RICERCA OPERATIVA

**LEZIONE 1: 19/09/2022**

**Modalità esame**: due prove scritte, si possono sostenere separatamente. Una volta superata una parte quella rimane valida per sempre (nel caso volessi rifarlo annullerei quella parte).

2 esercizi da svolgere, parte di vero o falso, domande chiuse (possono capitare domande aperte).

Per il punteggio si prende la media dei voti scritti a cui si aggiunge il punteggio di un progetto.

Il progetto può valere fino a 4 punti (possono essere fatti in momenti distinti) e può essere fatto in coppia.

**Introduzione:** 1° parte relativa alla modellizzazione mentre la seconda alla risoluzione (studio di algoritmi partendo dalle basi matematiche) di problemi di decisioni complesse. La complessità è determinata dall’ampiezza dello spazio delle scelte possibili (non si occupa di problemi complessi in cui lo spazio delle decisioni è limitato a poche scelte)

Per affrontare un problema di decisione il ricercatore operativo compie questi passi (già visti in modelli e algoritmi):

1. Individuo componenti del problema
2. Creo modello matematico del problema
3. Individuo un algoritmo di risoluzione
4. Validazione del modello

I dati sono tutto ciò che è noto a priori e non è sotto il controllo delle decisioni, le variabili sono le quantità sotto il diretto controllo del decisore. I vincoli sono le condizioni che limitano le possibili scelte del decisore. L’obiettivo è il criterio attraverso il cui il decisore confronta le sue scelte.

(parte analoga al corso di modelli e algoritmi)

Esempio:

Immagine che contiene tavolo

Descrizione generata automaticamente

**Individuo le componenti del problema**

Dati: i valori nella tabella. -> spesso sono già in formato matematico

Variabili: quantità di pacchi da realizzare. -> “riscrivo in matematichese”

Vincoli: disponibilità massima dovute dalle risorse. -> riscrivo come disequazione.

obiettivo: il valore di utilità totale da massimizzare. -> riscrivo come funzione.

**Costruisco modello matematico**

Bisogna tradurre nel linguaggio matematico (seguire le stesse regole usate nel corso modelli e algoritmi) -> modelli di *programmazione matematica*

La **regione ammissibile** è l’insieme dei punti che soddisfano tutti i vincoli e viene indicata con S.

I modelli di **problemi di programmazione lineare (PL)** e **problemi di programmazione lineare intera (PLI)**.

In entrambi le funzioni (sia quelle obiettivo che quelle dei vincoli) sono funzioni lineari, la differenza sta nel fatto che nel PLI sono presenti solo variabili che possono assumere valori interi.

**N.B**: I PLI sono più difficili da risolvere rispetti ai PL

Molti problemi reali hanno un modello di PL o PLI (più spesso PL che PLI).

**LEZIONE 2: 20/09/2022**

Una volta adattato il problema ad un modello matematico su carta esso viene passato ad un **risolutore** (prima del passaggio viene fatta una rappresentazione del modello con le strutture dati del risolutore, passaggio delicato).

La *scrittura del modello in linguaggio AMPL* permette di scrivere il modello matematico su carta per un traduttore in maniera ‘’semplice’’ e può essere utilizzato per più risolutori (mediante appositi *traduttori*).

L’utente specificherà il risolutore tramite il comando

AMPL: option solver nome\_risolutore;

Per problemi di PL e PLI i risolutori più utilizzati sono: *cplex,gurobi,* e *Ipsolve*. (gurobi più consigliato).

**ESEMPIO: PROBLEMA DIETA**

Sia dato un insieme PROD di prodotti ed un insieme di SOST di sostanze nutritive. Si sa che in un’unità del prodotto j si trova una quantità quant\_unit\_ij della sostanza nutritiva i. Si sa che il costo di un’unità di prodotto j è pari a cost\_unit\_ij. Tenendo conto che una dieta deve contenere una quantità minima quant\_min\_i di ciascuna sostanza nutritiva, si vuole determinare una dieta che abbia costo minimo.

Per quanto riguarda i dati si tratta di parametri numerici, che non hanno bisogno di traduzione in linguaggio matematico.

