Programma del corso di “Ricerca operativa” (6 c.f.u.) 2026/2027

Programma del corso:

- Introduzione: programmazione matematica, programmazione lineare.
- Modelli: modelli di programmazione lineare (intera).
- Cenni sulla Programmazione Lineare: geometria della programmazione lineare (vertici e soluzioni base), metodo del simplesso; dualità in programmazione lineare: problema duale, proprietà fondamentali, interpretazione economica.
- Cenni sulla Programmazione Lineare Intera: unimodularità, metodo del branch and bound.
- Alcuni problemi specifici con metodi di soluzione specifici:

--- Problema del cammino di costo minimo: algoritmo di Dijkstra.

--- Problema della pianificazione di progetti: metodo PERT.

--- Problema del massimo flusso: proprietà fondamentali, algoritmo di Ford-Fulkerson.

--- Problema della programmazione della produzione: metodo di Wagner-Whitin.

--- Problema di localizzazione di impianti: algoritmi di ricerca locale.

Libri di riferimento consigliati:

- R. Baldacci, M. Dell’Amico, *Fondamenti di Ricerca Operativa*, Pitagora Ed. Bologna (2002).
- M. Fischetti, *Lezioni di Ricerca Operativa*, Ed. Libreria Progetto Padova (1999, 2014)
- S. Martello, M.G. Speranza, *Ricerca Operativa per l’Economia e per l’Impresa*, Società Editrice Esculapio (2012).
- A. Sassano, *Modelli e Algoritmi della Ricerca Operativa*, Ed. Franco Angeli (1999).

Materiale scritto: il materiale scritto per l’esame è dato dagli “appunti” riportati di seguito.

Modalità di svolgimento della prova d’esame: prova scritta, prova orale facoltativa.

Orario ricevimento: venerdì, dalle 12 alle 14 (email: r.mosca@unich.it)

Appunti per il corso di “Ricerca operativa”   2026/2027

Questi appunti sono tratti liberamente dai seguenti libri, rispettivamente, del Prof. Matteo Fischetti e del Prof. Antonio Sassano:

[Fischetti]: M. Fischetti, *Lezioni di Ricerca Operativa*, Ed. Libreria Progetto Padova (2014).
[Sassano]: A. Sassano, *Modelli e Algoritmi della Ricerca Operativa*, Ed. Franco Angeli (1999).

1. Introduzione

La Ricerca Operativa è una disciplina che – in una frase – si occupa di modellare e risolvere problemi di ottimizzazione. Tale disciplina è relativamente recente, formalizzata nel corso della Seconda Guerra Mondiale, pur essendo già stata essenzialmente introdotta.

Una possibile descrizione di una possibile utilità della Ricerca Operativa è lo schema:

problema di ottimizzazione (informale) $\rightarrow$ modello matematico (formale) $\rightarrow$ risoluzione

che sintetizza quanto segue:

• primo passaggio: si parte da un dato problema di ottimizzazione (in termini informali) e si arriva possibilmente a un suo modello matematico (in termini formali); tale passaggio si ottiene grazie all'abilità e all'esperienza umana;

• secondo passaggio: si parte dal modello matematico di sopra e si arriva possibilmente alla sua risoluzione cioè al calcolo di una "soluzione ottima" del problema di ottimizzazione dato; tale passaggio si ottiene grazie a software specifici.

In particolare: per *problema di ottimizzazione* si intende un problema in cui un decisore deve scegliere, in un certo insieme di decisioni/soluzioni ammissibili, una decisione/soluzione che ottimizza una certa utilità; per *modello matematico*, in accordo con il libro [Fischetti], si intende un problema di Programmazione Matematica.

Un problema di ***Programmazione Matematica*** consiste nel calcolare

$$\mathbf{min}\ \{ f(x) : x \in X\}$$

dove:

$X \subseteq \mathbb{R}^n$ ($n \in \mathbb{N}$) un insieme detto *insieme delle soluzioni ammissibili* e

$f : X \rightarrow \mathbb{R}$ è una funzione detta *funzione obiettivo*.

Nota: il **min** può essere sostituito con il **max** cambiando il segno della funzione.

Risolvere un problema di Programmazione Matematica significa calcolare una *soluzione ottima* del problema cioè una soluzione $x^* \in X$ tale che $f(x^*) \leq f(x)$ per ogni $x \in X$.

Un problema di Programmazione Matematica è detto di ***Programmazione Lineare*** se:
X è l'insieme delle soluzioni di un sistema di equazioni e/o disequazioni lineari;
f è una funzione lineare.

Esempio:

$$\begin{array}{ll} \max & 12x_1 + 7x_2 \\ & 3\,x_1 + 2\,x_2 \leq 20 \\ & 5\,x_1 + 3\,x_2 \leq 50 \\ & x_1, x_2 \geq 0 \end{array}$$

Allora in questo esempio: $n = 2$, il che significa che ci sono 2 variabili (reali), cioè $X \subseteq \mathbb{R}^2$; in particolare, $X = \{\ (x_1, x_2) \in \mathbb{R}^2 : 3\,x_1 + 2\,x_2 \leq 20\ ;\ 5\,x_1 + 3\,x_2 \leq 50\ ;\ x_1 \geq 0\ ;\ x_2 \geq 0\ \}$, e $f : X \rightarrow \mathbb{R}$ è definita come $f(x_1, x_2) = 12x_1 + 7x_2$, cioè associa a ogni $(x_1, x_2) \in X$ il valore $12x_1 + 7x_2$.

Un problema di Programmazione Matematica è detto di ***Programmazione Lineare Intera*** se:
X è l'insieme delle soluzioni "intere" di un sistema di equazioni e/o disequazioni lineari;
f è una funzione lineare.

Esempio:

$$\begin{array}{ll} \max & 12x_1 + 7x_2 \\ & 3\,x_1 + 2\,x_2 \leq 20 \\ & 5\,x_1 + 3\,x_2 \leq 50 \\ & x_1, x_2 \geq 0 \\ & x_1, x_2 \ \underline{\text{interi}} \end{array}$$

In altri termini un problema di Programmazione Lineare Intera può essere visto come una variante di un problema di Programmazione Lineare attraverso l'aggiunta di "vincoli di interezza" per le variabili – oppure solo per alcune variabili nel qual caso si parla di Programmazione Lineare Intera Mista.

La Programmazione Lineare (Intera), pur costituendo un aspetto basilare/parziale della Programmazione Matematica, risulta utile per modellare una larga fascia di problemi di ottimizzazione di natura reale.

Così concludiamo questa introduzione riportando di seguito alcuni problemi di ottimizzazione di natura reale, che possono – come vedremo poi in dettaglio – essere modellati come problemi di Programmazione Lineare (Intera) tramite il primo passaggio dello schema, e che possono poi essere risolti tramite il secondo passaggio dello schema. Nel leggerli potremo osservare che nella vita reale noi comunque diamo una risposta a tali problemi, con criteri di buon senso, comunque determinando soluzioni ammissibili che sono presumibilmente buone anche se non necessariamente ottime; allo stesso tempo potremo osservare che tali problemi, nella loro essenza, possono presentarsi in altri contesti per i quali invece determinare una soluzione ottima (o comunque migliore possibile) può essere rilevante.

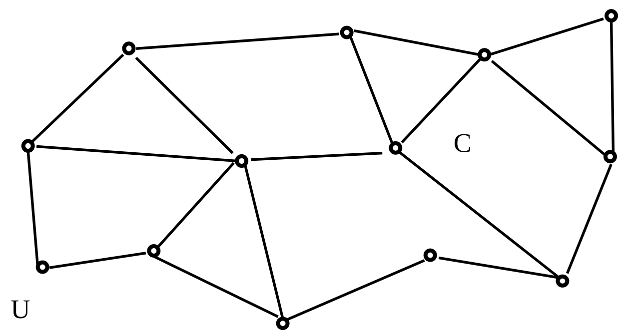
Una premessa tecnica per completezza.
Ricordiamo che un *grafo*, sia $G = (V, E)$, è una coppia di insiemi, l'insieme V detto dei *vertici* e l'insieme E detto degli *archi*, con $E \subseteq V \times V$ (cioè E è un sottoinsieme di coppie di elementi di V). Un grafo è in genere disegnato indicando: ogni vertice i con un punto; ogni possibile arco, cioè ogni possibile coppia (i, j) di vertici, con una linea fra il punto i e il punto j. Infine c'è una distinzione fra *grafo orientato*, in cui ogni arco è orientato (tipo senso unico), e *grafo non-orientato*, in cui ogni arco non è orientato (tipo doppio senso); se nulla è specificato, allora in genere il grafo è non-orientato. □

• Crediti formativi (c.f.u.)
Una persona ha a disposizione, nel prossimo semestre, un numero H di ore per preparare alcuni esami. Egli può preparare esami che fanno parte di un insieme $E = \{1, \dots, n\}$ di esami. Ogni esame i, per $i = 1, \dots, n$, per la preparazione necessita un tempo stimato h_i. D'altro canto ogni esame i, per $i = 1, \dots, n$, garantisce un credito formativo c_i. La persona decide di non preparare parzialmente un esame, cioè, decide o di prepararlo bene (impiegando tutto il tempo stimato necessario) o per niente. Il problema è scegliere quali esami preparare, con il vincolo di tempo esposto sopra, in modo da massimizzare il totale dei crediti degli esami scelti.

Commento: un problema del genere – come vedremo – può essere modellato definendo, per ogni esame i, una variabile decisionale "binaria" x_i che avrà valore 1 se l'esame i è scelto e avrà valore 0 altrimenti.

• Il percorso migliore
Una persona ogni mattina si reca da casa all'università. Il problema è scegliere un percorso da casa all'università (ce ne può essere più di uno) che sia di tempo minimo.

Commento: un problema del genere può essere visualizzato tramite un grafo; ogni arco è un tratto di strada mentre ogni vertice è un luogo specifico ad esempio un incrocio (in particolare, con riferimento al grafo disegnato di seguito, la casa è il vertice C mentre l'università è il vertice U); inoltre ad ogni arco può essere associato un numero che è indica i minuti che occorrono per percorrere tale arco (cioè il corrispondente tratto di strada); a questo punto un problema del genere – come vedremo – può essere modellato definendo, per ogni arco denotato con (i,j) dove i e j sono i vertici che individuano l'arco, una variabile decisionale "binaria" x_{ij} che avrà valore 1 se l'arco i fa parte del cammino di tempo minimo e avrà valore 0 altrimenti.

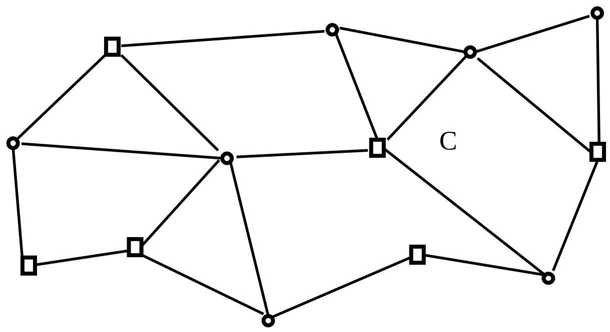


• <u>Attività fisica settimanale</u>
Una persona vuole fare attività fisica, con il vincolo di bruciare almeno una certa quantità di calorie, e con l'obiettivo di minimizzare il numero di ore da dedicare all'attività fisica. Egli può effettuare attività fisiche che fanno parte di un insieme $A = \{1, \ldots, n\}$ di attività fisiche. Ogni attività fisica i, per $i = 1, \ldots, n$, può essere eseguita (in una settimana) al più un certo numero di ore h_i . D'altro canto ogni attività fisica i, per $i = 1, \ldots, n$, garantisce di bruciare b_i calorie per ogni ora. La persona si pone il vincolo di bruciare almeno B calorie (in una settimana). Il problema è scegliere quante ore dedicare a ogni attività fisica (in una settimana), con il vincolo di bruciare almeno B calorie, in modo da minimizzare il numero di ore dedicate all'attività fisica.

Commento: un problema del genere – come vedremo – può essere modellato definendo, per ogni attività fisica i, una variabile decisionale x_i che indicherà il numero di ore dedicate all'attività i.

• <u>Commissioni</u>
Una persona una mattina deve fare un giro di commissioni, cioè, deve: uscire di casa, recarsi in diversi posti, e tornare a casa. Il problema è scegliere una sequenza di questi posti che permetta di effettuare il giro di commissioni in tempo minimo, cioè, il problema è scegliere un giro di commissioni (ce ne può essere più di uno) che sia di tempo minimo.

Commento: un problema del genere può essere visualizzato tramite un grafo; ogni arco è un tratto di strada mentre ogni vertice è un luogo specifico ad esempio un incrocio (in particolare, con riferimento al grafo disegnato di seguito, la casa è il vertice C e i posti relativi alle commissioni sino disegnati come quadrati); inoltre ad ogni arco può essere associato un numero che è indica i minuti che occorrono per percorrere tale arco (cioè il corrispondente tratto di strada); a questo punto un problema del genere – come vedremo – può essere modellato inizialmente definendo, per ogni arco denotato con (i,j) dove i e j sono i vertici che individuano l'arco, una variabile decisionale "binaria" x_{ij} che avrà valore 1 se l'arco i fa parte del giro di tempo minimo e avrà valore 0 altrimenti.



• <u>Buste della spesa</u>

Una persona si trova alla cassa di un supermercato e deve imbustare gli oggetti che ha acquistato, identificati da un insieme A, in due buste. Ogni oggetto i, per $i \in A$, ha un suo peso a_i. Il problema è distribuire gli oggetti fra le due buste (assumendo che ogni busta possa comunque contenere tutti gli oggetti) in modo da bilanciare il peso fra le due buste.

Commento: un problema del genere – come vedremo – può essere modellato definendo, per ogni oggetto i, una variabile decisionale "binaria" x_i che avrà valore 1 se l'oggetto i è imbustato nella prima busta e avrà valore 0 se l'oggetto è imbustato nella seconda busta.

2. Esempi di modelli di programmazione lineare (intera)

In questo capitolo vedremo molti esempi del primo passaggio dello schema nel caso in cui il modello matematico è un problema di Programmazione Lineare (Intera).

Per effettuare il primo passaggio dello schema in genere si procede:
(i) definendo le *variabili decisionali* del problema;
(ii) definendo di conseguenza la *funzione obiettivo* e i *vincoli* (cioè le equazioni e/o disequazioni lineari e gli eventuali vincoli di interezza) del problema.

A. Esempi base

I seguenti sette esempi sono introdotti come riferimento (tipo costellazioni) ricordando che il primo passaggio dello schema si ottiene grazie all'abilità e all'esperienza umana.

