Sommario

Programmazione Lineare & Modellazione 2
Simplesso: Esercizi pratici 4
Domande teoriche sul simplesso 6
Condizioni primale/duale e passaggio al problema duale 8
Enunciare le condizioni di complementarietà primale-duale 8
Dualità 8
Branch and Bound 10
Grafi 11
Bellman-Ford 13
Dijkstra 13
AMPL 14

Programmazione Lineare & Modellazione

SBAGLIATO formulazioni *NON LINEARI*!!!			**Corretto**	
			Vincoli	Domini
if $x_1 > 0$ then $x_2 = 0$ and if $x_2 > 0$ then $x_1 = 0$	$nand(y_1, y_2)$	$x_1 x_2 = 0$ $y_1 y_2 = 0$	$x_1 \le M y_1$ $x_2 \le M y_2$ $y_1 + y_2 \le 1$	$x_1, x_2 \ge 0$ $y_1, y_2 \in \{0,1\}$ $(M \to \infty)$
if $x > 0$ then $y = 1$	$(x > 0) \to (y = 1)$	$x(1-y) = 0$	$x \le My$	$x \ge 0, y \in \{0,$
$y_1 = 1$ or $y_2 = 1$	$y_1 \vee y_2$	$(1-y_1)(1-y_2) = 0$	$y_1 + y_2 \ge 1$	$y_1, y_2 \in \{0,1\}$
$y_1 = 1$ and $y_2 = 1$	$y_1 \wedge y_2$	$y_1 y_2 = 1$	$y_1 + y_2 = 2$	$y_1, y_2 \in \{0,1\}$
$y_1 = 1$ only if $y_2 = 1$	$y_1 \to y_2$	$y_1(1-y_2) = 0$	$y_1 \le y_2$	$y_1, y_2 \in \{0,1\}$
$y_1 = 1$ only if $y_2 = 0$	$y_1 \to \overline{y_2}$	$y_1 y_2 = 0$	$y_1 \le (1-y_2)$	$y_1, y_2 \in \{0,1\}$
$y_1 = 1$ xor $y_2 = 1$	$y_1 \neq y_2$		$y_1 + y_2 = 1$	$y_1, y_2 \in \{0,1\}$
etc. etc. etc.			etc. etc. etc.	

Queste formulazioni sono semanticamente corrette ma

NON ACCETTABILI

in un modello di programmazione lineare

Queste formulazioni sono corrette a patto di
- SPECIFICARE I DOMINI
- ATTIVARE le var. logiche

Modelli inseriti a livello indicativo (non servono per risolvere tutti i modelli, ma a seconda della natura del problema, servono per ricordarsi "che ci vogliono dei vincoli aggiuntivi non esplicitamente presenti nel problema", da mia esperienza):

Modelli di copertura di costo minimo

$$
\begin{aligned}
\min \quad & \sum_{i \in I} C_i x_i \\
s.t. \quad & \\
& \sum_{i \in I} A_{ij} x_i \ge D_j & \forall\, j \in J \\
& x_i \in \mathbb{R}_+ \; [\, \mathbb{Z}_+ \mid \{0,1\} \,] & \forall\, i \in I
\end{aligned}
$$

dove

- I insieme delle risorse da acquistare;
- J insieme delle domande da coprire;
- C_i costo (unitario) per l'utilizzo della risorsa $i \in I$;
- D_j ammontare della domanda di $j \in J$;
- A_{ij} capacità (unitaria) della risorsa i di soddisfare la domanda j.

Modelli di mix ottimo di produzione

$$
\begin{aligned}
\max \quad & \sum_{i \in I} P_i x_i \\
s.t. \quad & \\
& \sum_{i \in I} A_{ij} x_i \le Q_j & \forall\, j \in J \\
& x_i \in \mathbb{R}_+ \; [\, \mathbb{Z}_+ \mid \{0,1\} \,] & \forall\, i \in I
\end{aligned}
$$

dove

- I insieme dei beni che possono essere prodotti;
- J insieme delle risorse disponibili;
- P_i profitto (unitario) per il bene $i \in I$;
- Q_j quantità disponibile della risorsa $j \in J$;
- A_{ij} quantità di risorsa j necessaria per la produzione di un'unità del bene i.

Considerando gli esempi sopra proposti, rientrano in questo schema i problemi di assemblaggio di telefonini e del contadino.

Modelli di trasporto

$$
\begin{aligned}
\min \quad & \sum_{i \in I} \sum_{j \in J} C_{ij} x_{ij} \\
s.t. \quad & \\
& \sum_{j \in J} x_{ij} \le O_i & \forall\, i \in I \\
& \sum_{i \in I} x_{ij} \ge D_j & \forall\, j \in J \\
& x_{ij} \in \mathbb{R}_+ \; [\, \mathbb{Z}_+ \mid \{0,1\} \,] & \forall\, i \in I, j \in J
\end{aligned}
$$

dove

- I insieme dei centri di offerta;
- J insieme dei centri di domanda;
- C_{ij} costo (unitario) per il trasporto da $i \in I$ a $j \in J$;
- O_i ammontare dell'offerta in $i \in I$;
- D_j ammontare della domanda in $j \in J$.

Considerando gli esempi sopra proposti, rientra in questo schema il problema del trasporto di frigoriferi.

<u>Modello multiperiodale</u>

L'effetto di variabili in vincoli precedenti "ritorna" in quelli successivi.

[cfr. Piani di investimento nelle dispense]

$$
\begin{aligned}
& x_{A1} + x_{B1} \le 10000 && \text{(anno 1)} \\
& x_{A2} + x_{B2} + x_{C2} \le 10000 - x_{A1} - x_{B1} && \text{(anno 2)} \\
& x_{A3} + x_{B3} \le 10000 + 0.4x_{A1} - x_{B1} - x_{A2} - x_{B2} - x_{C2} && \text{(anno 3)} \\
& x_{A4} \le 10000 + 0.4x_{A1} + 0.7x_{B1} + 0.4x_{A2} - x_{B2} - x_{C2} - x_{A3} - x_{B3} && \text{(anno 4)} \\
& x_{D5} \le 10000 + 0.4x_{A1} + 0.7x_{B1} + 0.4x_{A2} + 0.7x_{B2} + 0.4x_{A3} + && \\
& \qquad - x_{C2} - x_{B3} - x_{A4} && \text{(anno 5)}
\end{aligned}
$$

