Dipartimento di Ingegneria dell'Informazione
Ricerca Operativa (INFLT, ETELT)
Anno accademico 2020/2021

Elaborato sull'utilizzo del risolutore Gurobi - Parte II

Suggerimenti

1. Per rispondere ai quesiti proposti, dovete sfruttare le conoscenze teoriche fino ad ora acquisite riguardanti la Programmazione Lineare Intera.
2. Relativamente al codice che andrà scritto in Java:
   - 2.1 se lo ritenete opportuno, potete importare librerie Java esterne per rendere più rapido qualche calcolo o procedimento;
   - 2.2 attenzione all'errore di macchina: potrebbe capitare che Gurobi vi calcoli un valore di una variabile pari a `0.9999999999`; ciò vuol dire che, nella realtà, il valore di quella variabile è pari a `1`;
   - 2.3 approssimare ogni valore calcolato alla quarta cifra decimale per arrotondamento.
   - 2.4 come sottolineato a lezione, fare attenzione alle variabili di surplus che Gurobi definisce **negative**.

Istruzioni

1. **Ogni risposta ai quesiti deve essere frutto di una o più linee di codice** (**non è consentito svolgere calcoli "a mente", su carta o tramite altri software e poi semplicemente stampare a video le risposte**)
2. Potete utilizzare qualsiasi classe e metodo forniti dall'interfaccia Java di Gurobi (cfr. documentazione: `https://www.gurobi.com/documentation/9.1/refman/index.html`).
3. Il codice sorgente prodotto dovrà essere **debitamente commentato**, evidenziando, a grandi linee, le rispettive tre parti di codice che sono servite per rispondere ai tre quesiti.
4. Il file eseguibile `.jar`, rinominato `gruppoX.jar`, dovrà produrre in output il file `risposte_gruppoX.txt` con la stampa delle risposte ai tre quesiti, secondo il formato descritto in Pagina 3.
5. Allegate infine una descrizione sintetica (**non più di mezza pagina**) di quanto avete fatto e delle modalità (algoritmi, regole teoriche) utilizzate per risolvere ciascun quesito.
6. Non è possibile contattare il docente o gli assistenti per richieste relative alla parte teorica o alla stesura del codice, mentre è possibile chiedere eventuali chiarimenti inerenti alla consegna.

CONSEGNA

La consegna è prevista **entro le 23:55 del 6 giugno 2021**. Devono essere caricati in Comunità Didattica, tramite l'oggetto "Consegna elaborato Gurobi - Parte II", **il codice sorgente Java** prodotto, il **file esegubile** `.jar` e la **descrizione sintetica**. L'elaborato del gruppo di chi non avesse caricato tutto il materiale richiesto entro il tempo limite sarà considerato **insufficiente**.

Testo del problema

La banca di Matriciopoli è in continua espansione e la mole di dati provenienti dalle m filiali da dover elaborare a fine giornata è diventata ingente. Per questo motivo, il reparto ICT ha deciso di affidarsi a una *grid computazionale* (= aggregazione di risorse di calcolo provenienti da domini di sicurezza e gestione differenti) per fornire a ogni filiale potenza di calcolo on-demand. Identifichiamo con il termine di *processo* l'elaborazione dei dati richiesta da una filiale. La grid computazionale acquistata consta di k processori e di dispositivi di storage per un totale di g GB di memoria. Ogni processo (identificato dalla filiale di provenienza) può avere n diverse configurazioni di esecuzione. Ogni configurazione differisce dalle altre per numero di processori e di GB necessari per eseguirla. Il processo p può quindi essere eseguito scegliendo, tra le n possibili, l'$i-esima$ configurazione che prevede l'utilizzo di a_{pi} processori e b_{pi} GB di memoria. Associato a una configurazione i vi è inoltre un profitto q_{pi} che indica quanto la filiale la ritenga vantaggiosa. La banca vuole decidere quale configurazione di utilizzo della grid scegliere per ogni filiale, massimizzando il profitto totale ottenuto, assicurandosi che almeno una risorsa della grid (processori o memoria) sia utilizzata almeno al 90%.

Quesiti

I Elaborare un modello matematico di Programmazione Lineare Intera per il problema fornito. Quindi, implementarlo e risolverlo tramite Gurobi, trovandone la soluzione ottima (**valore ottimo delle variabili**), il corrispettivo **valore della funzione obiettivo**, e riportando le informazioni indicate nell'esempio di output testuale.

II Fornire, se esiste, una soluzione ottima alternativa a quella trovata al punto I.

III Trovare il numero minimo k_{min} di processori necessari affinché il problema formulato al punto I abbia almeno una soluzione ammissibile.

Esempio di output `risposte_gruppoX.txt`

```
GRUPPO <numero gruppo>
Componenti: <cognome componente 1> <eventuale cognome componente 2>


QUESITO I:
funzione obiettivo = <valore funzione obiettivo>
filiale p_1: <profitto ottenuto per p_1>
filiale p_2: <profitto ottenuto per p_2>
...
filiale p_m: <profitto ottenuto per p_m>
processori inutilizzati = ?
GB inutilizzati = ?

QUESITO II:
filiale p_1: <profitto ottenuto per p_1>
filiale p_2: <profitto ottenuto per p_2>
...
filiale p_m: <profitto ottenuto per p_m>
processori inutilizzati = ?
GB inutilizzati = ?

(se non esiste una soluzione ottima alternativa, stampare solo la dicitura NON ESISTE)

QUESITO III:
k_min = <valore>
```

NB: le risposte non stampate in questo file verranno considerate **in bianco** e quindi valutate negativamente.