A1. *Componendo risorse*
Una ditta produce due tipi di prodotto, siano A e B, componendo tre risorse, siano R, S, T. In dettaglio:
per produrre 1 unità di A servono: 2 u. di R, 3 u. di S, 1 u. di T;
per produrre 1 unità di B servono: 0 u. di R, 1 u. di S, 3 u. di T.
Il ricavo unitario dalla vendita di A e B è rispettivamente 15 e 20.
Il costo unitario delle risorse R, S, T è rispettivamente 2, 1, 3.

– *massimizzare ricavi, con risorse date*
Si hanno rispettivamente 100, 70, 80 unità di R, S, T. Il problema è determinare le quantità di A e B da produrre in modo da massimizzare il ricavo totale.

Variabili decisionali: x_A, x_B (unità rispettivamente di A e B da produrre).

Modello: max $15\, x_A + 20\, x_B$

$2\, x_A$	≤ 100	(disponibilità R)
$3\, x_A + 1\, x_B$	≤ 70	(disponibilità S)
$1\, x_A + 3\, x_B$	≤ 80	(disponibilità T)
$x_A, x_B \geq 0$		
x_A, x_B interi		(se necessario)

– *minimizzare costi, con produzione data*

Bisogna produrre rispettivamente 50 e 70 unità di A e B. Il problema è determinare le quantità di R, S, T da acquistare in modo da minimizzare il costo totale. Tale problema si risolve direttamente mediante moltiplicazioni (le quantità sono: $2 \cdot 50 + 0 \cdot 70$; $3 \cdot 50 + 1 \cdot 70$; $1 \cdot 50 + 3 \cdot 70$), cioè ha una soluzione “forzata”, non c’è da modellare.

A2. *Decomponendo risorse*

Una ditta produce due tipi di prodotto, siano A e B, decomponendo tre risorse, siano R, S, T. In dettaglio:

da 1 unità di R si ricavano: 2 u. di A, 3 u. di B;

da 1 unità di S si ricavano: 7 u. di A, 2 u. di B;

da 1 unità di T si ricavano: 4 u. di A, 0 u. di B.

Il ricavo unitario dalla vendita di A e B è rispettivamente 30 e 25.

Il costo unitario delle risorse R, S, T è rispettivamente 20, 40, 30.

– *massimizzare ricavi, con risorse date*

Si hanno rispettivamente 40, 80, 50 unità di R, S, T. Il problema è determinare le quantità di A e B da produrre in modo da massimizzare il ricavo totale.

Tale problema si risolve direttamente mediante moltiplicazioni (le quantità sono: $2 \cdot 40 + 7 \cdot 80 + 4 \cdot 50$; $3 \cdot 40 + 2 \cdot 80 + 0 \cdot 50$), cioè ha una soluzione “forzata”, non c’è da modellare.

– *minimizzare costi, con produzione data*

Bisogna produrre rispettivamente 80 e 100 unità di A e B. Il problema è determinare le quantità di R, S, T da acquistare in modo da minimizzare il costo totale.

Variabili decisionali: x_R, x_S, x_T (unità rispettivamente di R, S, T da acquistare).

Modello: min $20\, x_R + 40\, x_S + 30\, x_T$

$2\, x_R + 7\, x_S + 4\, x_T \geq 80$ (produzione di A)

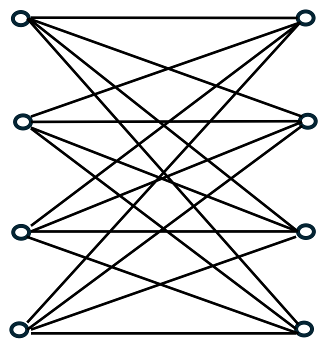
$3\, x_R + 2\, x_S + 0\, x_T \geq 100$ (produzione di B)

$x_R, x_S, x_T \geq 0$

x_R, x_S, x_T interi (se necessario)

A3. *Assegnamento*

Bisogna eseguire n lavori avendo a disposizione n macchine. In particolare: ogni macchina esegue un solo lavoro, ogni lavoro è eseguito da una sola macchina. Per eseguire il lavoro i (per $i = 1, \ldots, n$), la macchina j (per $j = 1, \ldots, m$) richiede un costo c_{ij} . Il problema è assegnare a ciascuna macchina un lavoro al fine di minimizzare il costo totale.



LAVORI MACCHINE

Variabili decisionali: x_{ij} per $i = 1, \ldots, n$ e $j = 1, \ldots, n$ [variabile x_{ij} $\leftrightarrow$ arco (i, j) del grafo];

x_{ij} avrà valore 1 se il lavoro i è assegnato alla macchina j;

x_{ij} avrà valore 0 se il lavoro i non è assegnato alla macchina j.

Modello: min $\displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} x_{ij}$

$\displaystyle\sum_{i=1}^{n} x_{ij} = 1$ per $j = 1, \ldots, n$

(ogni macchina esegue un solo lavoro)

$\displaystyle\sum_{j=1}^{n} x_{ij} = 1$ per $i = 1, \ldots, n$

(ogni lavoro è eseguito da una sola macchina)

$x_{ij} \geq 0$ per $i = 1, \ldots, n$ e $j = 1, \ldots, n$

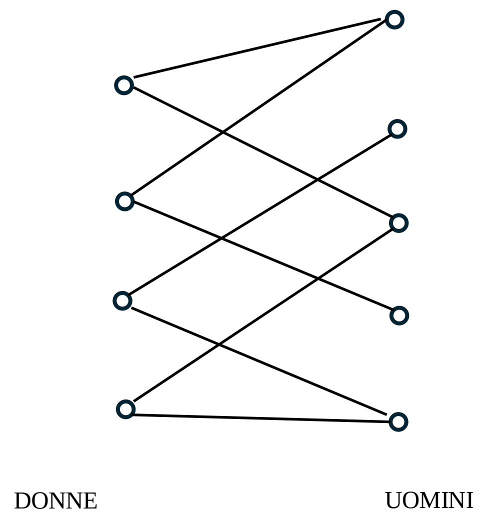
$x_{ij} \leq 1$ per $i = 1, \ldots, n$ e $j = 1, \ldots, n$

x_{ij} <u>intero</u> per $i = 1, \ldots, n$ e $j = 1, \ldots, n$

A4. *Matching*

Un insieme N di n donne e un insieme M di m uomini si recano presso un'agenzia matrimoniale. Mediante colloqui e questionari, l'agenzia matrimoniale stabilisce che presumibilmente la donna i (per $i = 1, \ldots, n$) è compatibile con un certo sottoinsieme M_i di M e che di conseguenza l'uomo j (per $j = 1, \ldots, m$) è compatibile con un certo sottoinsieme N_j di N. Il problema è formare il maggior numero possibile di coppie.

Sia F l'insieme delle coppie (i, j) *compatibili*, cioè, $F = \{(i, j) : i \in N_j\}$ (ovvero $F = \{(i, j) : j \in M_i\}$).



Variabili decisionali: x_{ij} per ogni coppia $(i, j) \in F$ [variabile $x_{ij} \leftrightarrow$ arco (i, j) del grafo];

x_{ij} avrà valore 1 se la donna i è accoppiata all'uomo j;

x_{ij} avrà valore 0 se la donna i non è accoppiata all'uomo j.

Modello: max $\displaystyle\sum_{(i,j)\in F} x_{ij}$

$$\sum_{(i,j)\in F} x_{ij} \le 1 \qquad \text{per } i = 1, \ldots, n$$

(ogni donna è accoppiata al più a un uomo)

$$\sum_{(i,j)\in F} x_{ij} \le 1 \qquad \text{per } j = 1, \ldots, m$$

(ogni uomo è accoppiato al più a una donna)

$$x_{ij} \ge 0 \qquad \text{per ogni } (i, j) \in F$$

$x_{ij} \leq 1$ per ogni $(i, j) \in F$

x_{ij} intero per ogni $(i, j) \in F$

A5. *Riempimento* (*Packing*)

Un'azienda ha a disposizione un budget b, per finanziare alcuni progetti, da scegliere fra n progetti. Ogni progetto i, per $i = 1, \ldots, n$, ha un costo a_i, D'altro canto ogni progetto i, per $i = 1, \ldots, n$, garantisce dopo un anno un profitto p_i. Non è possibile finanziare parzialmente un progetto. Il problema è scegliere quali progetti finanziare in modo da massimizzare il profitto totale dopo un anno.

Variabili decisionali: x_i per $i = 1, \ldots, n$;

x_i avrà valore 1 se il progetto i è scelto;

x_i avrà valore 0 se il progetto i non è scelto.

Modello: max $p_1x_1 + p_2x_2 + \ldots + p_nx_n$

$a_1x_1 + a_2x_2 + \ldots + a_nx_n \leq b$

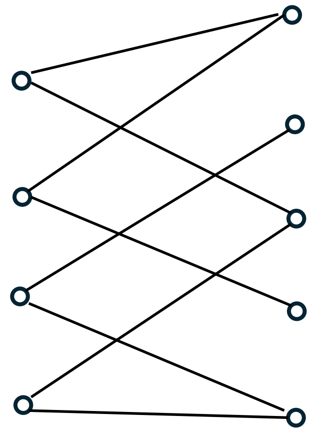
$x_i \geq 0$ per $i = 1, \ldots, n$

$x_i \leq 1$ per $i = 1, \ldots, n$

x_i intero per $i = 1, \ldots, n$

A6. *Copertura* (*Covering*)

In una scuola-lingue si insegnano m lingue. La scuola vuole convocare degli interpreti esterni, da scegliere in un insieme N di n interpreti, ognuno dei quali è qualificato solo per alcune lingue: per $j = 1, \ldots, m$, sia N_j il sottoinsieme di N degli interpreti qualificati per la lingua j. La scuola desidera che, per ognuna delle lingue insegnate, sia convocato almeno un interprete qualificato. Il problema è convocare il minor numero possibile di interpreti.



Variabili decisionali: x_i per $i = 1, \ldots, n$ [variabile $x_i \leftrightarrow$ vertice i in INTERPRETI del grafo];

x_i avrà valore 1 se l'interprete i è scelto;

x_i avrà valore 0 se l'interprete i non è scelto.

Modello: min $\sum_{i=1}^{n} x_i$

$$\sum_{i \in N_j} x_i \geq 1 \qquad \text{per } j = 1, \ldots, m$$

(per ogni lingua esiste almeno un interprete)

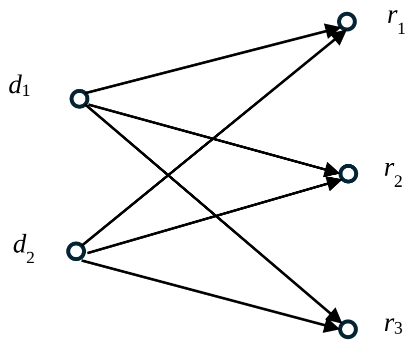
$x_i \geq 0$ per $i = 1, \ldots, n$

$x_i \leq 1$ per $i = 1, \ldots, n$

x_i <u>intero</u> per $i = 1, \ldots, n$

A7. *Trasporti*

Ci sono s località sorgente e t località destinazione. Ogni località sorgente $i \in \{1, \ldots, s\}$ ha a disposizione $d_i \geq 0$ unità di un certo tipo di merce, e ogni destinazione $j \in \{1, \ldots, t\}$ richiede almeno $r_j \geq 0$ unità della stessa merce. Per ogni coppia (i, j), per $i \in \{1, \ldots, s\}$ e $j \in \{1, \ldots, t\}$, sono inoltre stabiliti il costo unitario di trasporto c_{ij} e la quantità massima trasportabile q_{ij}. Il problema è pianificare i trasporti in modo da minimizzare il costo totale.



costi c_{ij}

SORGENTI DESTINAZIONI

Variabili decisionali: x_{ij} per $i = 1, \ldots, s$ e $j = 1, \ldots, t$ [variabile x_{ij} $\leftrightarrow$ arco (i, j) del grafo];
x_{ij} indica la quantità trasportata dalla sorgente i alla destinazione j.

Modello: min $\sum_{i=1}^{s}\sum_{j=1}^{t} c_{ij}x_{ij}$

$\sum_{j=1}^{t} x_{ij} \le d_i$ per $i = 1, \ldots, s$ (vincoli di disponibilità)

$\sum_{i=1}^{s} x_{ij} \ge r_j$ per $j = 1, \ldots, t$ (vincoli di richiesta)

$x_{ij} \le q_{ij}$ per $i = 1, \ldots, s$ e $j = 1, \ldots, t$ (vincoli di capacità)

$x_{ij} \ge 0$ per $i = 1, \ldots, s$ e $j = 1, \ldots, t$

x_{ij} intero per $i = 1, \ldots, s$ e $j = 1, \ldots, t$ (se necessario)

B. Esempi base - Esercizi

Formulare ognuno dei seguenti sette problemi in termini di Programmazione Lineare (Intera). Ognuno di tali problemi è riconducibile a uno dei sette esempi base introdotti prima.

B1. Un libero professionista ha a disposizione ogni settimana 10 ore da dedicare all'allargamento della propria attività. In particolare, c'è la possibilità di prestare consulenza a cinque nuovi clienti, siano A, B, C, D, E. Ognuno di essi richiede una quantità di tempo lavorativo settimanale, come di seguito: $t_A = 4$, $t_B = 5$, $t_C = 4$, $t_D = 3$, $t_E = 1$. Quindi il libero professionista comunque non potrebbe prestare consulenza per tutti loro. In base al tipo di consulenza da svolgere, a ognuno può chiedere una parcella settimanale che procura un ricavo, come di seguito: $p_A = 70$, $p_B = 30$, $p_C = 70$, $p_D = 40$, $p_E = 30$. Il problema è scegliere i clienti a cui prestare consulenza in modo da massimizzare il ricavo totale.

B2. Una ditta decide di intraprendere 4 attività. A tale scopo ci sono a disposizione 4 agenzie già operanti. Per diversi motivi, ogni agenzia può gestire al più una sola di queste attività, e ogni attività può essere gestita al più da una sola di queste agenzie. Infine, si stima a priori che assegnare all'agenzia i (per $i = 1, \ldots, 4$) l'attività j (per $j = 1, \ldots, 4$) garantisca un profitto p_{ij}. Il problema è assegnare ad ogni agenzia una attività in modo da massimizzare il profitto totale.
Variante: Con le ipotesi sopra indicate, si supponga che una delle agenzie debba essere chiusa per certi motivi. Il problema è scegliere quali attività intraprendere (dato che ora se ne possono intraprendere solo tre) in modo da massimizzare il profitto totale.