Logic to algebra

Statement	Constraint
$\neg P_1$	$\delta_1 = 0$
$P_1 \vee P_2$	$\delta_1 + \delta_2 \geq 1$
$P_1 \oplus P_2$	$\delta_1 + \delta_2 = 1$
$P_1 \wedge P_2$	$\delta_1 = 1,\ \delta_2 = 1$
$\neg(P_1 \vee P_2)$	$\delta_1 = 0,\ \delta_2 = 0$
$P_1 \implies P_2$	$\delta_1 \leq \delta_2$ (equivalent to: $(\neg P_1) \vee P_2$)
$P_1 \implies (\neg P_2)$	$\delta_1 + \delta_2 \leq 1$ (equivalent to: $\neg(P_1 \wedge P_2)$)
$P_1 \iff P_2$	$\delta_1 = \delta_2$
$P_1 \implies (P_2 \wedge P_3)$	$\delta_1 \leq \delta_2,\ \delta_1 \leq \delta_3$
$P_1 \implies (P_2 \vee P_3)$	$\delta_1 \leq \delta_2 + \delta_3$
$(P_1 \wedge P_2) \implies P_3$	$\delta_1 + \delta_2 \leq 1 + \delta_3$
$(P_1 \vee P_2) \implies P_3$	$\delta_1 \leq \delta_3,\ \delta_2 \leq \delta_3$
$P_1 \wedge (P_2 \vee P_3)$	$\delta_1 = 1,\ \delta_2 + \delta_3 \geq 1$
$P_1 \vee (P_2 \wedge P_3)$	$\delta_1 + \delta_2 \geq 1,\ \delta_1 + \delta_3 \geq 1$

20-17

To enforce "if $x = 0$ then $y = 1$": use

$$y \geq 1 - x.$$

Il prof ci tiene particolarmente alla modellazione; fare <u>bene</u> questi esercizi, concentrandosi su questi passi in questo ordine.

Per i 6 crediti (ciò vuol dire per i 7 sì):

- No vincoli spuri
- No dimensionamento variabile big M
- No esercizi dello zaino 0-1

Passi (occorre ragionare in termini <u>di decisioni da prendere)</u>:

1) Individuare quali sono le decisioni da prendere per risolvere il problema, ossia dobbiamo definire le variabili decisionali
    a. Queste variabili possono essere ad uno o a due indici; per esperienza, spesso, le variabili a due indici fanno danni; meglio mettere più variabili ad un indice. Preferibile mettere a due indici quando è evidente dal problema che servono entrambe le colonne di una tabella
    b. Attenzione anche alla scrittura dei vincoli; "indipendentemente da" fa capire che, probabilmente, avremo bisogno di una variabile ad indice solo e non due o dipende dal contesto
2) Scrivere la funzione obiettivo
3) Scrivere almeno uno degli altri vincoli, preferibilmente non gli ultimi. Almeno uno/due tra i primi aggiuntivi si scrivono subito
    a. A detta stessa del prof, almeno uno o due di questi servono per aiutare nella scrittura del modello
4) Scrivere il resto dei vincoli, considerando che:
    a. Se si ha a che fare con vincoli logici del tipo "posso decidere *se fare* la cosa X" va inserita una variabile binaria
        i. Questa variabile va *legata* ad una variabile normale per fare in modo stia in piedi.
        ii. Solitamente questo avviene con la forma big-M → $x_i \leq My_i$
            1. Essi servono a linearizzare un modello <u>evitando i termini quadratici</u> (moltiplicazione o somma di più termini singoli)
    b. Si possono avere diversi casi notevoli, quali:
        i. Costi fissi:
            1. In questo caso, va aggiunto un termine moltiplicativo a tutte le variabili decisionali legate per quantità e si crea una variabile decisionale apposita per costo fisso
        ii. Almeno:

1. Se vincolo logico, usare "insieme di variabili logiche >= numero" e legare le variabili con vincolo di big-M nel modo $x_i \geq M * y_i$
2. Se vincolo normale, usare "variabile decisionale >= numero"

iii. Al massimo:
1. Se vincolo logico, usare "insieme di variabili logiche <= numero" e legare le variabili con vincolo di big-M nel modo $x_i \geq M * y_i$
2. Se vincolo normale, usare "variabile decisionale <= numero"

iv. Esattamente:
1. Se vincolo logico, usare "insieme di variabili logiche = numero" e legare le variabili con vincolo di big-M nel modo $x_i \geq numero * y_i$

v. Penalità/Sconto

vi. *Scelgo* di pagarla, pertanto sarà una variabile binaria legata a variabili logiche già presenti e poi aggiunte in funzione obiettivo

vii. Budget
1. Variabili decisionali <= budget

viii. Capacità:
1. Variabili decisionali <= cap. minima
2. Variabili decisionali >= cap. massima

5) Scrivere i domini considerando che
   a. Al 90% le variabili sono intere
   b. Si mettono delle variabili reali qualora ci fossero frazioni/radici

Consigli ulteriori:

- se si sta scrivendo un vincolo ma non si riesce a capire come scriverlo, probabilmente può essere scritto in maniera più semplice, anche ripensando le stesse variabili. A me aiuta molto cercare di semplificare e considerare i punti detti.

Simplesso: Esercizi pratici

Si porta in forma standard, cioè

- Funzione obiettivo
  - Se è *max* diventa *min* e cambio tutti i segni; se è già *min* non si cambia nulla
- Vincoli di uguaglianza → Aggiungo le variabili di slack e la disuguaglianza diventa uguaglianza
  - Se si ha $\leq$ la variabile di slack viene aggiunta come positiva
  - Se si ha $\geq$ la variabile di slack viene aggiunta come negativa
- Variabili non negative (vincolo ≤ 0) → Aggiungo variabile positiva e cambio il segno a <u>tutte</u> le occorrenze della stessa variabile (il dominio è nella forma $\hat{x_i} = -x_i, \hat{x_i} \geq 0$)
- Termini noti non negativi → Se un termine noto è negativo, cambio segno a <u>tutta</u> la disuguaglianza

Si eseguono i calcoli, ricordandosi che si devono scrivere <u>alla fine</u> dei passaggi di pivoting per l'iterazione corrente di simplesso, le variabili in base (cioè, x_i nella riga corrispondente alla posizione che si sta attualmente calcolando). Si esegue il pivot rispetto alla colonna individuata nel tableau precedente (operazioni di somma/prodotto, etc.)