Le variabili saranno tradotte attraverso delle variabili matematiche del tipo

Il vincolo viene tradotto attraverso la seguente disequazione:

Inoltre bisogna aggiungere il vincolo (IMPLICITO) che ogni quantità di prodotto deve essere 0

Il nostro obiettivo è quello di trovare il:

Adesso è possibile procedere alla traduzione in **linguaggio AMPL**

**PARTI FONDAMENTALI DI AMPLS**

* *Insiemi di indici*: nel caso del problema della dieta sono i due insiemi PROD dei prodotti e SOST delle sostanze nutritive.
* *Parametri:* sono tutte le quanittà il cui valore è noto prima di risolvere il problema (stesso significato dei DATI), nel problema della dieta sono i costi unitari dei prodotti, le quantità minime delle sostanze nutritive e la quantità di sostanza.
* *Variabili*: stesso significato già visto delle variabili.
* *Vincoli:* anche qui stesso significato
* *Obiettivo*: stessa cosa.

**SINTASSI**

Un *insieme T* si dichiara (**N.B:** oltre ad essere dichiarato deve essere anche definito) attraverso la seguente sintassi:

set T;

nell’esempio della dieta:

set PROD;

set SOST;

Un parametro α si dichiara nel modo seguente:

param α;

Qualora sia noto che il parametro è sempre positivo conviene indicarlo esplicitamente nella dichiarazione nel modo seguente

param α>0;

Possono essere utilizzati anche altri tipi di vincoli (minore, minore uguale, etc.)

param α>0 integer;

indica un parametro maggiore di zero ed intero

È possibile dichiarare *vettori di parametri* con indici in un insieme T attraverso la seguente dichiarazione:

param α{T};

nel caso gli indici siano interi da 1 a n si può scirvere:

param α{1…n};

si possono definire anche array bidimensionali con insiemi di indici e nel modo seguente:

param α{ , };

Applicando quanto detto all’esempio avremo:

**param** cost\_unit{PROD}>0;

**param** quant\_unit{SOST,PROD}>= 0;

**param** quant\_min{PROD};

La dichiarazione delle *variabili* è del tutto analoga a quella dei parametri, la parola **param** viene sostituita da **var**.

Nel nostro esempio abbiamo un vettore di variabili x con indici in PROD che sarà definito in questo modo:

**var** x{PROD} >= 0;

È possibile aggiungere la parola chiave **integer** anche per le variabili, questo passaggio distingue un problema PL da PLI  
Supponiamo di avere una variabile con limiti superiori ed inferiori allora avremo:

**var** x{i **in** PROD}>=0, <= max\_prod[i];

la parola chiave **in** indica l’appartenenza della variabile x all’insieme dei prodotti, la *i* viene inserita anche in max\_prod perché indica la componente del vettore a cui applicare il vincolo.

Un *vincolo* viene dichiarato nel seguente modo

**Subject to** nome\_vincolo **:** formula\_vincolo;

Una collezione di vincoli indicizzata su un insieme di indici I viene dichiara in questo modo

**Subject to** nome\_vincolo [i **in** I]**:** formula\_vincolo;

Per il nostro esempio:

**Subject to** min\_sostanza {i in SOST}**:** **sum** {j in PROD} quant\_unit[i,j]\*x[j] >= quant\_min[i];

**sum** indica la sommatoria nella formula matematica ricavata prima

L’*obiettivo* si dichiara nel modo seguente:

**maximize** nome\_obiettivo : formula obiettivo;

(nel caso si debba minimizzare si usa la parola chiave **minimize** al posto di **maximize**).

I *commenti* sono indicati tra **###**. La definizione del modello generico appena vista viene salva in file .MOD

Una volta costruito il modello bisognerà inserire in un altro file i valori di insiemi e parametri, ovvero i dati di input dell’**istanza** del nostro problema in un file .DAT

Supponiamo di avere a disposizione i seguenti dati:

L’insieme PROD conteine pasta, verdura, carne

L’insieme SOST contiene vitamine,proteine

Ununità di verdura e carne costano rispettivamente 3,2 e 5

Le quantità minime di vitamine e proteine sono rispettivamente 8 e 6;

Le quantità di vitamine in un’unità di pasta, verdura o carne sono rispettivamente 0.3 e 0.4

Per la **definizione di un insieme T** conentente gli oggietti t1,t2,.., tn si usa la seguente sintassi:

**set** T **:=** t1,t2,…tn;

nel nostro esempio avremo:

**set PROD :=** pasta verdura carne;

**set SOST :=** vitamine proteine;

Per quanto riguarda la **definizione dei parametri** si usa la seguente sintassi:

**param a :=** valore\_parametro; **###PARAMETRO SINGOLO**

**param** a **:=** **###ARRAY UNIDIMENSIONALI DI PARAMETRI**

t\_1 valore\_1

…

t\_n valore\_n;

**param** a**: ###ARRAY BIDIMENSIONALI DI PARAMETRI**

s1 …. s\_m **:=** (formattazione in questo modo non obbligatoria)

t1 val (t1,s1) … val(t1,sm) (s\_m sono le colonne, t\_n sono le righe)

t\_n val(t\_n,s1) … val(t\_n,s\_m)

Nell’esempio:

**param** cost\_unit:=

Pasta 3

Verdure 2

Carne 5;

**param** quant\_min :=

vitamine 8

proteine 6;

**param** quant\_unit :

pasta verdure carne **:=**

vitamine 0.3 0.5 0.4

proteine 0.5 0.2 0.7

L’utilizzo di due file separati (.MOD e .DAT) permette la riusabilità dello stesso modello per più istanze del problema (con dati diversi).

I comandi per risolvere il problema appena introdotto tramite AMPL sono:

1. *reset*; (conviene farlo sempre inizialmente perché AMPL tiene in memoria l’ultimo modello e gli ultimi dati caricati) oppure *reset data*; se si vuole utilizzare lo stesso modello ma con dati diversi.
2. *option solver* *gurobi*; (al posto di gurobi può essere usato un altro risolutore)
3. *model dieta.mod;* (carichiamo il modello definito)
4. data dieta.dat; (carichiamo i dati)
5. solve; (risolve il problema)

Questo codice restituisce il seguente messaggio:

optimal solution; objective 45.2631 (valore funziona obiettivo).

2 simplex iterations (indica che ha utilizzato due iterazioni dell’algoritmo del simplesso).

Così facendo abbiamo ottenuto il **valore ottimo**, per ottenere la **soluzione ottima** bisogna display X; (con x nome della variabile).

E’ possibile anche scrivere tutti i comandi visti in un file .RUN in cui è possibile inserire altri costrutti sintattici più avanzati (statement, IF, cicli, etc.).

**LEZIONE 2: 23/09/2022**

**Seconda parte delle slide AMPL**

È scritto tutto nelle slide. (fino all’ultima slide)

**Modello matematico - Introduzione**

È la traduzione di un problema di decisione, di cui si ha una descrizione a parole, nel linguaggio matematico: le varie componenti del problema di decisione vengono tradotte in oggetti matematici come insiemi, numeri, variabili, equazioni e/o disequazioni, funzioni matematiche.

Nello specifico si parlerà di

* **variabili binarie** -> sono particolarmente importanti nella creazione di modelli di problemi di decisione, con i relativi vincoli logici
* **non linearità eliminabili** -> sono funzioni non lineari sostituibili da equivalenti espressioni lineari.
* **Problemi di particolare rilevanza nelle applicazioni pratiche.**

**Modello matematico – variabili binarie**

Possono assumere solo due valori (convenzionalmente 0 ed 1), vengono utilizzate nei problemi di decisione.

Supponiamo che nel nostro problema di decisione una certa variabile x abbia una *limitazione superiore* pari a B se ci si trova in uno tra due possibili stati.

La scelta i due possibili stati viene modellata con una variabile binaria δ e possiamo imporre che lo stato relativo a δ = 1 sia quello per cui x non può superare B. In altre parole, abbiamo la seguente relazione tra δ e x

‘’Se mi trovo in quel particolare stato allora ho la limitazione’’ –> vincolo di tipo logico, deve essere tradotto per essere accettabile dai modelli di programmazione matematica.

Nel caso delta sia uguale ad 1 abbiamo:

Nel caso fosse uguale a zero però abbiamo un addendo M, che è un limite superiore esplicito o implicito (ovvero derivato da altri vincoli del problema) sui valori che possono essere assunti da x *indipendentemente* dallo stato del sistema.

Supponiamo ora che una certa variabile x abbia una limitazione inferiore pari ad A se ci si trova in uno tra due possibili stati.

Di nuovo la scelta tra i due possibili stati viene modellata con una variabile binaria δ e possiamo imporre che lo stato relativo a δ = 1 sia quello per cui x non può essere inferiore ad A.

Quindi, abbiamo la relazione:

Dal vincolo logico si passa a quello matematico tramite una disequazione

-M è il limite inferiore esplicito o implicito sui valori che possono essere assunti da x.

(slide 31).