B3. Una persona vuole fare una dieta. In particolare deve assumere due tipi di sostanze, cioè proteine e vitamine, che può ricavare comprando tre tipi di alimenti, cioè frutta, latte, uova.
In dettaglio:
1 unità di frutta contiene: 0 u. di proteine, 7 u. di vitamine;
1 unità di latte contiene: 2 u. di proteine, 3 u. di vitamine;
1 unità di uova contiene: 5 u. di proteine, 1 u. di vitamine.
Il costo unitario di frutta, latte, uova è rispettivamente di 10, 20, 10.
La dieta richiede di assumere almeno 15 unità di proteine e 25 unità di vitamine.
Il problema è determinare le quantità di frutta, latte, uova, soddisfacendo la richiesta di sopra, da acquistare in modo da minimizzare il costo totale.

B4. Un'azienda vinicola desidera produrre due tipi di vino: uno da tavola, uno da dessert. Il profitto che l'azienda trae dalla produzione di 1 unità di vino da tavola è 3, mentre dalla produzione di 1 unità di vino da dessert è 7. Tale produzione necessita di una particolare combinazione di due tipi di uva: diciamo di tipo A e di tipo B rispettivamente. Per produrre 1 unità di vino da tavola, si ha bisogno di 3 unità di uva di tipo A, e di 2 unità di uva di tipo B. Per produrre 1 unità di vino da dessert, si ha bisogno di 1 unità di uva di tipo A, e di 4 unità di uva di tipo B. Infine, l'azienda ha a disposizione 1000 unità di uva di tipo A e 400 unità di uva di tipo B. Il problema è determinare le quantità di vino da tavola e da dessert da produrre in modo da massimizzare il profitto totale.

B5. Una ditta che tratta sale marino ha 2 depositi, siano A e B, che riforniscono 4 suoi negozi. La disponibilità di sale di ogni deposito è pari rispettivamente a $d_A = 190$ e $d_B = 370$ unità. La richiesta di sale da parte di ogni negozio è pari rispettivamente a $r_1 = 130$, $r_2 = 140$, $r_3 = 100$, $r_4 = 70$ unità. Il costo unitario c_{ij} di trasporto dal deposito i al negozio j è pari a: $c_{A1} = 2$, $c_{A2} = 3$, $c_{A3} = 3$, $c_{A4} = 1$, $c_{B1} = 3$, $c_{B2} = 2$, $c_{B3} = 1$, $c_{B4} = 2$. (Opzionale: per motivi logistici, il deposito A non può rifornire ogni singolo negozio con più di 100 unità, mentre il deposito B non può rifornire ogni singolo negozio con più di 150 unità). Il problema è pianificare i rifornimenti dei negozi in modo da minimizzare il costo totale.

B6. Un'agenzia vuole aprire delle filiali in una certa regione, che ha cinque province, siano 1, 2, 3, 4, 5. Per ogni provincia $i \in \{1, \ldots, 5\}$ c'è la possibilità di aprire una filiale F_i nel capoluogo della provincia i. In base alle caratteristiche di tali eventuali filiali e del territorio, si stima che:
la eventuale filiale F_1 servirebbe solo le province 1, 3;
la eventuale filiale F_2 servirebbe solo le province 2, 3;
la eventuale filiale F_3 servirebbe solo le province 3, 4;
la eventuale filiale F_4 servirebbe solo le province 1, 2, 4;
la eventuale filiale F_5 servirebbe solo le province 2, 3, 5.
Il problema è determinare quali filiali aprire, garantendo che ogni provincia sia servita, in modo da minimizzare il totale delle filiali aperte (cioè in modo da aprirne il meno possibile).
Variante: si assuma che aprire la filiale F_i comporti un costo c_i ($i = 1, \ldots, 5$): determinare quali filiali aprire, garantendo che ogni provincia sia servita, in modo da minimizzare il costo totale.

B7. In un'agenzia di viaggio si presentano 5 clienti. Le offerte dell'agenzia riguardano un insieme di 3 località. Ogni cliente gradirebbe recarsi in vacanza solo in un certo sottoinsieme delle località: il cliente 1 solo nella località 3; il cliente 2 solo nelle località 1, 3; il cliente 3 solo nella località 1; il cliente 4 solo nelle località 1, 3; il cliente 5 solo nelle località 1, 2. Per ogni località c'è a disposizione solo 1 posto. Il problema è accontentare il maggior numero possibile di clienti.

Variante 1: Con le ipotesi sopra indicate, si supponga che per ogni località j ci siano a disposizione solo b_j posti: nello specifico, per la località 1 ci sono 4 posti, per la località 2 ci sono 2 posti, per la località 3 di sono 2 posti. Il problema è accontentare il maggior numero possibile di clienti.

Variante 2: Con le ipotesi sopra indicate, si supponga che mandare un cliente in una località j renda all'agenzia un profitto p_j: nello specifico, la località 1 rende un profitto 150, la località 2 rende un profitto 250, la località 3 rende un profitto 100. Un problema alternativo da risolvere può essere il seguente: massimizzare il profitto dell'agenzia.

C. Cinque esempi specifici

Di seguito sono riportati, in termini di programmazione lineare (intera), i modelli di cinque problemi specifici che saranno ripresi nella seconda parte del programma del corso.

C1. *Cammino di costo minimo*

Sia $G = (V, A)$ un grafo orientato e siano s e t due vertici di G. Ad ogni arco $(i, j) \in A$ è associato un costo $q_{ij} \geq 0$. Il costo di un cammino da s a t è dato dalla somma dei costi degli archi che formano il cammino. Il problema è determinare un cammino da s a t di costo minimo.

Soluzione

Variabili decisionali: x_{ij} per ogni arco $(i, j) \in A$;

x_{ij} avrà valore 1 se l'arco (i, j) sta nel cammino;

x_{ij} avrà valore 0 se l'arco (i, j) non sta nel cammino.

Modello: min $\sum_{(i,j)\in A} q_{ij} x_{ij}$

$$\sum_{(s,j)\in A} x_{sj} = 1$$

$$\sum_{(i,j)\in A} x_{ij} - \sum_{(j,i)\in A} x_{ji} = 0 \quad \text{per ogni vertice } i \neq s, t \text{ di } G$$

$x_{ij} \geq 0$ per ogni $(i, j) \in A$

$x_{ij} \leq 1$ per ogni $(i, j) \in A$

x_{ij} intero per ogni $(i, j) \in A$

C2. *Pianificazione di progetti*

Un *progetto* è un insieme di n attività A_i, per $i \in \{1, \ldots, n\}$, ciascuna con una durata $d_i \geq 0$ nota. Fra alcune attività sono specificate relazioni di precedenza $A_i < A_j$ ad indicare che l'istante di completamento dell'attività A_i deve precedere l'istante di inizio dell'attività A_j. Il problema è pianificare le attività in modo da minimizzare il tempo di completamento del progetto.

Soluzione

Variabili decisionali: t_i^- e t_i^+ per $i = 1, \ldots, n$;

t_i^- indica l'istante in cui l'attività A_i ha inizio;

t_i^+ indica l'istante in cui l'attività A_i ha termine;

si introduce anche una variabile ausiliaria y che avrà valore $y = \max\{t_i^+ : i = 1, \ldots, n\}$

Modello: min y

$t_i^+ \geq t_i^- + d_i$ per $i = 1, \ldots, n$

$t_i^+ \leq t_j^-$ per ogni relazione di precedenza $A_i < A_j$

$t_i^- \geq 0$ per $i = 1, \ldots, n$

$t_i^+ \leq y$ per $i = 1, \ldots, n$

C3. *Massimo flusso*

Sia $G = (V, A)$ un grafo orientato e siano s e t due vertici di G. Ad ogni arco $(i, j) \in A$ è associata una capacità $b_{ij} \geq 0$. Un flusso da s a t è un'assegnazione di valori $x_{ij} \geq 0$ per ogni arco $(i, j) \in A$ tale che:

:: $\sum_{(i,j)\in A} x_{ij} - \sum_{(j,i)\in A} x_{ji} = 0$ per ogni vertice $i \neq s, t$ di G (cioè per ogni vertice $i \neq s, t$ di G, la somma di ciò che entra è pari alla somma di ciò che esce);

:: $x_{ij} \leq b_{ij}$ per ogni arco $(i, j) \in A$.

Il valore di un flusso da s a t è $\sum_{(s,j)\in A} x_{sj}$.

Il problema è determinare un flusso da s a t di valore massimo.

Soluzione

Variabili decisionali: x_{ij} per ogni arco $(i, j) \in A$;

x_{ij} indica l'assegnazione di flusso per l'arco (i, j).

Modello: max $\sum_{(s,j)\in A} x_{sj}$

$\sum_{(i,j)\in A} x_{ij} - \sum_{(j,i)\in A} x_{ji} = 0$ per ogni vertice $i \neq s, t$ di G

$x_{ij} \leq b_{ij}$ per ogni $(i, j) \in A$

$x_{ij} \geq 0$ per ogni $(i, j) \in A$

x_{ij} <u>intero</u> per ogni $(i, j) \in A$ (se necessario)

C4. *Programmazione della produzione*

Un'azienda deve produrre un bene, per soddisfare un piano di vendita di 4 mesi, che stabilisce le vendite da effettuare alla fine di ogni mese. La capacità produttiva varia da mese a mese, così come il costo unitario di produzione e il costo unitario di giacenza (alla fine di ogni mese nel magazzino di cui l'azienda può servirsi). Non ci sono giacenze in magazzino all'inizio, né si desidera averne alla fine dei 4 mesi. In dettaglio:

Mese	vendite da effettuare	capacità prod.	costo prod.	costo giacenza
1	20 unità	40 unità	34	2
2	30 unità	50 unità	36	3
3	50 unità	30 unità	32	2
4	40 unità	50 unità	38	

Il problema è determinare le quantità di bene da produrre in ogni mese, soddisfacendo il piano di vendita, in modo da minimizzare il costo totale.

Soluzione

Variabili decisionali: x_i per $i = 1, 2, 3, 4$;

x_i indica la quantità di prodotto da produrre nel mese i.

Modello: min $34x_1 + 36x_2 + 32x_3 + 38x_4 + 2(x_1 - 20) + 3(x_1 + x_2 - 50) +$

$+ 2(x_1 + x_2 + x_3 - 100)$

$x_1 \leq 40$

$x_2 \leq 50$

$x_3 \leq 30$

$x_4 \leq 50$

$x_1 \geq 20$

$x_1 + x_2 \geq 50$

$x_1 + x_2 + x_3 \geq 100$

$x_1 + x_2 + x_3 + x_4 = 140$

$x_i \geq 0$ per $i = 1, 2, 3, 4$

x_i intero per $i = 1, 2, 3, 4$ (se necessario)

C5. *Localizzazione di impianti*

Bisogna attivare un certo numero di impianti, da scegliere fra m potenziali impianti, che dovranno rifornire n clienti. Attivare il potenziale impianto j (per $j \in \{1, \ldots, m\}$) comporta un costo $f_j \geq 0$. Ogni impianto si intende abbia capacità di rifornimento illimitata. Ogni cliente sarà collegato ad un unico impianto: in particolare collegare un cliente i ad un impianto j comporta un costo c_{ij} (per $i \in \{1, \ldots, n\}$ e per $j \in \{1, \ldots, m\}$). Il problema è scegliere quali impianti attivare, collegando ogni cliente a un impianto, in modo da minimizzare i costi complessivi di attivazione degli impianti e di collegamento ai clienti.

Soluzione

Variabili decisionali: x_{ij} per $i = 1, \ldots, n$, e $j = 1, \ldots, m$; y_j per $j = 1, \ldots, m$;

x_{ij} avrà valore 1 se il cliente i è collegato all'impianto j;

x_{ij} avrà valore 0 se il cliente i non è collegato all'impianto j;

y_j avrà valore 1 se l'impianto j è attivato;

y_j avrà valore 0 se l'impianto j non è attivato.

Modello: min $\displaystyle\sum_{j=1}^{m} f_j y_j + \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij} x_{ij}$

$\displaystyle\sum_{j=1}^{m} x_{ij} = 1$ per $i = 1, \ldots, n$

(ogni cliente è collegato ad un unico impianto)

$x_{ij} \leq y_j$ per $i = 1, \ldots, n$, e $j = 1, \ldots, m$

(ogni impianto si attiva se c'è almeno un cliente collegato)

$x_{ij} \geq 0$ per $i = 1, \ldots, n$, e $j = 1, \ldots, m$

$x_{ij} \leq 1$ per $i = 1, \ldots, n$, e $j = 1, \ldots, m$

x_{ij} <u>intero</u> per $i = 1, \ldots, n$, e $j = 1, \ldots, m$

$y_j \geq 0$ per $j = 1, \ldots, m$

$y_j \leq 1$ per $j = 1, \ldots, m$

y_j <u>intero</u> per $j = 1, \ldots, m$

D. Esempi "con la grande M"

Un artificio molto utile nel formulare un problema in termini di Programmazione Lineare (Intera) sembra essere quello della "grande M"*; di seguito riportiamo tre esempi, ognuno in un certo contesto, ma naturalmente questo artificio può essere applicato in diversi contesti.*

D1. *Prima variante a "problema dei trasporti": **costi fissi**.*

Nel problema dei trasporti la funzione costo per il trasporto di merce da sorgente i a deposito j è $c_{ij}x_{ij}$. Si consideri la variante in cui tale funzione sia $F_{ij} + c_{ij}x_{ij}$, dove $F_{ij} \geq 0$ rappresenta un costo fisso (un costo di avviamento), tale che $F_{ij} \geq 0$ se $x_{ij} > 0$ (cioè se il trasporto fra i e j avviene) e $F_{ij} = 0$ se $x_{ij} = 0$ (cioè se il trasporto fra i e j non avviene).