Imposto i tableau, ricordando che ogni volta mi chiedo:
- È in forma canonica rispetto alla base corrente? Basta avere le colonne (<u>anche sparse/non in ordine</u>) della matrice identità nel tableau rispetto agli elementi della base
  - Es. se la base è $[x_4, x_5, x_6]$ allora avrò gli 1 in corrispondenza di questi elementi in colonna

- È ammissibile? → Tutti i $\overline{b}$ devono essere positivi (se parto da una base ammissibile, rimango sempre in base ammissibile; mai dire che è impossibile)
- È ottima? → Devo avere tutti i costi ridotti ≥ 0 (in altri termini, tutte le variabili fuori base hanno costi ridotti >= 0)
- È illimitata? → Mi trovo ad avere un coefficiente nella riga R_0 dei costi ridotti con tutti i valori negativi sotto nella stessa colonna (in altri termini, tutte le variabili in base hanno costi ridotti <= 0)
- Entra in base? → La variabile costo ridotto negativo di indice minimo (Bland); si va per ordine posizionale di indice
    - Es. se devo scegliere tra x_2 e x_4 scelgo x_2 per ordine posizionale di indice
- Esce dalla base → La variabile che ha rapporto minimo prendendo in colonna la variabile entrante e facendo i rapporti minimi. Come prima, si prende la variabile di indice minimo posizionale (Bland)
    - Es. se entra in base x_2 prendo considero tutti i rapporti tra la variabile della colonna di $\overline{b}$ e gli elementi di questa, quindi $\arg\min$ su tutta la colonna di x_2 e gli elementi in riga corrispondenti
- Ricordarsi che si sostituisce una variabile entrante/uscente esattamente nella posizione da cui è stata tolta
    - Es. se da x_1, x_2, x_5 tolgo x_2 per far entrare es. x_3, l'ordine rimane x_1, x_3, x_5)

Ad ogni passaggio, eseguo il pivoting ogni volta prendendo in colonna la variabile che esce e in riga la variabile che entra.

Quando trovo la soluzione, considero che è tutta la colonna $\overline{b}$ in cui prendo il valore della funzione obiettivo (solitamente $-z$ e cambio segno per questo motivo); normalmente, il suo valore migliora (o non peggiora) tra i passaggi) e come vincoli dobbiamo fare delle precisazioni, riferite al problema di partenza (noi lo convertiamo in forma standard) ed esclusivamente alle variabili di slack:
- I vincoli laschi, nel momento in cui abbiamo una variabile di slack > 0
    - Si dice anche "soddisfatti col minore/maggiore stretto nella soluzione ottima"
- I vincoli saturi, nel momento in cui abbiamo una variabile di slack pari a 0 (uguale a 0)
    - Si dice anche "soddisfatti all'uguaglianza nella soluzione ottima"

Nella scrittura della soluzione, si deve considerare che nel tableau abbiamo $-z$, dunque va invertito il segno. Poi, si scrivono tutte le variabili (es. ho 7 variabili e scriverò $(x_1, x_2, x_3, x_4, x_5, x_6, x_7)$ con rispettivo valore (positivo se in base, quindi guardo quello che sta in $\overline{b_\iota}$, 0 altrimenti se fuori base sarà pari a 0, distinguendo opportunamente tra saturi e laschi).

Si può inoltre dire che "per verifica, i valori della funzione obiettivo e il modo di soddisfazione dei vincoli possono essere controllati sostituendo i valori delle variabili nella formulazione originaria".

Se si parte da un problema di massimo → la soluzione è $z_{MAX} = -z_{MIN}$. Esempio: soluzione del problema è $z = 5/2$. La soluzione considera $-z$, quindi sarà $-5/2$. Questa però è appunto $z_{MAX} = -z_{MIN} = -\left(-\frac{5}{2}\right) = \frac{5}{2}$

Info pratiche:

- Illimitato → Tutta colonna sotto un costo ridotto <=0
- Degenere → La f.o. mantiene stesso valore (non aumenta e rimane con valore uguale)

Domande teoriche sul simplesso

- Individuazione basi ottime senza operazioni di pivot
    - Basta vedere dove sono gli 1 della matrice identità e dove ci sono costi ridotti non negativi [quindi, sia uguali a 0 che maggiori] e scegliere come base quella che ha i coefficienti più a sinistra

- Scrivere la soluzione di base corrente e dire se è ottima.
    - Basta individuare il valore di z (ricordandosi che è $-z$) e scegliere come base la prima per i coefficienti sparsi della matrice identità (cioè, se ho più coefficienti tra i quali poter scegliere per capire la base, scelgo quello più a sinistra)

- Stabilire, SENZA EFFETTUARE LE OPERAZIONI DI PIVOT, quale sarà il valore della funzione obiettivo alla fine della prossima iterazione del simplesso. GIUSTIFICARE LA RISPOSTA!
    - Usiamo la regola di Bland per selezionare la variabile entrante ed eseguiamo il solito rapporto minimo tra $\overline{b_i}$ e il punto della variabile entrante
        - Se si ha un rapporto minimo pari a 0, la f.o. non migliora
        - Se si ha un rapporto minimo in cui il minimo elemento è positivo, basta eseguire il prodotto tra il coefficiente individuato dall'operazione di rapporto minimo e il costo ridotto della stessa colonna.
          $(coefficiente\ R_0\ riga\ costi\ ridotti * coefficiente\ rapporto\ minimo) = valore\ rapporto\ minimo$
        - Successivamente, si esegue una somma algebrica con il valore della funzione obiettivo (solitamente, si ha $-(valore\ f.o.) - (valore\ rapporto\ minimo)$
        - Il costo sarà negativo di solito della f.o.; è tutto normale, dato che vuol dire che migliora con la successiva iterazione (se non è degenere, naturalmente)

- Alla fine della prossima iterazione sarà cambiata la base corrente: sarà cambiato anche il vertice del poliedro associato alla nuova base? GIUSTIFICARE LA RISPOSTA!
    - Per calcoli, si ragiona che si effettua una sottrazione tra $\overline{b_i}$ e $\theta(valore\ riga\ x_i)$, per tutte le variabili in base. In base alla regola di Bland e rispetto a chi esce dalla base e letteralmente rifacciamo lo stesso calcolo rispetto alla variabile che entra in base, tenendo le altre. Capiamo così se l vertice è cambiato o meno.
    - Letteralmente, se ho due variabili con stesso rapporto minimo, allora so che l'iterazione sarà degenere e il vertice del poliedro non cambia valore
    - Se invece non ho due variabili con rapporto minimo, allora l'iterazione non sarà degenere e quindi il vertice del poliedro cambia valore; questo succede perché si ha una variabile a costo ridotto negativo che consente di migliorare la base corrente.

