddisfa l'ordine intero, per tutti i prodotti richiesti, o l'ordine è perso.

Costruire un modello di programmazione stocastica a due stadi per la massimizzazione del profitto atteso.

Soluzione

Indici: i prodotti; j ordini; m macchine; s scenario.

Dati: d_{ij} domanda di prodotto i in ordine j ; p_j ricavo da ordine j (facile da calcolare sulla base dei dati); c_i costo prodotto i ; r_{im} fabbisogno risorsa m per prodotto i ; A_m disponibilità risorsa m ; π^s probabilità scenario s ; q_j^s resa (percentuale pezzi buoni) per prodotto i in scenario s .

Variabili decisionali: $x_i \geq 0$ quantità prodotta di i ; $\delta_j^s \in \{0,1\}$ se soddisfiamo ordine j in scenario s .

$$\begin{aligned} \max \quad & \sum_i \sum_s \pi^s p_j \delta_j^s - \sum_i c_i x_i \\ & \sum_i r_{im} x_i \leq A_m \quad \forall m \\ & \sum_j d_{ij} \delta_j^s \leq q_j^s x_i \quad \forall i, s \end{aligned}$$

Domanda 2

Spiegare come si applica l'idea della decomposizione lagrangiana a un modello di programmazione stocastica multistadio secondo l'approccio progressive hedging.

Domanda 3

Un chiosco vende panini caldi, la cui preparazione richiede un tempo uniformemente distribuito tra limiti dati (supponiamo tra 90 e 120 secondi). Chi prepara i panini li mette in uno scaldavivande che ha un numero limitato di posti, diciamo 6. Quando tutti i posti sono occupati, chi prepara i panini si ferma. La produzione di panini riparte quando si libera un posto, a seguito della richiesta di un cliente.

I clienti arrivano secondo un processo di Poisson con rate costante (supponiamo un cliente ogni 4 minuti in media) e chiede uno, due o tre panini (diciamo con probabilità uniformi per semplicità). Se il numero desiderato di panini è già pronto, il cliente li ritira e se ne va. Altrimenti aspetta fino a quando il numero di panini richiesti è pronto. La disciplina di coda è FIFO e i clienti sono pazienti (non se ne vanno e aspettano il tempo necessario).

Scrivere un programma MATLAB (script o funzione) che valuti il tempo medio di attesa in coda.

Suggerimento: MATLAB gestisce dinamicamente l'allocazione di memoria per gli array:

```
>> V = [3 5 7]
V =
     3     5     7
>> V(4)=9
V =
     3     5     7     9
>> V(1)=[]
V =
     5     7     9
>> V(1)=[]
V =
     7     9
```

Non è il massimo dell'eleganza e dell'efficienza, ma se vi serve potete usare questa caratteristica.

Soluzione

Può essere utile, prima di scrivere il codice, **chiarirsi come descrivere lo stato del sistema**, quali sono gli **eventi rilevanti e come questi alterano lo stato**.

I due eventi sono l'arrivo di un cliente e il **completamento di un panino**. Il tempo in cui si verifica il prossimo di questi due eventi fa parte dello stato, come pure il **clock di sistema**.

Lo stato comprende anche la coda dei clienti, con l'informazione del **tempo di arrivo in coda** e il **numero di panini ancora da ritirare**. Facendo le cose in modo grezzo **usiamo due array per queste informazioni**. Ci serve anche sapere **se il sistema è bloccato**, ovvero il buffer di panini è pieno e quindi si blocca la produzione.

Gestione dell'evento completamento panino:

- Se c'è cliente in coda, **diminuisci la sua richiesta**. Se domanda completata, **rimuovilo e registra il tempo di attesa in coda**.
- Altrimenti, **aumenta il numero di panini nel buffer**. Se il buffer è pieno, **setta lo stato di blocco** e poni a infinito il tempo del prossimo completamento. **Altrimenti schedula il prossimo evento di completamento**.

Gestione dell'evento arrivo cliente:

- **Schedula il prossimo arrivo cliente**.
- **Genera domanda**.
- Se c'è cliente in coda, **mettiti in coda registrando il tempo di arrivo**.
- Altrimenti, **se ci sono tutti i panini richiesti, prendili** (diminuendo la dimensione del buffer e se necessario sbloccando la produzione dei panini e schedulando il prossimo completamento) ed esci dal sistema, registrando waiting time zero.
- **Altrimenti prendi i panini che ci sono** (con i dati del problema non è necessario sbloccare la produzione, ma per generalità lo si potrebbe fare) **e mettiti in coda** (in testa, registrando tempo di arrivo e domanda residua).

Vedere codice MATLAB.

NB: il codice non è il massimo dell'eleganza, e ci sono soluzioni alternative.

Il punto non è fare codice perfetto (e nemmeno funzionante e a prova di bomba), ma impostare correttamente la soluzione.