Soluzione

Si introducono nuove variabili decisionali: y_{ij} per $i = 1, \ldots, s$ e $j = 1, \ldots, t$;

y_{ij} avrà valore 1 se si trasporta merce da sorgente i a destinazione j (cioè se $x_{ij} > 0$)

y_{ij} avrà valore 0 se non si trasporta merce da sorgente i a destinazione j (cioè se $x_{ij} = 0$)

Modello: min $\displaystyle\sum_{i=1}^{s}\sum_{j=1}^{t} c_{ij}x_{ij} + \sum_{i=1}^{s}\sum_{j=1}^{t} F_{ij}y_{ij}$ [variazione in funzione obiettivo]

$\displaystyle\sum_{j=1}^{t} x_{ij} \leq d_i$ per $i = 1, \ldots, s$ (vincoli di disponibilità)

$\displaystyle\sum_{i=1}^{s} x_{ij} \geq r_j$ per $j = 1, \ldots, t$ (vincoli di richiesta)

$x_{ij} \geq 0$ per $i = 1, \ldots, s$ e $j = 1, \ldots, t$

x_{ij} intero per $i = 1, \ldots, s$ e $j = 1, \ldots, t$ (se necessario)

[aggiunta di vincoli]

$x_{ij} \leq M\, y_{ij}$ per $i = 1, \ldots, s$ e $j = 1, \ldots, t$

$y_{ij} \geq 0$ per $i = 1, \ldots, s$ e $j = 1, \ldots, t$

$y_{ij} \leq 1$ per $i = 1, \ldots, s$ e $j = 1, \ldots, t$

y_{ij} intero per $i = 1, \ldots, s$ e $j = 1, \ldots, t$

Commento sul vincolo $x_{ij} \leq M\, y_{ij}$:

M è un numero molto grande (più grande di ogni possibile termine della formulazione);

se $x_{ij} > 0$, allora $x_{ij} \leq M\, y_{ij}$ implica $y_{ij} = 1$ (quindi il costo fisso nella f. obiettivo è attivato);

se $x_{ij} = 0$, allora $x_{ij} \leq M\, y_{ij}$ è sempre soddisfatto, non crea problemi (si osservi che essendo il termine F_{ij} positivo, la f. obiettivo determinerà $y_{ij} = 0$).

D2. *Seconda variante a "problema dei trasporti":* ***lotti minimi***.

Nel problema dei trasporti ogni quantità effettivamente trasportata x_{ij} (cioè per cui $x_{ij} > 0$) non è vincolata ad assumere almeno un certo valore. Si consideri la variante in cui ogni quantità effettivamente trasportata x_{ij} (cioè per cui $x_{ij} > 0$) debba essere maggiore o uguale a un valore L_{ij} che rappresenta il lotto minimo da trasportare in caso di effettivo trasporto.

Soluzione

Si introducono nuove variabili decisionali: y_{ij} per $i = 1, \ldots, s$ e $j = 1, \ldots, t$;

y_{ij} avrà valore 1 se si trasporta merce da sorgente i a destinazione j (cioè se $x_{ij} > 0$)

y_{ij} avrà valore 0 se non si trasporta merce da sorgente i a destinazione j (cioè se $x_{ij} = 0$)

Modello: min $\sum_{i=1}^{s}\sum_{j=1}^{t} c_{ij} x_{ij}$

$\sum_{j=1}^{t} x_{ij} \le d_i$ per $i = 1, \ldots, s$ (vincoli di disponibilità)

$\sum_{i=1}^{s} x_{ij} \ge r_j$ per $j = 1, \ldots, t$ (vincoli di richiesta)

$x_{ij} \ge 0$ per $i = 1, \ldots, s$ e $j = 1, \ldots, t$;

x_{ij} intero per $i = 1, \ldots, s$ e $j = 1, \ldots, t$. (se necessario)

[aggiunta di vincoli]

$x_{ij} \le M\, y_{ij}$ per $i = 1, \ldots, s$ e $j = 1, \ldots, t$

$x_{ij} \ge L_{ij}\, y_{ij}$ per $i = 1, \ldots, s$ e $j = 1, \ldots, t$

$y_{ij} \ge 0$ per $i = 1, \ldots, s$ e $j = 1, \ldots, t$

$y_{ij} \le 1$ per $i = 1, \ldots, s$ e $j = 1, \ldots, t$

y_{ij} intero per $i = 1, \ldots, s$ e $j = 1, \ldots, t$

Commento sui vincoli $x_{ij} \le M\, y_{ij}$ e $x_{ij} \ge L_{ij}\, y_{ij}$:

M è un numero molto grande (più grande di ogni possibile termine della formulazione);

se $x_{ij} > 0$, allora $x_{ij} \le M\, y_{ij}$ implica $y_{ij} = 1$ e di conseguenza $x_{ij} \ge L_{ij}\, y_{ij}$ diventa $x_{ij} \ge L_{ij}$;

se $x_{ij} = 0$, allora nessun problema è creato (si osservi che $0 \ge L_{ij}\, y_{ij}$ implica $y_{ij} = 0$, mentre $0 \le M\, y_{ij}$ è sempre soddisfatto).

D3. *Schedulazione su un processore (con deadlines e release times):* ***vincoli disgiuntivi****.*

Bisogna far eseguire n lavori ad un processore, che ne può eseguire uno alla volta.

Ogni lavoro i, per $i \in \{1, \ldots, n\}$, è caratterizzato da:

p_i = tempo di processamento;

r_i = istante prima del quale il lavoro i non può iniziare (*release time*);

d_i = istante entro il quale il lavoro i deve essere completato (*deadline*).

Problema: assegnare al processore una sequenza dei lavori, rispettando i release times e i deadlines, in modo da minimizzare la somma degli istanti di completamento delle lavorazioni.

Soluzione

Variabili decisionali: C_i per $i = 1, \ldots, n$; x_{ij} per $i = 1, \ldots, n$ e $j = 1, \ldots, n$ ($\neq i$);

C_i = istante di completamento del lavoro i;

x_{ij} avrà valore 1 se il lavoro i è eseguito prima del lavoro j

x_{ij} avrà valore 0 se il lavoro i non è eseguito prima del lavoro j

$$\min \quad \sum_{i=1}^{n} C_i$$

$$\begin{array}{ll}
C_i \geq r_i + p_i & \text{per } i = 1, \ldots, n \\
C_i \leq d_i & \text{per } i = 1, \ldots, n \\
C_i \leq C_j - p_j + M(1 - x_{ij}) & \text{per } i = 1, \ldots, n \text{ e } j = 1, \ldots, n \ (\neq i) \\
C_j \leq C_i - p_i + M x_{ij} & \text{per } i = 1, \ldots, n \text{ e } j = 1, \ldots, n \ (\neq i) \\
x_{ij} \geq 0 & \text{per } i = 1, \ldots, n \text{ e } j = 1, \ldots, n \ (\neq i) \\
x_{ij} \leq 1 & \text{per } i = 1, \ldots, n \text{ e } j = 1, \ldots, n \ (\neq i) \\
x_{ij} \ \underline{\text{intero}} & \text{per } i = 1, \ldots, n \text{ e } j = 1, \ldots, n \ (\neq i)
\end{array}$$

Commento sui due vincoli $C_i \leq C_j - p_j + M(1 - x_{ij})$ e $C_j \leq C_i - p_i + M x_{ij}$:

M è un numero molto grande (più grande di ogni possibile termine della formulazione);

se $x_{ij} = 1$ (cioè i precede j), allora si ha $C_i \leq C_j - p_j$ (che è un vincolo sensato/utile per il problema) e $C_j \leq C_i - p_i + M$ (che diventa un vincolo sempre soddisfatto);

se $x_{ij} = 0$ (cioè i è preceduto da j), allora si ha $C_i \leq C_j - p_i + M$ (che diventa un vincolo sempre soddisfatto) e $C_j \leq C_i - p_i$ (che è un vincolo sensato/utile per il problema).

E. Altri esempi

E1. Produzione su una macchina

Una macchina M che lavora non più di 45 ore alla settimana può produrre 3 tipi di prodotto, P_1, P_2, P_3. Per ciascun tipo di prodotto sono noti: il ricavo unitario, la produzione effettuata da M in 1 ora, la produzione massima effettuabile da M in una settimana (come di seguito).

	ricavo unitario	produzione in 1 ora	produzione max settimanale
P_1	4	50 pezzi	1000 pezzi
P_2	12	25 pezzi	500 pezzi
P_3	3	75 pezzi	1500 pezzi

Determinare le quantità di P_1, P_2, P_3 da produrre in una settimana in modo da massimizzare il ricavo totale.

E2. Produzione in due stabilimenti

Una casa automobilistica dispone di 2 stabilimenti dove vengono prodotti 4 tipi di automobili. La capacità produttiva dei 2 stabilimenti è rispettivamente pari a 8000 e 12000 automobili (si intende che, riguardo la capacità produttiva, per ogni stabilimento è indifferente produrre una automobile di un certo tipo oppure di un altro), mentre la quantità (esatta) da produrre per ogni tipo di automobile è rispettivamente pari a 5000, 4800, 3800, 6400 unità. Produrre nello stabilimento i una automobile di tipo j garantisce un profitto p_{ij} : tali valori p_{ij} sono riportati nella tabella di seguito.

		automobili			
		1	2	3	4
stabilimenti	1	2	3	4	1
	2	2	2	6	2

Determinare le quantità di automobili da produrre in ogni stabilimento in modo da massimizzare il profitto totale.

E3. Scelta dei processi

Una raffineria può utilizzare due procedimenti di raffinazione. Nel processo Alfa, 1 u. di greggio libico e 2 u. di greggio nigeriano producono 5 u. di benzina e 2 u. di gasolio. Nel processo Beta, 4 u. di greggio libico e 2 u. di greggio nigeriano producono 3 u. di benzina e 8 u. di gasolio. Le disponibilità di greggio sono: 100 u. di greggio libico e 150 u. di greggio nigeriano. Le vendite previste (cioè le quantità minime da produrre) sono: 200 u. di benzina e 75 u. di gasolio; i profitti unitari sono p_B e p_G rispettivamente.
Determinare quante volte utilizzare il processo Alfa e il processo Beta in modo da massimizzare il profitto totale.
(suggerimento: definire variabili decisionali x_1 e x_2 che indicano rispettivamente quante volte usare rispettivamente il processo Alfa e il processo Beta)

E4. Pubblicità
Un'agenzia pubblicitaria ha annunciato di essere in grado di investire in modo ottimo il denaro dei suoi clienti usando la programmazione lineare.
Ci sono un insieme N di n audience (esempio: i single, le coppie sposate, ecc.) e un insieme M di m mezzi pubblicitari (esempio: periodici, radio, TV, ecc.). Per ogni audience i (per $i = 1, \ldots, n$) il cliente desidera un livello di esposizione E_i che indica il numero di persone appartenenti all'audience i da raggiungere. Per ogni audience i (per $i = 1, \ldots, n$) e per ogni mezzo pubblicitario j (per $j = 1, \ldots, m$) si definisce un coefficiente a_{ij} che indica il numero di persone appartenenti alla i-esima audience che vengono raggiunte quando si spende 1 Euro nel j-esimo mezzo pubblicitario. In che modo l'agenzia può usare la programmazione lineare? (suggerimento: definire variabili decisionali x_j = somma destinata al mezzo pubblicitario j).

Soluzioni

E1. Variabili decisionali: x_i per $i = 1, 2, 3$;
x_i indica la quantità di prodotto P_i da produrre (in una settimana).

Modello: max $4x_1 + 12x_2 + 3x_3$

$$x_1 \leq 1000$$
$$x_2 \leq 500$$
$$x_3 \leq 1500$$
$$(1/50)\, x_1 + (1/25)\, x_2 + (1/75)\, x_3 \leq 45$$
$$x_i \geq 0 \qquad \text{per } i = 1, 2, 3$$

x_i <u>intero</u> per $i = 1, 2, 3$

E2. Variabili decisionali: x_{ij} per $i = 1, 2$, e $j = 1, 2, 3, 4$;
x_{ij} indica la quantità prodotta nello stabilimento i di automobili di tipo j.

Modello: max $2x_{11} + 3x_{12} + 4x_{13} + 1x_{14} + 2x_{21} + 2x_{22} + 6x_{23} + 2x_{24}$

$x_{11} + x_{12} + x_{13} + x_{14} \leq 8000$

$x_{21} + x_{22} + x_{23} + x_{24} \leq 12000$

$x_{11} + x_{21} = 5000$

$x_{12} + x_{22} = 4800$

$x_{13} + x_{23} = 3800$

$x_{14} + x_{24} = 6400$

$x_{ij} \geq 0$ per $i = 1, 2$, e $j = 1, 2, 3, 4$

x_{ij} <u>intero</u> per $i = 1, 2$, e $j = 1, 2, 3, 4$ (se necessario)

E3. max $p_B (5x_1 + 3x_2) + p_G (2x_1 + 8x_2)$

$x_1 + 4x_2 \leq 100$

$2x_1 + 2x_2 \leq 150$

$5x_1 + 3x_2 \geq 200$

$2x_1 + 8x_2 \geq 75$

$x_1, x_2 \geq 0$

E4. min $\sum_{j=1}^{m} x_j$

$\sum_{j=1}^{m} a_{ij} x_j \geq E_i$ $\quad i = 1, \ldots, n$

$x_j \geq 0$ $\quad j = 1, \ldots, m$

F. Appendice: esempi in ambito sanitario

In letteratura esistono molti articoli sulle possibili applicazioni della Ricerca Operativa allo studio di problemi in ambito sanitario; in particolare sembra che l'utilità di tali possibili applicazioni, che sembrano comunque da adattare a realtà specifiche, sia quella di fornire un supporto alle decisioni.

Link sull'argomento: http://www.choir.utwente.nl/en/orchestra/ e http://orahs.di.unito.it/

Di seguito ci sono alcune possibili applicazioni che usano la Programmazione Linare (Intera).

1. Turni in ospedale (cfr. libro M. Fischetti; inoltre per diversi aspetti cfr. [a2] e [a8])
Bisogna definire i turni degli infermieri. Ogni giorno è necessaria la presenza di un certo numero di infermieri. In particolare il turno di un infermiere dura cinque giorni consecutivi di lavoro seguiti da due giorni consecutivi di riposo. Il problema è minimizzare il numero degli infermieri coinvolti.

Parametri:
:: sia $D = \{1, \ldots, 7\}$ l'insieme dei giorni della settimana;
:: per $i = 1, \ldots, 7$, sia r_i la richiesta stimata di infermieri per il giorno i;
:: i turni sono del tipo: cinque giorni lavorativi consecutivi, seguiti da due giorni di riposo.