- Riusciamo ad individuare una soluzione di base corrispondente? Quale?
    - Il tableau è in forma canonica e individuo come soluzione di base una tra quelle in cui si hanno i coefficienti della matrice identità. Inoltre, occorre dire quanto vale z e quanto vale x (cioè, quanto valgono tutti i coefficienti dentro e fuori dalla base ammissibile)
    - Perché non è ottima?
        - Non sappiamo se sia ottima, avremo dei coefficienti di costo ridotto negativo (condizione sufficiente ma non necessaria per ottimalità)

  - È ottima?
    - Basterà avere tutti coefficienti di costo ridotto positivo; in altri termini, tutte le variabili fuori base hanno costi ridotti positivi

- Perché la teoria del simplesso non consente l'operazione di pivot sull'elemento X?
  - L'operazione non è consentita, in quanto quell'elemento non corrisponde al rapporto minimo
  - In altre parole, la variabile corrispondente alla colonna su cui si trova l'elemento riquadrato su cui non si può fare pivot, assumerebbe un valore tale da portare a 0 la corrispondente variabile alla riga su cui si trova l'elemento riquadrato su cui non si può fare pivot con un valore troppo alto per soddisfare i restanti vincoli e le altre variabili dovrebbero assumere valori negativi

- Su quali elementi è possibile effettuare il pivot secondo le regole del simplesso (indipendentemente da regole anticiclo)?
  - Prendendo come elementi la variabile che entra e la variabile che esce <u>a prescindere da Bland</u>, quindi letteralmente tutte le variabili che per riga/colonna corrispondono al rapporto minimo

- Considerando le variabili ordinate per indice crescente, quale sarà il cambio base secondo le regole del simplesso e applicando Bland?
  - Si considerano le regole dette (variabile che entra/variabile che esce secondo quanto scritto sopra) e poi prendo le variabili con indice posizionale minore (secondo Bland)

- (Non una domanda teorica, ma presente negli esercizi di calcoli del simplesso); in base a quale teorema è possibile determinare direttamente il valore della soluzione ottima?
  - Se il problema primale è illimitato
    - il problema duale è inammissibile (corollario del teorema della dualità debole)
  - Se il problema primale ammette soluzione ottima
    - Il corrispondente problema duale ha soluzione ottima finita e i valori coincidono (teorema della dualità forte)

- Supponiamo di effettuare un cambio base in cui entra in base la variabile X: perché la soluzione di base ottenuta a seguito di questo cambio base è sicuramente degenere?
  - In questo caso, avremmo X righe corrispondenti al rapporto minimo, pertanto avremo che almeno una esce dalla base con valore 0, mentre le altre Y assumeranno valore 0 rimanendo in base.

<u>In limitati casi</u> potrebbe essere che venga indicato di risolvere il problema di PL partendo da una base.

Quello che si fa è:

- Portare il problema in forma standard
- Considerare una delle due basi organizzando i dati in forma tableau tralasciando la f.o.
- Eseguire le operazione di pivot rispetto al primo elemento della base
  - Ad esempio, se devo scrivere la cosa rispetto alla base $[x_1, x_5, x_6]$ considero come elemento di pivot x_1
  - Ad esempio, se devo scrivere la cosa rispetto alla base $[x_4, x_5, x_6]$ considero come elemento di pivot x_4

- Se riesco a seguito di tutto il pivot ad ottenere una soluzione ammissibile, posso applicare il metodo del simplesso e risolvere.

<u>Allego esempio grafico di quanto appena detto per chiarezza.</u>

Consideriamo la base $B = \{x_4, x_5, x_6\}$ e verifichiamo se è una base ammissibile. Organizziamo i dati in forma tableau tralasciando la funzione obiettivo.

$$\begin{array}{c|cccccc|c|} & x_1 & \hat{x}_2 & x_3 & x_4 & x_5 & x_6 & \bar{b} \\ \hline x_4 & 2 & -1 & -2 & \boxed{-1} & 0 & 0 & 1 \\ x_5 & 1 & 0 & 2 & 0 & 1 & 0 & 2 \\ x_6 & -1 & 2 & -1 & 0 & 0 & 1 & 3 \\ \hline \end{array}$$

$$\begin{array}{c|cccccc|c|} & x_1 & \hat{x}_2 & x_3 & x_4 & x_5 & x_6 & \bar{b} \\ \hline x_4 & -2 & 1 & 2 & 1 & 0 & 0 & -1 \\ x_5 & 1 & 0 & 2 & 0 & 1 & 0 & 2 \\ x_6 & -1 & 2 & -1 & 0 & 0 & 1 & 3 \\ \hline \end{array}$$

La soluzione associata alla base $B = \{x_4, x_5, x_6\}$ è $x = (x_1, \hat{x}_2, x_3, x_4, x_5, x_6) = (0, 0, 0, -1, 2, 3)$. Osserviamo che $x_4 = -1 < 0$, quindi la base B non è una base ammissibile e il metodo del simplesso non può, dunque, essere applicato a questa base.

Condizioni primale/duale e passaggio al problema duale

<u>(se problema di massimo va letta a rovescio)</u>

Primale ($\min c^T x$)	Duale ($\max u^T b$)
$a_i^T x \geq b_i$	$u_i \geq 0$
$a_i^T x \leq b_i$	$u_i \leq 0$
$a_i^T x = b_i$	u_i libera
$x_j \geq 0$	$u^T A_j \leq c_j$
$x_j \leq 0$	$u^T A_j \geq c_j$
x_j libera	$u^T A_j = c_j$

Esempio 1

$u^T A \sim c^T$ $\quad u^T A_1 \quad c_1$

$u^T A_2 \sim c_2$

$$\begin{array}{llllll} \min & 10x_1 & +20x_2 & +0 \cdot x_3 & & \\ s.t. & 2x_1 & -x_2 & & \geq 1 & u_1 \\ & & x_2 & +x_3 & \leq 2 & u_2 \\ & x_1 & & -2x_3 & = 3 & u_3 \\ & & 3x_2 & -x_3 & \geq 4 & u_4 \\ \hline & x_1 & & & \geq 0 & \\ & & x_2 & & \leq 0 & \\ & & & x_3 & \text{libera} & \end{array}$$

