

Elaborato sull'utilizzo del risolutore Gurobi - Parte I

Suggerimenti

1. Per rispondere ai quesiti proposti, dovete sfruttare le conoscenze teoriche fino ad ora acquisite riguardanti la Programmazione Lineare.
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 18 aprile 2021**. Devono essere caricati in Comunità Didattica, tramite l'oggetto “Consegna elaborato Gurobi - Parte I”, il **codice sorgente Java** prodotto, il **file eseguibile** .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

Alla fine di ogni giornata lavorativa, la banca di Matriciopoli deve aggiornare gli estratti conto di migliaia di carte di credito. Si stima che ognuna delle h filiali in media debba processare g TB di dati. Il reparto ICT, raccolti tutti i dati di ogni filiale, li invia ad una rete di n calcolatori che possono lavorare in parallelo e deve quindi decidere come suddividere il carico. Ciascun calcolatore i , $i = 1, \dots, n$ ha una capacità di elaborazione pari ad α_i GB/s e i dati possono essere inviati ad esso con una velocità di β_i GB/s. La banca vuole che in ogni caso ogni calcolatore elabori almeno l' $\Omega\%$ e al massimo il $\Theta\%$ dei dati totali e che le elaborazioni siano terminate entro τ secondi. La banca vuole minimizzare il massimo carico di lavoro assegnato tra tutti i calcolatori.

NB: si intenda il carico di lavoro come quantità reale.

Quesiti

I Elaborare un modello matematico di Programmazione Lineare per il problema fornito. Quindi, implementarlo e risolverlo tramite Gurobi, trovandone la soluzione ottima (**valore ottimo delle variabili** e il corrispondente **valore della funzione obiettivo**).

II Indicare:

- le **variabili in base** e quelle **fuori base** all'ottimo
- i **coefficienti di costo ridotto** all'ottimo
- se la soluzione ottima trovata è multipla e/o degenere
- specificare quali vincoli identificano il vertice ottimo e quali no.

NB: se il problema assegnatovi ha k variabili originali, la $(k+i)$ -esima deve essere la variabile di slack/surplus associata all' i -esimo vincolo.

III Sia m il numero di vincoli del vostro problema in forma standard. Generare in maniera casuale tre matrici $m \times m$ e affrontare per ciascuna di esse i seguenti quesiti:

- verificare se la matrice sia di base ed in caso determinarne la rispettiva soluzione di base;
- verificare se la soluzione di base trovata è ammissibile e ottima;
- determinare il tableau associato alla soluzione di base trovata.

NB: è necessario scrivere il codice per la verifica e la determinazione del tableau anche nel caso in cui le tre matrici generate casualmente non siano di base.

Esempio di output risposte_gruppoX.txt

GRUPPO <numero gruppo>

Componenti: <cognome componente 1> <eventuale cognome componente 2>

QUESITO I:

funzione obiettivo = <valore funzione obiettivo>

soluzione di base ottima: [<valore di x_1 >, <valore di x_2 >, ..., <valore di x_q >]

QUESITO II:

variabili in base: [<1 se x_1 è in base, 0 altrimenti>, ... <1 se x_q è in base, 0 altrimenti>]

coefficienti di costo ridotto: [<costo ridotto di x_1 >, ..., <costo ridotto di x_q >]

soluzione ottima multipla: <Sì|No>

soluzione ottima degenerare: <Sì|No>

vincoli vertice ottimo: <elenco dei nomi dei vincoli>

QUESITO III:

(per ogni matrice generata stampare:)

indici variabili scelte: [a, b, c,...]

base: <Sì|No>

(se matrice di base stampare:)

soluzione: [<valore di x_1 >, <valore di x_2 >, ..., <valore di x_q >]

ammissibile: <Sì|No>

ottima: <Sì|No>

tableau:

	x_1	x_2	x_3	...	
x_i	a_{11}	a_{12}	a_{13}	...	b_i
x_j	a_{21}	a_{22}	a_{23}	...	b_j
x_z	a_{31}	a_{32}	a_{33}	...	b_z
...
	r_1	r_2	r_3	...	o

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

NB: $q = m + k$