Variabili decisionali:
x_i per $i = 1, \ldots, 7$;
$x_i =$ numero di infermieri che inizieranno il turno nel giorno i;

Modello:

$$\min \sum_{i=1}^{7} x_i$$

$$x_4 + x_5 + x_6 + x_7 + x_1 \geq r_1$$

$$x_5 + x_6 + x_7 + x_1 + x_2 \geq r_2$$

$$x_6 + x_7 + x_1 + x_2 + x_3 \geq r_3$$

$$x_7 + x_1 + x_2 + x_3 + x_4 \geq r_4$$

$$x_1 + x_2 + x_3 + x_4 + x_5 \geq r_5$$

$x_2 + x_3 + x_4 + x_5 + x_6 \geq r_6$

$x_3 + x_4 + x_5 + x_6 + x_7 \geq r_7$

$x_i \geq 0$ per $i = 1, \ldots, n,$

x_i <u>intero</u> per $i = 1, \ldots, n.$

<u>2. Sala operatoria</u> (cfr. [a9] slide 22-28; inoltre per diversi aspetti cfr. [a1] e [a6])

Nella sala operatoria di un reparto di Ortopedia devono essere eseguite delle operazioni. La durata giornaliera della disponibilità della sala operatoria è nota. Le durate delle operazioni sono note mediante dati storici. Il problema è determinare una sequenza delle operazioni da eseguire, con l'obiettivo di minimizzare il numero di giorni in cui la sala operatoria è utilizzata.

Parametri:

:: sia $P = \{1, \ldots, n\}$ l'insieme delle operazioni da effettuare;

:: per $i = 1, \ldots, n$, sia d_i la durata stimata (in ore) per l'operazione i;

:: sia $G = \{1, \ldots, m\}$ l'insieme dei giorni in un orizzonte temporale di m giorni;

:: per $j = 1, \ldots, m$, sia K_j il tempo (in ore) in cui è possibile utilizzare la sala operatoria nel giorno j.

Variabili decisionali:

x_{ij} per $i = 1, \ldots, n$, e per $j = 1, \ldots, m$;

$x_{ij} = 1$ se l'operazione i è effettuata nel giorno j;

$x_{ij} = 0$ altrimenti;

y_j per $j = 1, \ldots, m$;

$y_j = 1$ se la sala operatoria è utilizzata nel giorno j;

$y_j = 0$ altrimenti;

Modello:

$$\min \quad \sum_{j=1}^{m} y_j$$

$$\sum_{j=1}^{m} x_{ij} = 1 \qquad \text{per } i = 1, \ldots, n$$

$$\sum_{i=1}^{n} d_i x_{ij} \leq K_j \qquad \text{per } j = 1, \ldots, m$$

$$\sum_{i=1}^{n} x_{ij} \leq M\, y_j \qquad \text{per } j = 1, \ldots, m$$

(dove M è uno scalare molto grande)

$x_{ij} \in \{0, 1\}$ per $i = 1, \ldots, n$ e per $j = 1, \ldots, m$

$y_j \in \{0, 1\}$ per $j = 1, \ldots, m$

<u>3. Allocazione di pazienti in reparti</u> (per diversi aspetti cfr. [a8])

Il problema è allocare il maggior numero possibile di persone, fra quelle che chiedono di essere ricoverate in reparti, tenendo conto che ogni reparto ha a disposizione solo un certo numero di posti e che ogni persona può essere ricoverata solo in certi reparti.

Parametri:

:: sia $P = \{1, \ldots, n\}$ l'insieme delle persone che deve essere ricoverata in ospedale;

:: sia $R = \{1, \ldots, m\}$ l'insieme dei reparti dell'ospedale;

:: per $j = 1, \ldots, m$, sia d_j la disponibilità di posti nel reparto j;

:: sia $F = \{(i, j)$: la persona i può essere ricoverata nel reparto $j\}$

(cioè F è l'insieme delle coppie persona/reparto compatibili, considerando il fatto che ogni persona può essere ricoverata solo in alcuni reparti, per diversi motivi ad esempio per evitare un contagio).

Variabili decisionali:

x_{ij} per ogni $(i, j) \in F$;

$x_{ij} = 1$ se la persona i è assegnata al reparto j;

$x_{ij} = 0$ se la persona i non è assegnata al reparto j.

Modello: max $\displaystyle\sum_{(i,j)\in F} x_{ij}$

$$\sum_{(i,j)\in F} x_{ij} \leq 1 \qquad \text{per } i = 1, \ldots, n$$

$$\sum_{(i,j)\in F} x_{ij} \leq d_j \qquad \text{per } j = 1, \ldots, m$$

$$x_{ij} \in \{0, 1\} \qquad \text{per ogni } (i, j) \in F$$

4. <u>Localizzazione di CUP o di Guardie Mediche</u> (cfr. libro M. Fischetti)

Il problema è stabilire in quali quartieri di una grande città istallare dei CUP, con il vincolo che ogni utente possa raggiungere un CUP in un tempo non superiore a un tempo fissato, con l'obiettivo di istallarne il minor numero possibile.

Lo stesso problema può essere posto con riferimento a comuni di una regione invece che a quartieri di una grande città e con riferimento alle Guardie Mediche invece che ai CUP.

Parametri:

:: sia $Q = \{1, \ldots, n\}$ l'insieme dei quartieri di una grande città;

:: per $i = 1, \ldots, n$ e per $j = 1, \ldots, n$, sia t_{ij} il tempo medio per andare dal quartiere i al quartiere j;

:: sia T il tempo massimo tollerato;

:: per $i = 1, \ldots, n$, sia $R(i) = \{j \in Q: t_{ij} \leq T\}$, cioè $R(i)$ è l'insieme dei quartieri raggiungibili dal quartiere i in un tempo $\leq T$.

Variabili decisionali:

x_i per $i = 1, \ldots, n$;

$x_i = 1$ se nel quartiere i è istallato un CUP;

$x_i = 0$ altrimenti.

Modello:

$$\min \quad \sum_{i=1}^{n} x_i$$

$$\sum_{j \in R(i)} x_j \geq 1 \qquad \text{per } i = 1, \ldots, n$$

$$x_i \in \{0, 1\} \qquad \text{per } i = 1, \ldots, n$$

5. <u>Acquisto di macchinari</u>

Il problema è acquistare nuovi macchinari ospedalieri, con un budget fissato, con l'obiettivo di massimizzare l'utilità dei macchinari acquistati (dove l'*utilità* di un macchinario può essere intesa come il numero delle persone che, secondo una stima, beneficeranno di quel macchinario in un certo orizzonte temporale).

Parametri:

:: sia $M = \{1, \ldots, n\}$ l'insieme dei macchinari eventualmente da acquistare;

:: sia b il budget a disposizione per tali acquisti;

:: per $i = 1, \ldots, n$: sia c_i il *costo* del macchinario i, e sia u_i l'*utiltà* del macchinario i (intesa come il numero delle persone che, secondo una stima, beneficeranno di quel macchinario in un certo orizzonte temporale).

Variabili decisionali:

x_i per $i = 1, \ldots, n$;

$x_i = 1$ se il macchinario i è acquistato;

$x_i = 0$ altrimenti.

Modello:

$$\max \quad \sum_{i=1}^{n} u_i x_i$$

$$\sum_{i=1}^{n} c_i x_i \leq b$$

$$x_i \in \{0, 1\} \qquad \text{per } i = 1, \ldots, n$$

6. Ridefinizione della rete ospedaliera (tagli di risorse)

Il problema è ridefinire i reparti della rete ospedaliera di un Regione a seguito di tagli di risorse. Per semplicità, fissiamo un solo tipo di reparto, ad esempio il tipo di reparto di Ortopedia.

Parametri:

:: $\boldsymbol{R} = \{\boldsymbol{R_1, \ldots, R_n}\}$ = insieme dei reparti di Ortopedia della Regione;

:: $\boldsymbol{D} = \{\boldsymbol{D_1, \ldots, D_m}\}$ = insieme di distretti [indicativi] in cui è divisa la Regione;

:: per $i = 1, \ldots, n$: $\boldsymbol{r_i}$ = numero massimo di pazienti attesi che si stima sia possibile ricoverare nel reparto R_i in un certo orizzonte temporale;

:: per $j = 1, \ldots, m$: $\boldsymbol{d_j}$ = numero di pazienti (attesi) che si stima siano da ricoverare in uno fra i reparti in R (cioè in un reparto di Ortopedia) dal distretto D_j in un certo orizzonte temporale;

:: per $i = 1, \ldots, n$: $\boldsymbol{C_i}$ = "costo fisso" = costo stimato per la sola apertura del reparto R_i in un certo orizzonte temporale;

:: per $i = 1, \ldots, n$ e per $j = 1, \ldots, m$: $\boldsymbol{c_{ij}}$ = "costo marginale" = costo stimato per ricoverare nel reparto R_i un paziente del distretto D_j in un certo orizzonte temporale [tale costo può essere inteso come $c_i + q_{ij}$ dove: c_i è il costo stimato sostenuto dalla Regione per ricoverare nel reparto

R_i un generico paziente, e q_{ij} è il costo stimato sostenuto dal paziente del distretto D_j per essere ricoverato nel reparto R_i]

Problema: individuare quali reparti eventualmente chiudere fra $R_1, \ldots, R_n$ nel contesto di una gestione dei ricoveri che garantisca a ogni paziente il ricovero presso uno fra i reparti in R e che minimizzi la somma dei "costi fissi" e dei "costi marginali".

Variabili decisionali:
x_{ij} per $i = 1, \ldots, n$ e per $j = 1, \ldots, m$
y_i per $i = 1, \ldots, n$
dove:
x_{ij} = numero di pazienti (attesi) ricoverati nel reparto R_i dal distretto D_j
$y_i = 1$ se il reparto R_i è da non chiudere
$y_i = 0$ se il reparto R_i è da chiudere

Modello:

$$\min \quad \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij} x_{ij} + \sum_{i=1}^{n} C_i y_i$$

$$\sum_{i=1}^{n} x_{ij} \geq d_j \qquad \text{per } j = 1, \ldots, m$$

$$\sum_{j=1}^{m} x_{ij} \leq r_i \qquad \text{per } i = 1, \ldots, n$$

$$\sum_{j=1}^{m} x_{ij} \leq M\, y_i \quad \text{per } i = 1, \ldots, n \quad \text{(dove } M \text{ è uno scalare molto grande)}$$

$$x_{ij} \geq 0 \qquad \text{per } i = 1, \ldots, n \text{ e per } j = 1, \ldots, m$$

$$x_{ij} \geq \underline{\text{intero}} \qquad \text{per } i = 1, \ldots, n \text{ e per } j = 1, \ldots, m$$

$$y_i \in \{0, 1\} \qquad \text{per } i = 1, \ldots, n$$

7. Altro

Alcune altre possibili applicazioni sono le seguenti:
:: Gestione della risorsa sangue [a4],
:: Trattamenti di radioterapia [a3], [a5],
:: Gestione delle scorte (ad esempio di farmaci) [a10].

Ricordiamo infine che i modelli di Programmazione Lineare (Intera) sono molto adattabili a modifiche/varianti delle situazioni modellate.

Riferimenti

[a1] J.T. Blake, J. Donald, Mount Sinai hospital uses integer programming to allocate operating room time, *Interfaces*, 32 (2002), pp. 63-73

[a2] B. Cheang, H. Li, A. Lim, and B. Rodrigues, Nurse rostering problems-a bibliographic survey, *European Journal of Operational Research*, 151**,** 447-460 (2003).

[a3] D. Conforti, F. Guerriero, R. Guido, Non-block scheduling with priority for radiotherapy treatments**,** *European Journal of Operational Research*, 201 (1) (2010), pp. 289-296

[a4] V. De Angelis, N. Ricciardi, and G. Storchi, Optimizing blood assignment in a donation - transfusion system, *International Transactions in Operational Research,* August 2001

[a5] M. Ehrgott, R. Johnston, Optimisation of beam directions in intensity modulated radiation therapy planning, *OR Spectr* 25 (2003) 251–264

[a6] A. Guinet, S. Chaabane, Operating theatre planning, *International Journal of Production Economics*, 85 (2003), pp. 69-81

[a7] A. Mazier, X. Xie, M. Sarazin, Real-Time Patients Assignment: a Method for Improving Emergency Department Flow. In: Proceedings of the IEEE workshop on Health Care Management. Venetia (Italia), 2010

[a8] B. Satheeshkumar, S. Nareshkumar and S. Kumaraghuru, Linear programming applied to nurses shifting problems, *International journal of science and research*, 3 (2014) 171–173

[a9] www.unive.it/persone/pesenti

[a10] https://www.vkok.ee/logontrain/wp.../Riga-3-july-2014.pdf

2.1 Esercizi sul Capitolo 2

Gruppo P : *Formulare in termini di programmazione lineare (intera) i seguenti problemi; ognuno di essi è riconducibile in modo diretto a uno degli esempi base del gruppo A; in particolare le soluzioni sono riportate di seguito indicando solo l'esempio base di riferimento.*

P1. La Protezione Civile sta organizzando dei soccorsi da portare a 8 diverse zone, siano 1, 2, 3, 4, 5, 6, 7, 8, devastate da un cataclisma. Ci sono a disposizione 7 squadre di soccorso, siano A, B, C, D, E, F, G. Ogni squadra può soccorrere solo una zona. Inoltre, ogni squadra è adeguata solo per alcune zone, a seconda dei danni provocati dal cataclisma nelle varie zone. In particolare:
A è adeguata solo per 3, 5, 6, 8;
B solo per 3, 7;
C solo per 2, 3, 7;
D solo per 2, 4;
E solo per 1, 3, 4, 5, 6, 7;
F solo per 1, 4, 5;
G solo per 3, 8.
Il problema è soccorrere il maggior numero di zone in modo adeguato.
(Nota: tale problema è equivalente a massimizzare il numero di abbinamenti compatibili)

P2. Una multinazionale decide di aprire nuove filiali, avendo a disposizione un budget pari a 15. In particolare, c'è la possibilità di scegliere fra 7 potenziali filiali, siano A, B, C, D, E, F, G. Ognuna di tale potenziale filiale richiede un costo di avvio, come di seguito: $c_A = 7$, $c_B = 5$, $c_C = 3$, $c_D = 9$, $c_E = 4$, $c_F = 8$, $c_G = 10$. D'altro canto, si stima che ognuna di esse possa garantire dopo un anno un profitto, come di seguito: $p_A = 4$, $p_B = 5$, $p_C = 7$, $p_D = 8$, $p_E = 3$, $p_F = 5$, $p_G = 9$. Si intende che una filiale non può essere aperta parzialmente (cioè, o una filiale si apre oppure no). Il problema è scegliere quali filiali aprire in modo da massimizzare la somma dei profitti dopo un anno.