Primale ($\min c^T x$)	Duale ($\max u^T b$)
$a_i^T x \geq b_i$	$u_i \geq 0$
$a_i^T x \leq b_i$	$u_i \leq 0$
$a_i^T x = b_i$	u_i libera
$x_j \geq 0$	$u^T A_j \leq c_j$
$x_j \leq 0$	$u^T A_j \geq c_j$
x_j libera	$u^T A_j = c_j$

$$\max \; 1 \cdot u_1 + 2 \cdot u_2 \cdot 3u_3 \; 4 \cdot u_4$$
$$2 \cdot u_1 + 0 \cdot u_2 + 1 \cdot u_3 + 0 \cdot u_4 \leq 10$$
$$-u_1 + u_2 + 3u_4 \geq 20$$
$$+u_2 - 2u_3 - u_4 = 0$$
$$u_1 \geq 0 \quad u_2 \leq 0 \quad u_3 \text{ libera} \quad u_4 \geq 0$$

Enunciare le condizioni di complementarietà primale-duale

<u>Enunciato</u> delle condizioni di complementarietà primale duale:

Dati un problema primale $\min c^T x$ s.t. $Ax \geq b, x \in \mathbb{R}^n_+$ e il corrispondente duale $\max u^T b$ s.t. $u^T A \leq c, u \in \mathbb{R}^m_+$, e due vettori $\bar{x} \in \mathbb{R}^n_+$ e $\bar{u} \in \mathbb{R}^m_+$, $\bar{x}$ e $\bar{u}$ sono soluzioni ottime rispettivamente per il primale e per il duale se e solo se: $\bar{x}$ è ammissibile primale, $\bar{u}$ è ammissibile duale, $u_i(a_i^T x - b_i) = 0, \forall i = 1 \dots m$, e $(c_j - u^T A_j)x_j = 0, \forall j = 1 \dots n$, dove a_i^T è la riga i-esima di A e A_j è la colonna j-esima di A.

Dualità

Abbiamo il problema e la verifica di una certa soluzione. Quindi:

1) Verifica ammissibilità primale dela soluzione data → Sostituisco i valori della soluzione che mi viene data dentro i vincoli e verifico se sono rispettate tutte le disuguaglianze

2) Passaggio al problema duale
   a. Se ho problema di min passo a problema di max (o viceversa)
   b. Considero come termini u_i tutti i termini noti (compresi i loro coefficienti)

c. Per ogni vincolo, prendo tutti le variabili con coefficienti alla colonna i corrispondente alla posizione u_i, e:
    i. Quando passo da min a max riporto l'opposto del segno della corrispondente variabile di dominio primale
    ii. Quando passo da max a min, riporto lo stesso segno della corrispondente variabile di dominio primale
    iii. se non c'è nulla per la variabile x_i quando si scrive il vincolo duale, si vede come $= 0$ (questo anche per quando si deve trovare l'opposto tra min-max o max-min; l'opposto di una uguaglianza è sempre una uguaglianza)

d. Si inseriscono i domini delle variabili, considerando che:
    i. Se passo da problema di min a problema di max, il dominio delle duali corrisponde allo stesso segno di uguaglianze/disuguaglianze delle variabili primali
    ii. Se passo da problema di max a problema di min, il dominio delle duali corrisponde all'opposto del segno di uguaglianze/disuguaglianze delle variabili primali

3) CCPD - Applicazione delle condizioni di complementarietà primale-duale
    a. Primo pezzo: vincoli primali
        i. Prendo $u_i\ con\ "i"$ posizione della riga attuale e lo moltiplico per il vincolo primale in posizione i portando a sinistra il termine noto (cambiando quindi di segno)
        Es. $-x_1 - x_2 + 2x_3 + x_4 \geq 1 \Rightarrow u_1(-x_1 - x_2 + 2x_3 + x_4 - 1) = 0$
        ii. Sostituisco quindi i valori della soluzione iniziale data nel vincolo e:
            1. Se ho un valore > 0 , allora questo viene considerato come condizione
            2. Se ho un valore $= 0$, allora non mi dice nulla e non lo considero (*non posso dedurre condizioni*)
        iii. Se ho vincoli di uguaglianza non posso dire nulla (*deriva dall'ammissibilità primale e non posso dedurre condizioni di complementarietà per la variabile x_i) → no CCPD*
    b. Secondo pezzo: vincoli duali
        i. Prendo $x_i\ con\ "i"$ posizione della riga attuale e lo moltiplico per il vincolo duale in posizione i portando a sinistra il termine noto (cambiando quindi di segno)
        Es. $-u_1 - u_2 \leq 2 \Rightarrow (-u_1 - u_2 - 2)x_1$
        ii. Sostituisco il valore di x_i in quella posizione e faccio le stesse verifiche dei sottocasi (1) e (2) della seconda condizione del problema primale
        iii. Se ho vincoli di uguaglianza, devo verificare che non faccia già parte dei vincoli; nel qual caso lo considero (*deriva dall'ammissibilità duale*), altrimenti no

4) Sistema di equazioni CCPD e ammissibilità duale - Sistema di equazioni per l'imposizione delle condizioni di complementarietà primale duale (ccpd) trovate e delle condizioni di ammissibilità duale:
    a. Metto insieme tutte le condizioni che ho trovato fino ad ora e vado a sostituire i valori di u_i oppure di x_i che ho già dal sistema di equazioni e risolvo trovando tutte le soluzioni

5) Verifica ammissibilità duale
    a. Verifico se tutti i valori di u_i soddisfano i vincoli duali (cioè, soddisfano le disuguaglianze)
    b. Verifico se tutti i valori di u_i soddisfano il dominio duale (es. $u_2 = 2$ soddisfa $u_2 \geq 0$)

6) Conclusioni
    a. (Happy Ending)
        i. x è ammissibile primale (come da verifica)
        ii. u è ammissibile duale (come da costruzione e da verifica)
        iii. x, u sono in scarti complementari
        iv. Le due soluzioni sono ottime per i rispettivi problemi primale e duale
        v. Per verifica si confrontino i valori delle f.o. dei problemi primale e duale; saranno uguali per il teorema della dualità forte
    b. (Bad Ending) → Si noti che il problema chiede "dimostrare se è ottima", cosa che negli esercizi d'esame non capita mai
        i. La soluzione trovata è l'unica soluzione del sistema di cui al punto 4 e, quindi, l'unica che soddisfa i vincoli duali di uguaglianza e che è in scarti complementari con la soluzione primale data. Tale soluzione però non è ammissibile per il problema duale. Pertanto non è possibile trovare nessuna soluzione ammissibile duale che sia in scarti complementari con la soluzione primale data, che, quindi, non è ottima

Branch and Bound

Minimo → [LB; S.A.]