P3. Una manifattura che lavora tabacco desidera produrre due tipi di sigari: uno di qualità elevata (Originale), l'altro di qualità media (Antico). Il profitto che l'azienda presume di trarre dalla produzione di 1 unità di sigaro Originale è 10, mentre dalla produzione di 1 unità di sigaro Antico è 5. Tale produzione necessita di una particolare combinazione di due tipi di tabacco, diciamo di tipo A e di tipo B rispettivamente. Per produrre 1 unità di sigaro Originale, si ha bisogno di 7 unità di tabacco di tipo A, e di 3 unità di tabacco di tipo B. Per produrre 1 unità di sigaro Antico, si ha bisogno di 2 unità di tabacco di tipo A, e di 8 unità di tabacco di tipo B. Infine, la manifattura ha a disposizione 1000 unità di tabacco di tipo A e 2400 unità di tabacco di tipo B. Il problema è determinare le quantità di Originale e di Antico da produrre in modo da massimizzare il profitto totale.

P4. Quattro concessionarie, siano A, B, C, D, rimangono sprovviste di autovetture di un certo tipo (a causa di un blocco ferroviario). Si può ovviare a tale imprevisto rifornendo tali concessionarie da altre due concessionarie vicine, che hanno a disposizione (in giacenza) quel tipo di autovetture. Tali concessionarie hanno a disposizione rispettivamente 8 e 15 autovetture. Le concessionarie A, B, C, D hanno bisogno rispettivamente di 3, 4, 7, 2 autovetture. La consegna di ogni singola autovettura dal concessionario i = 1, 2 al concessionario j = A, B, C, D richiede un costo c_{ij}. In dettaglio: $c_{1A} = 4$; $c_{1B} = 4$; $c_{1C} = 5$; $c_{1D} = 5$; $c_{2A} = 2$; $c_{2B} = 7$; $c_{2C} = 5$; $c_{2D} = 7$. Il problema è pianificare le consegne ai concessionari A, B, C, D, in modo da minimizzare il costo totale.

P5. Il lavoro di una certa ditta consiste nel ricavare tre materie prime, siano A, B, C, decomponendo il materiale che essa estrae da due cave, siano cava 1 e cava 2. In particolare:
da 1 unità di materiale estratto dalla cava 1, la ditta ricava 5 u. di A, 3 u. di B, 0 u. di C;
da 1 unità di materiale estratto dalla cava 2, la ditta ricava 3 u. di A, 2 u. di B, 3 u. di C.
Il costo di estrazione di 1 unità di materiale dalla cava 1 è 30.
Il costo di estrazione di 1 unità di materiale dalla cava 2 è 40.
Il piano di produzione richiede che vengano ricavati almeno 70 u. di A, 40 u. di B, 20 u. di C.
Il problema è determinare il numero di unità di materiale da estrarre rispettivamente dalla cava 1 e dalla cava 2 in modo da minimizzare il costo totale (soddisfacendo il piano di produzione).

P6. Una persona riceve in eredità cinque ville. In generale pensa di venderle, ma vuole tenerne per sé un certo sottoinsieme S tale che: almeno una delle ville in S stia al mare; almeno una delle ville in S stia in un posto tranquillo; almeno una delle ville in S stia vicino a un ospedale; almeno una delle ville in S sia verde.
La situazione è la seguente:
Villa 1: non al mare; non in posto tranquillo; vicino ospedale; verde;
Villa 2: non al mare; in posto tranquillo; vicino ospedale; non verde;
Villa 3: al mare; non in posto tranquillo; non vicino ospedale; verde;
Villa 4: al mare; in posto tranquillo; non vicino ospedale; non verde.
Villa 5: al mare; non in posto tranquillo; vicino ospedale; non verde.
Il problema è massimizzare il numero delle ville da vendere.
(Nota: tale problema è equivalente a minimizzare il numero delle ville da tenere)

P7. Una TV privata vuole decidere il palinsesto delle sue trasmissioni per una delle prossime settimane, focalizzando in particolare sui programmi di prima serata. Ci sono a disposizione 10 programmi: quindi la TV privata vuole determinare, per ognuna delle 7 sere della settimana, quale programma mandare in onda (si intende che ogni programma può andare in onda solo una sera). Ogni programma i, se messo in onda, comporta un costo c_i, per $i = 1, \ldots, 10$ (per la realizzazione, per l'acquisto dall'esterno, …). In base a contatti già presi con vari sponsor, la TV privata stima che se il programma i andrà in onda la sera j allora otterrà (grazie agli sponsor) un ricavo r_{ij}, per $i = 1, \ldots, 10$ e per $j = 1, \ldots, 7$. Il problema è determinare, per ognuna delle 7 sere della settimana, quale programma mandare in onda in modo da massimizzare il profitto totale (dato da ricavo totale meno costo totale).

Soluzioni

P1. Riconducibile a “Matching”

P2. Riconducibile a “Riempimento”

P3. Riconducibile a “Combinando risorse”

P4. Riconducibile a “Trasporti”

P5. Riconducibile a “Decomponendo risorse”

P6. Riconducibile a “Covering”

P7. Riconducibile a “Assegnamento”.

Gruppo **Q** : *Formulare in termini di programmazione lineare (intera) i seguenti problemi; ognuno di essi è riconducibile in modo più o meno a uno degli esempi base del gruppo A, con l'introduzione di varianti, in particolare spesso della variante legata alla "grande M"; le soluzioni sono riportate di seguito.*

Q1. Un'azienda produce due prodotti, siano A e B. La produzione avviene mediante lavorazione sull'unica macchina M che l'azienda possiede: M può produrre solo 1 unità (di ogni prodotto) alla volta, e può lavorare al massimo 1500 ore in una settimana. Inoltre la produzione di ciascun prodotto richiede un certo quantitativo di prodotto grezzo P, che l'azienda possiede in quantità di 1200 unità.
Per produrre 1 unità di A c'è bisogno di 5 ore di lavorazione di M e di 4 unità di P. Per produrre 1 unità di B c'è bisogno di 4 ore di lavorazione di M e di 3 unità di P. Inoltre l'azienda: (i) riguardo A, vuole che se ne producano almeno 100 unità ma non più di 250 unità; (ii) riguardo B, vuole che, nel caso in cui se ne produca qualcosa, se ne producano almeno 100 unità. Il profitto che l'azienda trae dalla produzione di 1 unità di A è 25, mentre il profitto che trae dalla produzione di 1 unità di B è 15.
Il problema è determinare le unità di A e B da produrre in modo da massimizzare il totale dei profitti in una settimana.

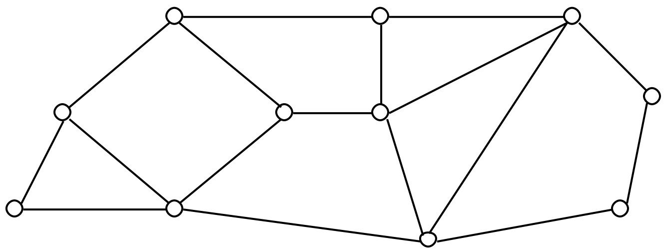
Q2. Il Comune di una città ha bandito 4 appalti, siano A, B, C, D (per rispettivi 4 lavori). In città ci sono 7 ditte, siano 1, 2, …, 7. Ogni ditta i, per $i = 1, 2, \ldots, 7$, presenta un preventivo di spesa s_{ij} per ogni appalto j, per j = A, B, C, D. Per motivi precauzionali/legislativi, il Comune non vuole/può assegnare più di 2 appalti a una stessa ditta.
Il problema è determinare a quale ditta assegnare ogni appalto, in modo da minimizzare il totale delle spese.

Q3. Una ditta farmaceutica produce vitamine B e C, estraendole da tre tipi di alimento, siano A1, A2, A3. Da 1 unità di A1, si estraggono 3 u. di B, e 2 u. di C. Da 1 unità di A2, si estraggono 2 u. di B, e 3 u. di C. Da 1 unità di A3, si estraggono 1 u. di B, e 5 u. di C. Tali tipi di alimento sono da acquistare. Acquistare A1 e A2 è semplice (basta recarsi vicino la ditta) e il loro costo unitario è rispettivamente 3 e 4. Acquistare A3 è complicato, nel senso che: il costo unitario è 2, però nel caso di acquisto (maggiore di 0 unità), bisogna acquistarne almeno 30 unità e bisogna pagare un costo fisso pari a 200 per il trasporto. La ditta deve produrre almeno 150 unità di vitamina B, e almeno 300 unità di vitamina C.
Il problema è determinare quante unità di A1, A2, A3 acquistare in modo da minimizzare il totale dei costi.

Q4. Un'agenzia di collocamento, riceve una mattina via email un insieme M di m richieste di lavoro, a fronte di un insieme N di n persone che sono iscritte per cercare un lavoro.

Ogni persona i, per $i = 1, \dots, n$, può essere abbinato solo a 1 lavoro nell'ambito di un certo sottoinsieme M_i di M ; d'altro canto ogni lavoro j, per $j = 1, \dots, m$, può essere abbinato solo a 1 persona nell'ambito di un certo sottoinsieme N_j di N.
Il problema è collocare più persone possibili, cioè, massimizzare il numero di accoppiamenti compatibili persona/lavoro (suggerimento: definire un insieme di coppie compatibili).

Q5. In un museo bisogna istallare delle telecamere in modo che tutti i corridoi siano visualizzati. Nel grafo $G = (V, E)$ sottostante [V è l'insieme dei nodi, E è l'insieme degli archi] ogni arco rappresenta un corridoio, e ogni nodo il punto di incrocio fra i corridoi che incidono su di esso. Istallando una telecamera su un nodo, si visualizzano tutti e soltanto gli archi che incidono su di esso.
Il problema è determinare i nodi su cui istallare le telecamere, garantendo tutti gli archi siano visualizzati, in modo da istallare il minor numero possibile di telecamere.
[come notazione, per ogni arco $e \in E$ si può indicare con $V(e)$ l'insieme formato dalla coppia di nodi che forma l'arco e; così, se si posiziona una telecamera su un nodo $v \in V$, allora si visualizzano tutti e soltanto gli archi e tali che $v \in V(e)$].



Q6. Una casa automobilistica ha 3 impianti di produzione, uno in Italia (I), uno in Polonia (P), uno in Slovenia (S). La casa automobilistica produce 4 tipi di autovetture, siano 1, 2, 3, 4. Ciascun impianto può produrre indifferentemente ognuno di questi tipi di autovetture. In generale si stima che ogni impianto (in un anno) possa produrre al più rispettivamente d_I, d_P, d_S, autovetture (indipendentemente dal tipo di autovetture). D'altro canto si stima che (in un anno) debbano essere prodotte rispettivamente r_1, r_2, r_3, r_4 autovetture di tipo 1, 2, 3, 4. Il costo per produrre in un impianto i (per i = I, P, S) 1 autovettura di tipo j (per $j = 1, 2, 3, 4$) è c_{ij}. Inoltre tenere aperto ognuno di questi impianti (in un anno) costa rispettivamente C_I, C_P, C_S [nota: un impianto è da tenere aperto solo se vi si produce almeno 1 autovettura di un qualunque tipo].
Il problema è determinare la produzione in ciascun stabilimento di ciascuna autovettura, in modo da minimizzare il totale dei costi (in un anno).

Q7. Una ditta ha la possibilità di effettuare degli investimenti, da scegliere fra n possibili investimenti. Ciascun investimento $i = 1, \dots, n$, è finanziabile in due anni, diciamo anno 1 e anno 2, cioè investire in i significa dover versare una somma a_{i1} all'inizio dell'anno 1 e una

somma a_{i2} all'inizio dell'anno 2: poi, alla fine dell'anno 2, l'investimento i garantisce un profitto p_i. La ditta può investire una somma b_1 all'inizio dell'anno 1, e una somma b_2 all'inizio dell'anno 2 (tali disponibilità non consentono di effettuare tutti gli investimenti). Ogni investimento non può essere effettuato parzialmente, cioè o si effettua oppure no.
Il problema è scegliere gli investimenti da effettuare in modo da massimizzare il profitto totale alla fine dell'anno 2.

Soluzioni

Q1.
Variabili decisionali:
x_A = quantità prodotta di A
x_B = quantità prodotta di B
y_B variabile binaria
y_B = 1 se si produce qualcosa di B (cioè se $x_B > 0$)
y_B = 0 altrimenti (cioè se $x_B = 0$)

$$\begin{aligned}
\max\ & 25x_A + 15x_B \\
& 5x_A + 4x_B \leq 1500 \\
& 4x_A + 3x_B \leq 1200 \\
& 100 \leq x_A \leq 250 \\
& x_B \leq My_B \quad \text{(dove M è uno scalare molto grande)} \\
& x_B \geq 100y_B \\
& x_A, x_B \geq 0 \\
& 0 \leq y_B \leq 1 \\
& y_B \ \underline{\text{intero}}
\end{aligned}$$

Q2.
Variabili decisionali:
x_{ij} per $i = 1, \ldots, 7$ e $j = 1, \ldots, 4$
x_{ij} = 1 se l'appalto i è assegnato alla ditta j
x_{ij} = 0 altrimenti

$$\begin{aligned}
\min\ & \sum_{i=1}^{7}\sum_{j=1}^{4} s_{ij}x_{ij} \\
& \sum_{i=1}^{7} x_{ij} = 1 \qquad \text{per } j = 1, \ldots, 4 \\
& \sum_{j=1}^{4} x_{ij} \leq 2 \qquad \text{per } i = 1, \ldots, 7 \\
& 0 \leq x_{ij} \leq 1 \qquad \text{per } i = 1, \ldots, 7 \text{ e per } j = 1, \ldots, 4 \\
& x_{ij} \ \underline{\text{intero}} \qquad \text{per } i = 1, \ldots, 7 \text{ e per } j = 1, \ldots, 4
\end{aligned}$$

Q3.

Variabili decisionali:

x_i = quantità di alimento Ai da acquistare, per i = 1, 2, 3

y_3 variabile binaria

y_3 = 1 se si acquista qualcosa di A3 (cioè se $x_3 > 0$)

y_3 = 0 altrimenti (cioè se $x_3 = 0$)

min $\quad 3x_1 + 4x_2 + 2x_3 + 200\, y_3$

$3x_1 + 2x_2 + \quad x_3 \geq 150$

$2x_1 + 3x_2 + \ 5x_3 \geq 300$

$x_3 \leq \mathrm{M}y_3$ (dove M è uno scalare molto grande)

$x_3 \geq 30y_3$

$x_1, x_2, x_3 \geq 0$

$0 \leq y_3 \leq 1$

y_3 <u>intero</u>

Q4.

Sia F l'insieme delle coppie (i, j) *compatibili*, cioè, $F = \{(i, j) : i \in P_j \text{ e } j \in M_i\}$;

Variabili decisionali:

x_{ij} per ogni coppia $(i, j) \in F$

x_{ij} = 1 se la persona i è collocata per lavoro j

x_{ij} = 0 altrimenti

max $\quad \sum_{(i,j)\in F} x_{ij}$

$\sum_{(i,j)\in F} x_{ij} \leq 1 \qquad$ per $i = 1, \ldots, n$

$\sum_{(i,j)\in F} x_{ij} \leq 1 \qquad$ per $j = 1, \ldots, m$

$0 \leq x_{ij} \leq 1 \qquad$ per ogni $(i, j) \in F$

x_{ij} <u>intero</u> $\qquad$ per ogni $(i, j) \in F$

Q5.

Variabili decisionali:

x_v per ogni nodo v di G.

x_v = 1 se sul nodo v è istallata una telecamera

x_i = 0 altrimenti.

min $\quad \sum_{v\in V} x_v$

$$\sum_{v \in V(e)} x_v \geq 1 \qquad \text{per ogni arco } e \text{ di } G$$

$0 \leq x_v \leq 1$ per ogni nodo v di G

x_v intero per ogni nodo v di G

Q6.

Variabili decisionali: x_{ij} per ogni i = I, P, S e j = 1, …, 4;

x_{ij} indica la quantità di autovetture di tipo j prodotte nello stabilimento i

y_i variabile binaria, per i = I, P, S

y_i = 1 se nello stabilimento i si produce qualcosa (cioè se $\sum_{j=1}^{4} x_{ij} > 0$)

y_i = 0 altrimenti (cioè se $\sum_{j=1}^{4} x_{ij} = 0$)

$$\min \quad \sum_{i=I,P,S} \sum_{j=1}^{4} c_{ij} x_{ij} + \sum_{i=I,P,S} C_i y_i$$

$$\sum_{j=1}^{4} x_{ij} \leq d_i \qquad \text{per } i = \text{I, P, S}$$

$$\sum_{i=I,P,S} x_{ij} \geq r_j \qquad \text{per } j = 1, \ldots, 4$$

$x_{ij} \geq 0$ per i = I, P, S e j = 1, …, 4;

x_{ij} intero per i = I, P, S e j = 1, …, 4;

$$\sum_{j=1}^{4} x_{ij} \leq \mathrm{M} y_i \qquad \text{per } i = \text{I, P, S} \quad \text{(dove M è uno scalare molto grande)}$$

$0 \leq y_i \leq 1$ per i = I, P, S

y_i intero per i = I, P, S

Q7.

Variabili decisionali:

x_i per i = 1, …, n

x_i = 1 se l'investimento i è effettuato

x_i = 0 altrimenti

$$\max \quad \sum_{i=1}^{n} p_i x_i$$

$$\sum_{i=1}^{n} a_{i1} x_i \leq b_1$$

$$\sum_{i=1}^{n} a_{i2} x_i \leq b_2$$

$0 \leq x_i \leq 1$ per i = 1, …, n $\qquad$ x_i intero per i = 1, …, n

Gruppo R : *I seguenti tre problemi, per i quali non sono riportate le soluzioni, sono legati a problemi reali (ad esempio, il problema dell'Assedio è legato al fatto che sembra che nella Seconda Guerra Mondiale gli Anglo-Americani abbiano usato la programmazione matematica, in particolare durante l'assedio navale subito dagli Inglesi da parte dei Tedeschi nella prima parte della guerra) e sono introdotti come possibile approfondimento.*

R1 *Assedio*

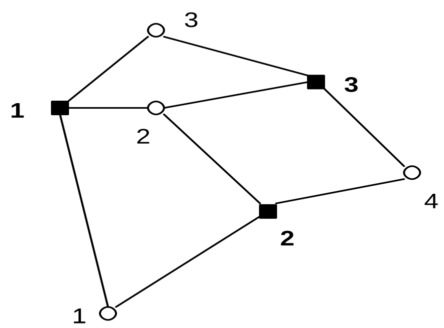
Durante una guerra una città di 25.000 abitanti viene posta sotto assedio dai suoi nemici. Ci si pone il problema di razionare la distribuzione di cibo. Le riserve alimentari sono composte da 3 alimenti, R, S, T, in quantità rispettivamente pari a 7.500.000, 5.000.000 e 12.000.000 unità. Si stima che per sopravvivere ogni persona abbia bisogno di 3 princìpi alimentari, diciamo A, B, C, in quantità giornaliera pari ad almeno 30, 25, 70 unità:

da 1 unità di R, si ottengono 2 u. di A, 1 u. di B, 10 u. di C;
da 1 unità di S, si ottengono 10 u. di A, 4 u. di B, 15 u. di C;
da 1 unità di T, si ottengono 7 u. di A, 12 u. di B, 0 u. di C.

Il problema è determinare una gestione ottimale di distribuzione degli alimenti in modo da garantire la sopravvivenza il maggior numero di giorni (in altri termini: il problema è calcolare quanti giorni al più la città può resistere garantendo una distribuzione di cibo necessaria per la sopravvivenza).

R2 (prima versione) *Magazzini*

In una rete di distribuzione (già attiva) di un certo bene, formata da magazzini che riforniscono negozi, si vuole studiare la possibilità di ridurre il numero dei magazzini così da confermarne (attivi) solo alcuni. Per vari motivi, non tutti i magazzini possono rifornire tutti i negozi. In figura, i magazzini sono quadrati neri, i negozi sono tondi bianchi, e la presenza di un arco indica la possibilità di rifornimento da un magazzino a un negozio.



Problema 1: Il problema è scegliere quali magazzini confermare, in modo da minimizzare il totale dei magazzini confermati, garantendo che ogni negozio possa rifornirsi.

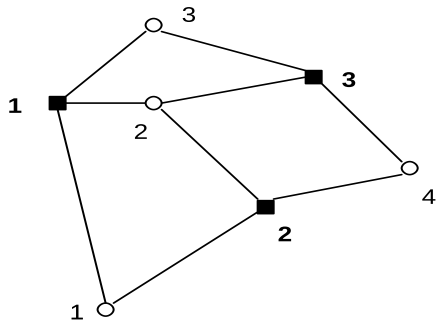
Variante del Problema 1: Nelle ipotesi del problema precedente, si assuma che:
:: confermare un magazzino i comporti un costo C_i.
Il problema è scegliere quali magazzini confermare, in modo da minimizzare il totale dei costi per la conferma dei magazzini, garantendo che ogni negozio possa rifornirsi.

Problema 2: Si assuma che:
:: confermare un magazzino i comporti un costo C_i;
:: ogni magazzino i abbia una capacità di rifornimento mensile pari a d_i;
:: ogni negozio j abbia una richiesta mensile pari a r_j.
Il problema è scegliere quali magazzini confermare, in modo da minimizzare il totale dei costi per la conferma dei magazzini, garantendo che ogni negozio possa rifornirsi e che sia rifornito mensilmente secondo la sua richiesta.

Variante del Problema 2: Nelle ipotesi del problema precedente, si assuma che:
:: il rifornimento mensile dal magazzino i al negozio j comporti un costo unitario pari a c_{ij}.
Il problema è scegliere quali magazzini confermare e pianificare i trasporti dai magazzini confermati ai negozi, in modo da minimizzare il totale dei costi per la conferma dei magazzini e dei costi per il trasporto su un orizzonte temporale di 40 mesi, garantendo che ogni negozio possa rifornirsi e che sia rifornito mensilmente secondo la sua richiesta.

R2 (seconda versione) *Aiuti umanitari*

Un'organizzazione umanitaria vuole attivare dei centri di smistamento, al fine di rifornire alcuni villaggi, avendo a disposizione dei potenziali centri. Per vari motivi, non tutti i potenziali centri possono rifornire tutti i villaggi. In figura, i potenziali centri sono quadrati neri, i villaggi sono tondi bianchi, e la presenza di arco indica la possibilità di rifornimento da un potenziale centro a un villaggio.



<u>Problema 1</u>: Il problema è scegliere quali centri attivare, in modo da minimizzare il totale dei centri attivati, garantendo che ogni villaggio possa rifornirsi.

<u>Variante del Problema 1:</u> Nelle ipotesi del problema precedente, si assuma che:
:: attivare un centro *i* comporti un costo C_i.
Il problema è scegliere quali centri attivare, in modo da minimizzare il totale dei costi per l'attivazione dei centri, garantendo che ogni villaggio possa rifornirsi.

<u>Problema 2</u>: Si assuma che:
:: attivare un centro *i* comporti un costo C_i;
:: ogni centro *i* abbia una capacità di rifornimento mensile pari a d_i;
:: ogni villaggio *j* abbia una richiesta mensile pari a r_j.
Il problema è scegliere quali centri attivare, in modo da minimizzare il totale dei costi per l'attivazione dei centri, garantendo che ogni villaggio possa rifornirsi e che sia rifornito mensilmente secondo la sua richiesta.

<u>Variante del Problema 2:</u> Nelle ipotesi del problema precedente, si assuma che:
:: il rifornimento mensile dal centro *i* al villaggio *j* comporti un costo unitario pari a c_{ij}.
Il problema è scegliere quali centri attivare e pianificare i trasporti dai centri attivati ai villaggi, in modo da minimizzare il totale dei costi per l'attivazione dei centri e dei costi per il trasporto su un orizzonte temporale di 40 mesi, garantendo che ogni villaggio possa rifornirsi e che sia rifornito mensilmente secondo la sua richiesta.

R3 *Spesa settimanale*

Una persona per fare la spesa frequenta una volta alla settimana uno dei seguenti tre supermarket a seconda di come capita: AAA, BBB, CCC. Questa settimana però la persona vuole operare in modo da minimizzare il costo complessivo per fare la spesa. La lista dei prodotti da comprare, con i relativi prezzi unitari per ogni supermarket, è riportata nella seguente tabella:

	AAA (prezzo unitario)	BBB (prezzo unitario)	CCC (prezzo unitario)
Pomodoro: 2 unità	2,80	3,00	3,50
Mozzarella: 4 unità	1,50	1,60	1,70
Pasta: 5 unità	0,80	0,80	0,90
Minestrone: 2 unità	2,50	3,20	3,80
Vino: 2 unità	4,50	4,30	4,20

Bisogna considerare anche i seguenti rispettivi costi fissi (che sintetizzano i costi in termine di tempo e di utilizzo eventuale dell'automobile): recarsi presso AAA comporta un costo di 4,00, recarsi presso BBB comporta un costo di 2,50, recarsi presso CCC comporta un costo di 0,50.

Il problema è come operare, cioè specificare cosa acquistare in ogni supermarket, al fine di minimizzare il costo complessivo per fare la spesa. □

3. Cenni sulla Programmazione Lineare (PL)

Un problema di *Programmazione Lineare,* in breve PL, è un problema di Programmazione Matematica

$$\mathbf{min}\{f(x) : x \in X\}$$

dove:

$X \subseteq \mathbb{R}^n$ è l'insieme delle soluzioni di un sistema di equazioni e/o disequazioni lineari;

$f : X \to \mathbb{R}$ è una funzione lineare.

Osservazione

Ogni problema di PL può essere rappresentato equivalentemente in uno dei seguenti due modi:

$\mathbf{min} \quad \mathbf{cx}$	$\mathbf{min} \quad \mathbf{cx}$
$\mathbf{Ax} \leq \mathbf{b}$	$\mathbf{Ax} = \mathbf{b}$
$\mathbf{x} \geq \mathbf{0}$	$\mathbf{x} \geq \mathbf{0}$
forma CANONICA	forma STANDARD

cioè ci si può sempre ricondurre o alla forma CANONICA [in cui, oltre ai *vincoli di non-negatività* $\mathbf{x} \geq \mathbf{0}$, ci sono solo disuguaglianze] oppure alla forma STANDARD [in cui, oltre ai *vincoli di non-negatività* $\mathbf{x} \geq \mathbf{0}$, ci sono solo uguaglianze].

Le formulazioni di sopra adottano una notazione vettoriale; in particolare, $\mathbf{x}$ è un vettore con n componenti, $\mathbf{c}$ è un vettore con n componenti, $\mathbf{A}$ è una matrice di dimensione $n \times m$, $\mathbf{b}$ è un vettore con m componenti; se ci sono dubbi, si può rivedere tale notazione più in dettaglio, ad esempio rivedendo come è definito il prodotto fra due vettori e come è definito il prodotto fra una matrice e un vettore (cfr. "notazione vettoriale per sistemi di equazioni lineari").

Al fine di mostrare che ogni problema di PL può essere rappresentato equivalentemente in uno di questi due modi (cioè comunque ricondotto ad essi), ci sono delle *regole di trasformazione*, riportate di seguito in modo minimale:

1) conversione da **max** a **min**

   infatti $\mathbf{max}\ \ \mathbf{wx} = -\ \mathbf{min}\ (-\ \mathbf{wx})$

2) conversione vincoli da “≤” a “≥”

   infatti $a_i\mathbf{x} \leq b_i \;\rightarrow\; a_i\mathbf{x} + s_i = b_i$, $s_i \geq 0$

   dove a_i è la i-sima riga di $\mathbf{A}$, b_i è la i-esima componente di $\mathbf{b}$, s_i è una nuova variabile

3) variabili x_i non vincolate in segno

   infatti $x_i \;\rightarrow\; = x^+_i - x^-_i$, $x^+_i \geq 0$, $x^-_i \geq 0$

   dove x_i è la i-sima componente di $\mathbf{x}$, mentre x^+_i e x^-_i sono nuove variabili □

A questo punto, per comodità, indichiamo un problema di PL come $\mathbf{min}\{\mathbf{cx} : \mathbf{x} \in P\}$, dove P è l’insieme delle soluzioni di un sistema di equazioni e/o disequazioni lineari.

***Definizione*:** Un problema di PL si dice

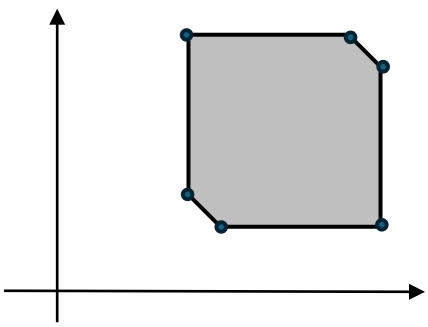
:: *illimitato* se $\mathbf{min}\{\mathbf{cx} : \mathbf{x} \in P\} = -\infty$

:: *impossibile* se $P = \emptyset$. □

La definizione di sopra descrive due casi limite.