Massimo → [S.A.; UB]

1) Individuare se si tratta di problema di minimo o di massimo

- Se si tratta di problema di minimo i LB aumentano (o non decrescono) di padre in figlio
    - In questo caso avremo come struttura [LB; S.A]
- Se si tratta di problema di massimo gli UB decrescono (o non crescono) di padre in figlio
    - In questo caso avremo come struttura [S.A.; UB]

2) Individuare nodi da poter chiudere

In generale, non chiudo i nodi che sono già sviluppati (quindi normalmente, il nodo radice e nodi che hanno dei figli) e poi:

- Se si tratta di problema di minimo
    - Chiudo i nodi che hanno un LB >= S.A (tengo i nodi con LB < S.A)
- Se si tratta di problema di massimo
    - Chiudo i nodi che hanno un UB <= S.A (tengo i nodi con UB > S.A.)

3) Intervallo ottimo / Intervallo in cui è sicuramente compreso il valore della f.o. / Miglior valore per una soluzione ammissibile

- Se si tratta di problema di minimo
    - Considero il miglior UB (minimo) tra tutti i nodi (attuale soluzione ammissibile/incumbent) e come LB il minore tra i nodi aperti (quindi, non P_0, P_1, P_2)
- Se si tratta di problema di massimo
    - Considero il miglior LB (massimo) tra tutti i nodi (attuale soluzione ammissibile/incumbent) e come UB il maggiore tra i nodi aperti (quindi, non P_0, P_1, P_2)
- Se si parla di miglior valore come soluzione ammissibile → Si cerca minimo-massimo tra tutti i possibili nodi
- Se si parla di miglior valore come valore ottimo → Si cerca minimo-massimo tra i soli nodi aperti

4) Quale sarà il nodo sviluppato per primo in una strategia Best Bound First?

- Se si tratta di problema di minimo
  - Si sceglie il nodo con il miglior LB (quello minimo) tra i nodi aperti
- Se si tratta di problema di massimo
  - Si sceglie il nodo con il miglior UB (quello massimo) tra i nodi aperti

5) Si supponga che lo sviluppo di cui al punto precedente porti a due nodi figli, di cui uno è relativo ad un insieme di soluzioni vuoto. Si dia un esempio di valori di LB e UB relativi al secondo nodo, che consentano di riconoscere subito la soluzione ottima del problema.

- Chiamiamo il nodo aperto per esempio P_7; la selezione viene fatta solo nei nodi tuttora aperti, compreso P_7. Si deve considerare P_7 come figlio del nodo best bound first e P_8 che porta ad una soluzione non ammissibile (può essere anche P_8 e P_7, la sostanza è avere due nodi).
  - Se per un problema di minimo, dobbiamo prendere un LB che rispetti la proprietà padre-figlio (quindi >= LB del nodo padre), mentre prendo come UB una nuova incumbent, cioè un valore che sia <= a tutti i LB presenti
  - Se per un problema di massimo, Il LB deve essere una nuova incumbent, dunque >= dei nodi che si vogliono chiudere (quindi, maggiore al loro UB), mentre l'UB deve essere compatibile con il fatto di essere figlio del nodo best-bound first, quindi essere <= UB del nodo padre
- Il valore deve essere tale da permettere la chiusura anche del nodo $P_7\ oppure\ P_8$, quindi possibilmente dentro l'intervallo UB/LB individuato
- Normalmente, si può avere lo stesso valore per UB e LB per semplicità (cosa comune)

6) Individuare possibili valori per UB per mantenere la coerenza con problema di massimo (vuol dire che si avrà "UB?" sul testo)

- Se si tratta di problema di massimo, gli UB decrescono (o non crescono) di padre in figlio; quindi, l'UB dovrà essere compreso tra l'UB del nodo padre (come estremo superiore) e l'UB massimo tra i nodi figli
- $UB \in [UB\ massimo\ nodi\ figli, UB\ nodo\ padre]$

7) Individuare possibili valori per LB per mantenere la coerenza con problema di minimo (vuol dire che si avrà "LB?" sul testo)

- Se si tratta di problema di minimo, i LB crescono (o non decrescono) di padre in figlio; quindi, il LB dovrà essere compreso tra il LB del nodo padre (come estremo inferiore) e il LB minimo tra i nodi figli
- $LB \in [LB\ nodo\ padre, LB\ minimo\ nodi\ figli]$

Grafi

Domande:

- Si scelga l'algoritmo da utilizzare e si motivi la scelta:
  - Quando si ha un massimo numero di archi/hop (anche se i costi sono tutti positivi) → Bellman-Ford
    - Posso applicare solo l'algoritmo di Bellman-Ford che è l'unico che dia la possibilità di calcolare i cammini minimi con il massimo numero di archi. Infatti, è possibile dimostrare che, all'iterazione k dell'algoritmo, le etichette corrispondono ai cammini minimi che utilizzano al più k archi. Applicheremo quindi Bellman-Ford fermandoci alla k-esima iterazione, dopo l'inizializzazione.

- Quando si hanno costi ridotti negativi → Bellman-Ford
    - Si sceglie l'algoritmo di Bellman-Ford in quanto esistono archi con costo negativo. Bellman-Ford è l'unico algoritmo visto in grado di garantire convergenza alla soluzione ottima del problema dei cammini minimi in presenza di archi di costo negativo, sebbene mediamente meno efficiente dell'algoritmo di Dijkstra, che non garantisce di trovare la soluzione al problema dei cammini minimi se esistono archi di costo negativo.
- Se costi ridotti tutti positivi e non ci sono max hop → Dijkstra, perché più efficiente computazionalmente
    - Essendo tutti i costi sugli archi non negativi, possiamo applicare l'algoritmo di Dijkstra, il più efficiente tra quelli visti

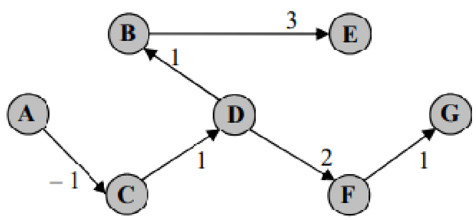
Albero dei cammini minimi:

Archi corrispondenti ai predecessori dell'ultima riga; in altre parole, guardo l'iterazione rispetto all'ultima volta in cui un particolare nodo è stato aggiornato, prendendone i predecessori.

Punto b)

Iter.	A	B	C	D	E	F	G	*Aggiornati*
h=0	0(-)	+∞(-)	+∞(-)	+∞(-)	+∞(-)	+∞(-)	+∞(-)	A
h=1	0(-)	5(A)	–1(A)	3(A)	+∞(-)	+∞ (-)	+∞(-)	B, C, D
h=2	0(-)	4(D)	–1(A)	0(C)	8(B)	3(C)	+∞(-)	B, D, E, F
h=3	0(-)	1(D)	–1(A)	0(C)	~~7(B)~~ 5(F)	2(D)	4(F)	B, E, F, G
h=4	0(-)	1(D)	–1(A)	0(C)	4(B)	2(D)	3(F)	E, G
h=5	0(-)	1(D)	–1(A)	0(C)	4(B)	2(D)	3(F)	//

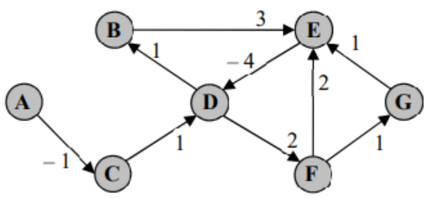
L'albero dei cammini minimi si ottiene riportando gli archi corrispondenti ai predecessori letti nell'ultima riga della tabella:



Grafo dei cammini minimi:

Tutti i cammini minimi; questo vuol dire almeno tutti gli archi dell'albero più tutti quelli ammissibili a costo <= rispetto all'albero (etichette ottime)

Il grafo dei camminimi minimi si ottiene completando con tutti gli archi che soddisfano all'uguaglianza il vincolo duale $\pi_j = \pi_i + c_{ij}$:



- Si riporti un cammino minimo con al più X archi da Y a Z

Si costruisce la catena dei predecessori e si verifica la costruzione del cammino; normalmente, questo viene fatto

- È possibile, basandosi solo sulla tabella precedente e senza fare calcoli, determinare l'esistenza di un ciclo negativo?

Basandosi solo sulla tabella e senza ulteriori calcoli, è possibile determinare il ciclo negativo perché siamo arrivati fino in fondo ai calcoli con etichette instabili (quindi, con nodi appena aggiornati o su cui si ha avuto almeno un aggiornamento).

Note fattuali:

- Quante iterazioni fare? Tante quante il numero di nodi.
- Non possiamo parlare di albero dei cammini minimi qualora si parla di massimo numero di archi; possiamo invece individuare dei cammini minimi (nel senso che potremmo avere cammini a costo inferiore che però hanno più archi e questo non vale).
- Se la lista dei nodi aggiornati è vuota, le etichette sono stabili e pertanto è possibile determinare albero o grafo dei cammini minimi; altrimenti no.
- Riportare i passi dell'algoritmo scelto in una tabella e giustificarne i passi.

Bellman-Ford

Indicazioni teoriche (da scrivere in esame)

Si utilizza una tabella che riporta una riga per ogni iterazione dell'algoritmo. Ogni colonna della tabella è dedicata ad un nodo e riporta, iterazione dopo iterazione, l'evoluzione delle rispettive etichette. L'ultima colonna riporta i nodi aggiornati nel corso dell'iterazione: all'iterazione successiva è sufficiente controllare solo gli archi uscenti da questi nodi (vincolo di Bellman).

La tabella riporta al riga 0 di inizializzazione e una riga per ogni iterazione. All'iterazione h si controllano gli archi (i, j) uscenti da ciascun nodo i nella colonna $Aggiornati$ alla riga $h-1$, e si aggiornano i costi e i predecessori del nodo j all'iterazione h qualora l'etichetta del nodo i all'iterazione $h-1$ più il costo dell'arco (i, j) sia strettamente minore dell'etichetta corrente del nodo j.

L'algoritmo si ferma qualora la lista dei nodi aggiornati sia vuota (convergenza delle etichette ai costi dei cammini minimi da A verso gli altri nodi).

Indicazioni pratiche (per capire le cose)

Nell'individuazione dei cammini minimi, ricostruisco la catena dei predecessori:

- Partendo solo nel disegno dall'ultimo nodo e poi vaglio tutte le alternativa prendendo gli ultimi nodi aggiornati e, tramite i predecessori, costruire un percorso minimo fino al nodo origine
  - Se ricostruendo i cammini minimi, torno su un arco già trovato in precedenza, in quel caso ho un ciclo negativo

Esempio di tabella e cosa scrivere per BF:

Iterazione	Nodo A	Nodo B	Nodo C	Nodo D	Nodo E	Nodo F	Aggiornamenti
$Inizio$	$0_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	A
$h=1$	$0_\wedge$	3_A	2_A	$+\infty_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	B, C
$h=2$	$0_\wedge$	3_A	2_A	9_B	4_C	9_C	D, E, F
$h=3$	$0_\wedge$	3_A	2_A	5_E	4_C	6_E	D, F
$h=4$	$0_\wedge$	3_A	2_A	5_E	4_C	6_E	F

Dijkstra

Indicazioni teoriche (da scrivere in esame)

Legenda della tabella (la scrivo io per chiarezza di contenuto questa, ndr; occorre giustificare i passi):

- $\hat{v}$ rappresenta l'etichetta minima di ogni iterazione
- $\bar{S}$ rappresentano le etichette ancora da fissare
- Il segno * rappresenta l'etichetta fissata
- Il segno – rappresenta l'etichetta controllata ma non aggiornata (normalmente ha stesso costo rispetto a quello che attualmente ha e quindi non vado ad aggiornare)
- Il segno x rappresenta l'etichetta non controllata perché il nodo è già fissato

- Gli spazi vuoti servono per indicare che non considero più l'etichetta nelle varie iterazioni in quanto fissata
- L'algoritmo termina quando non ci sono più nodi in $\bar{S}$

Iterazione	Nodo A	Nodo B	Nodo C	Nodo D	Nodo E	Nodo F	$\bar{S}$	$\hat{v}$
$Inizio$	$0_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	A, B, C, D, E, F	
$h = 1$	*	3_A	2_A	$+\infty_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	B, C, D, E, F	A
$h = 2$		-	*	9_B	4_C	9_C	B, D, E, F	C
$h = 3$		*		5_E	-	6_E	D, E, F	B
$h = 4$				-	*	-	D, F	E
$h = 5$				*		-	F	D
$h = 6$						*	$\emptyset$	F

Ad ogni iterazione, percorriamo il grafo e scegliamo il percorso con costo minore. Ad ogni iterazioni, scegliamo e fissiamo un'etichetta che ha costo minore, aggiungendo un'etichetta all'insieme di quelle fissate e controllando quanto accade per le altre (segnando etichette controllate ma non aggiornate oppure non controllate perché il nodo è già fissato).

Si procede inoltre con le verifiche della condizione di Bellman (vincolo duale) sugli archi che escono dall'etichetta minima e portano nodi nell'insieme dei nodi da fissare. Le etichette calcolate e i relativi puntatori rappresentano, rispettivamente, una soluzione duale ammissibile e i predecessori su dei cammini dall'origine ai diversi nodi.

L'algoritmo termina quando tutte le etichette sono state fissate, cioè quando tutti i nodi di $\bar{S}$ vengono trasferiti in S e quando tutte le etichette sono state correttamente fissate, quindi $\bar{S} = \emptyset$

AMPL

Non appaiono quasi mai all'esame e normalmente si tratta di tradurre le sommatorie in un file .mod; al massimo, può capitare di dover scrivere un file .dat e, in quel caso, anche il file .run.

Si segua questa struttura logica.

File.mod:

- Avere un *set* per indicare gli insiemi
- Avere dei *param* indicizzati con le graffe degli insiemi → es. $O_i\ sarà\ param\ O\{I\}$;
- Le variabili intere sono >=0 *integer*
- Le variabili reali sono >=0 e basta
- Le variabili comprese tra 0 ed 1 sono *binary*
- La f.o. viene scritta con *maximize/minimize* e poi nella forma $due\ punti - indici\ di\ riferimento$
- Normalmente, i vincoli sono con *s.t.* e:
  - L'indice più esterno è quello per cui viene scritto il vincolo
    - Es. Indice più esterno è j e allora $s.t.\ \ v1\{j\ in\ J\}: \{sum\ i\ in\ I\}$
- Attenzione a mettere (costante * variabile)
  - Cioè scrivere $2 * x_1\ e\ non\ 2x_1$

File .dat:
- Si descrivono gli insiemi con := su una sola linea
- Si descrivono eventuali parametri costanti con := in linea
- I param vanno con ≔ se in una sola linea, altrimenti
  - Se in forma matriciale (es $C\{I,J\}$ diventa con I in colonna e J in riga; la forma di scrittura è $$nome\ parametro, :=, J\ in\ riga\ e\ :=, poi\ mettendo\ I\ in\ riga$$
- Si aggiunge (tr) per mettere I in riga e J in colonna
- Ricordarsi di mettere i punti e virgola da tutte le parti, specie dopo le variabili matriciali

File .mod:
- reset
- model file.mod
- data file.dat
- option solver *unoqualunque*
- solve
- display funzioneobiettivo, variabile1, variabile2, etc.

Esempio notevole e di studio:

6. Si vuole risolvere con **AMPL** un problema di trasporto di alberi da un insieme di origini I a un insieme di destinazioni J. Ciascuna origine i mette a disposizione O_i alberi e ciascuna destinazione richiede D_j alberi. Il costo unitario di trasporto da i a j è C_{ij} e si ha un costo fisso F_i per l'organizzazione dei trasporti da ciascuna origine i. Non è inoltre possibile organizzare il trasporto in più di N origini. Il modello per la minimizzazione dei costi è riportato affianco e utilizza le variabili x_{ij} per indicare il numero di alberi trasportati da i a j, e y_i che vale 1 se si organizza il trasporto da i, 0 altrimenti.

$$\min \sum_{i \in I, j \in J} C_{ij}\, x_{ij} - \sum_{i \in I} F_i\, y_i$$

$$\text{s.t.} \sum_{i \in I} x_{ij} \geq D_j \quad , \quad \forall j \in J$$

$$\sum_{j \in J} x_{ij} \leq O_i\, y_i \quad , \quad \forall i \in I$$

$$\sum_{i \in I} y_i \leq N$$

$$x_{ij} \in \mathbb{Z}_+, \;\; y_i \in \{0,1\}, \;\; \forall\, i \in I, \;\; j \in J$$

a. Si traduca nel linguaggio **AMPL** il modello proposto (**file .mod**).

b. Si produca il **file .dat** per l'istanza con origini Croazia, Svezia, Gran Bretagna e Canada (disponibilità di 1000, 2000, 3000 e 4000 alberi rispettivamente), destinazioni Italia, Francia e Germania (con richieste di 5000, 3000 e 2000 rispettivamente), $N = 3$, costi fissi F_i di 1000 euro per tutte le origini, e costi di trasporto verso Italia, Francia e Germania (nell'ordine) pari a: dalla Croazia 10, 20 e 30 euro; dalla Svezia 40, 50 e 60 euro; dalla Gran Bretagna 70, 80 e 90 euro; dal Canada 100, 110 e 120 euro.

c. Si scriva uno script di AMPL (**file .run**) che risolve l'istanza specificata e visualizza il valore della funzione obiettivo e delle variabili per una soluzione ottima.

Punto a)

```
set I;              set J;
param O{I};         param D{J};
param C{I,J};       param F{I};
param N;
var x{I,J} >=0 integer;
var y{I} binary;
minimize fo: sum{i in I, j in J} C[i,j]*x[i,j] - sum{i in I} F[i]*y[i];
s.t. d{j in J}: sum{i in I} x[i,j] >= D[j];
s.t. o{i in I}: sum{j in J} x[i,j] <= O[i] * y[i];
s.t. n: sum{i in I} y[i] <= N;
```

Punto b)

```
set I := Croazia Svezia GranBretagna Canada;
set J := Italia Francia Germania;
param :        F     O :=
Croazia        1000  1000
Svezia         1000  2000
GranBretagna   1000  3000
Canada         1000  4000;
param D := Italia 5000 Francia 3000 Germania 2000;
param N := 4;
param C :      Italia     Francia     Germania :=
Croazia        10         20          30
Svezia         40         50          60
GranBretagna   70         80          90
Canada         100        110         120;
```

Punto c)

```
reset;
model ampl.mod;
data ampl.dat;
option solver cplexamp;
solve;
display fo, x, y;
```