***Definizione*:** L’insieme delle soluzioni di un sistema di equazioni e/o disequazioni lineari si dice *politopo*; un politipo limitato, cioè che può essere circoscritto, si dice *politopo*. □

***Definizione*:** Un punto di un poliedro P si dice *vertice* di P se non si trova su un segmento che ha per estremi 2 punti di P. □

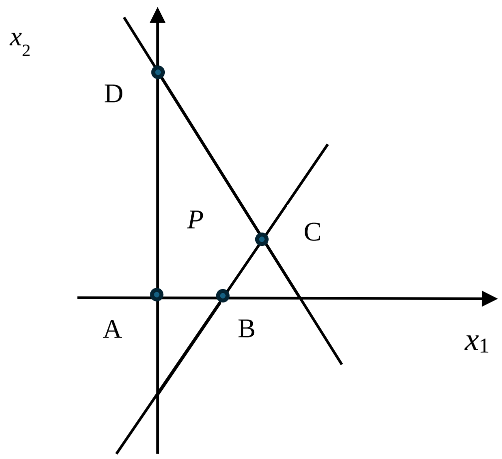


Nella figura di sopra è disegnato in grigio un poliedro in due dimensioni, nello specifico un politopo, evidenziando con dei pallini i suoi vertici; così i vertici, in accordo con la definizione, sono proprio quelli che intendiamo intuitivamente.

***Teorema* 3.1:** Dato un problema di PL $\mathbf{min}\{\mathbf{cx} : \mathbf{x} \in P\}$, se P è un politopo, allora esiste almeno una soluzione ottima coincidente con un vertice di P. □

Un esempio di applicazione : Consideriamo il seguente problema di PL.

$$\begin{array}{ll} \min & -x_1 - x_2 \\ & 6x_1 + 4x_2 \leq 24 \\ & 3x_1 - 2x_2 \leq 6 \\ & x_1, x_2 \geq 0 \end{array}$$



Dal Teorema 3.1 si può disegnare il seguente algoritmo per determinare una soluzione ottima: *lista tutti i vertici, calcola il valore della funzione per ognuno di essi, scegli il "migliore".*

A = (0, 0) → il valore della funzione in A è : – 0 – 0 = 0

B = (2, 0) → il valore della funzione in A è : – 2 – 0 = – 2

C = (3, 3/2) → il valore della funzione in A è : – 3 – 3/2 = – 9/2

D = (0, 6) → il valore della funzione in A è : – 0 – 6 = – 6

Allora, dal Teorema 1, si ha che D è una soluzione ottima del problema, cioè, $(x_1, x_2) = (0, 6)$ è una soluzione ottima del problema. □

Tuttavia in generale non è possibile riconoscere i vertici "graficamente" come nell'esempio di sopra. Così serve una *caratterizzazione algebrica dei vertici*, cioè, serve un risultato che permetta di riconoscere i vertici senza poter contare su una possibile rappresentazione grafica.

Si consideri un problema di PL in forma STANDARD:

$$\begin{aligned} \textbf{min} \quad & \mathbf{cx} \\ & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

con $P = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} = \mathbf{b} \text{ , } \mathbf{x} \geq \mathbf{0}\}$.

Assumiamo che: **A** è una matrice di dimensione $n \times m$ e che rango(**A**) = m; ciò implica che esiste una sottomatrice **B** di **A** con det(**B**) $\neq 0$.

Definizione**:** Una sottomatrice **B** di **A**, di dimensione $m \times m$ e con det(**B**) = 0, si dice *base* di **A**. Le variabili associate alle colonne di **B** si dicono *variabili in base* mentre le altre variabili si dicono *variabili fuori base*. □

Osservazione

Data una base **B** di **A**, se si "fissa" il valore delle variabili fuori base, allora il valore delle variabili in base è univocamente determinato. Infatti:

A può essere scritta come [**B**, **F**] dove **F** è la sottomatrice di **A** che completa **B** per ottenere **A**;

x può essere scritto come $[\mathbf{x_B}, \mathbf{x_F}]$ dove $\mathbf{x_B}$ è il (sotto)vettore delle variabili in base mentre $\mathbf{x_F}$ è il (sotto)vettore delle variabili fuori base;

allora il sistema $\mathbf{Ax} = \mathbf{b}$ può essere scritto come $\mathbf{Bx_B} + \mathbf{Fx_F} = \mathbf{b}$, da cui si ottiene $\mathbf{Bx_B} = \mathbf{b} - \mathbf{Fx_F}$, da cui si ottiene $\mathbf{Bx_B} = \mathbf{b'}$ (scrivendo $\mathbf{b' = b - Fx_F}$ dato che il valore delle variabili fuori base è fissato) che è un sistema nelle variabili in $\mathbf{x_B}$ con **B** matrice quadrata con det(**B**) $\neq 0$, cioè, è un sistema che secondo il Teorema di Cramer ammette soluzione e tale soluzione è unica. □

Definizione**:** Data una base **B** di **A**, la soluzione che si ottiene "fissando" il valore delle variabili fuori base come $\mathbf{x_F} = \mathbf{0}$ (cioè fissando il valore di ogni variabile fuori base uguale a 0) si dice *soluzione base associata alla base* **B**. Essa è $[\mathbf{x_B}, \mathbf{x_F}] = [\mathbf{B^{-1}b}, \mathbf{0}]$. In particolare tale soluzione è *ammissibile* se $[\mathbf{x_B}, \mathbf{x_F}] \geq \mathbf{0}$, cioè, se $\mathbf{B^{-1}b} \geq \mathbf{0}$. □

***Teorema* 3.2:** Un punto **x** del poliedro $P = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} = \mathbf{b}\ ,\ \mathbf{x} \geq \mathbf{0}\}$ è vertice di P se e solo se **x** è una soluzione base associata a una base **B** di **A**. □

Così il Teorema 3.2 è la *caratterizzazione algebrica dei vertici* che era auspicata.
In particolare, dal Teorema 1 e dal Teorema 2, si ottiene il seguente corollario:

***Corollario* 3.3:** Dato un problema di PL $\mathbf{min}\{\mathbf{cx} : x \in P\}$ con $P = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} = \mathbf{b}\ ,\ \mathbf{x} \geq \mathbf{0}\}$, se P è un politopo, allora esiste almeno una soluzione ottima coincidente con una soluzione base associata a una base **B** di **A**. □

<u>Il metodo del Simplesso</u>

Il metodo del Simplesso è un metodo per risolvere un problema di PL.
L'idea è sempre quella di "visitare" solamente i vertici (per poi calcolare il corrispondente valore della finzione obiettivo e per poi scegliere il valore migliore).
Allo stesso tempo, si vorrebbe visitare il numero minore possibile di vertici, dato che essi sono dell'ordine $\binom{n}{m} = \frac{n!}{m!(n-m)!}$ cioè quanto sono le possibili basi in accordo con il Teorema 3.2.
A tale fine il metodo del Simplesso adopera i seguenti "utensili":

--- CRITERIO DI OTTIMALITA': permette di riconoscere una soluzione ottima se visitata;
--- SPOSTAMENTO MIGLIORATIVO: permette di spostarsi da una soluzione ad un'altra soluzione (cioè da un vertice a un altro vertice) che sia non peggiore.

Quanto sopra è il nucleo del metodo del Simplesso. Tale metodo è stato ideato da George Dantzig nel 1947. Ci sono alcuni aspetti e dettagli che andrebbero discussi [ad esempio, come si fa ad evitare che tale metodo visiti ciclicamente uno stesso gruppo di vertici, ognuno non peggiore dell'altro] e per i quali si può fare riferimento al libro [Fischetti].

Concludiamo comunque evidenziando il seguente aspetto che riguarda la complessità computazionale nel caso peggiore. Il metodo del Simplesso ha una complessità computazionale nel caso peggiore che è ritenuta inefficiente, cioè, non polinomiale; tuttavia il metodo del Simplesso è considerato efficiente in pratica; riguardo questo contrasto, è stato dimostrato che è molto improbabile che il metodo del Simplesso sia inefficiente in pratica, pur essendo appunto inefficiente dal punta di vista teorico (ciò mi è stato spiegato dal Prof. Luca Moscardelli il quale ha indicato in particolare il seguente link https://en.wikipedia.org/wiki/Smoothed_analysis). D'altro canto esiste un altro metodo per risolvere i problemi di PL, detto "metodo dell'Ellissoide", che sotto ipotesi non ristrettive ha una complessità computazionale nel caso peggiore che è ritenuta efficiente; tuttavia il "metodo dell'Ellissoide" sembra essere ostico nel suo utilizzo pratico, cioè, è considerato inefficiente in pratica. Così, in finale, che il metodo del Simplesso è comunque l'algoritmo comunemente usato per risolvere problemi di PL.

3.1 Dualità in PL

La Dualità è una legge che assegna a ogni problema di PL, sia P detto *primale*, un altro problema di PL, sia D detto *duale* [di P]. Al fine di vedere come è definita la Dualità, ricordando che ogni problema di PL è riconducibile sia a un problema di PL in forma CANONICA sia a un problema di PL in forma STANDARD, vediamo solo come è definita la Dualità in questi due casi.

- forma CANONICA

$\min \ \mathbf{cx}$	$\max \ \mathbf{ub}$
$\mathbf{Ax} \geq \mathbf{b}$	$\mathbf{A}^T\mathbf{u} \leq \mathbf{c}$
$\mathbf{x} \geq \mathbf{0}$	$\mathbf{u} \geq \mathbf{0}$
primale	*duale*

- forma STANDARD

$\min \ \mathbf{cx}$	$\max \ \mathbf{ub}$
$\mathbf{Ax} = \mathbf{b}$	$\mathbf{A}^T\mathbf{u} \leq \mathbf{c}$
$\mathbf{x} \geq \mathbf{0}$	
primale	*duale*

Vediamo solo un esempio, nel caso della forma CANONICA, di seguito:

$\min \ 5x_1 + 7x_2$	$\max \ 10u_1 + 40u_2$
$4x_1 + x_2 \leq 10$	$4u_1 + 2u_2 \leq 5$
$2x_1 + 3x_2 \leq 40$	$u_1 + 3u_2 \leq 7$
$x_1, x_2 \geq 0$	$u_1, u_2 \geq 0$
primale	*duale*

Si può osservare, intendendo per vincoli solamente quelli diversi dai vincoli di non-negatività (cioè quelli diversi dai vincoli $x_1, x_2 \geq 0$ e $u_1, u_2 \geq 0$), che: i coefficienti della funzione obiettivo del primale corrispondono ai termini noti del duale, e viceversa; a ogni vincolo del primale corrisponde una variabili del duale (in dettaglio, al vincolo $4x_1 + x_2 \leq 10$ corrisponde la variabile u_1, al vincolo $2x_1 + 3x_2 \leq 40$ corrisponde la variabile u_2), e viceversa; la matrice dei vincoli del duale è la trasposta della matrice dei vincoli del primale.

Proprietà fondamentali della Dualità

***Proposizione* 3.4:** Siano P un primale e sia D il duale di P: allora il duale di D è P. □

In altri termini, se la Dualità associa a un problema P un problema D, allora la Dualità associa al problema D il problema P. Ancora, in altri termini, si ha che reiterando la Dualità a un problema P si riottiene il problema P.

***Teorema* 3.5 (*Dualità forte*):** Consideriamo una coppia di problemi primale duale:

$\min \ \mathbf{cx}$	$\max \ \mathbf{ub}$
$\mathbf{Ax} \geq \mathbf{b}$	$\mathbf{A^T u} \leq \mathbf{c}$
$\mathbf{x} \geq \mathbf{0}$	$\mathbf{u} \geq \mathbf{0}$
primale	*duale*

Se il primale ammette soluzione, allora si ha

$\min\{\mathbf{cx} : \mathbf{Ax} = \mathbf{b} , \mathbf{x} \geq \mathbf{0}\} = \max\{\mathbf{ub} : \mathbf{A^T u} \geq \mathbf{b} , \mathbf{u} \geq \mathbf{0}\}$. □

***Teorema* 3.6 (*Dualità debole*):** Consideriamo una coppia di problemi primale duale:

$\min \ \mathbf{cx}$	$\max \ \mathbf{ub}$
$\mathbf{Ax} \geq \mathbf{b}$	$\mathbf{A^T u} \leq \mathbf{c}$
$\mathbf{x} \geq \mathbf{0}$	$\mathbf{u} \geq \mathbf{0}$
. *primale*	*duale*

Se sia il primale sia il duale ammettono soluzione, allora per ogni coppia (**x', u'**) dove **x'** è una soluzione ammissibile del primale e **u'** è una soluzione ammissibile del duale si ha $\mathbf{u'b} \leq \mathbf{cx'}$ □

***Corollario* 3.7:** Consideriamo una coppia di problemi primale duale:

$\min \ \mathbf{cx}$	$\max \ \mathbf{ub}$
$\mathbf{Ax} \geq \mathbf{b}$	$\mathbf{A^T u} \leq \mathbf{c}$
$\mathbf{x} \geq \mathbf{0}$	$\mathbf{u} \geq \mathbf{0}$
primale	*duale*

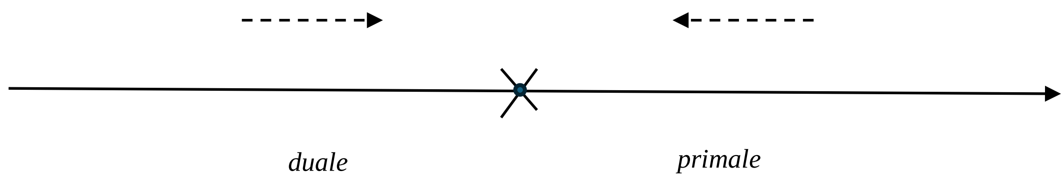
Allora sono possibili solo 4 casi:

1) entrambi i problemi ammettono soluzione e si ha

$\mathbf{min}\{\mathbf{cx} : \mathbf{Ax} = \mathbf{b} , \mathbf{x} \geq \mathbf{0}\} = \mathbf{max}\{\mathbf{ub} : \mathbf{A^Tu} \geq \mathbf{b} , \mathbf{u} \geq \mathbf{0}\}$;

2) il problema primale è illimitato e il problema duale è impossibile;
3) il problema primale è impossibile e il problema duale è illimitato;
4) entrambi i problemi sono impossibili. □

Si noti che la formula $\mathbf{min}\{\mathbf{cx} : \mathbf{Ax} = \mathbf{b} , \mathbf{x} \geq \mathbf{0}\} = \mathbf{max}\{\mathbf{ub} : \mathbf{A^Tu} \geq \mathbf{b} , \mathbf{u} \geq \mathbf{0}\}$ indica che i due rispettivi valori delle due rispettive soluzioni ottime sono uguali, cioè ad esempio $\mathbf{min}\{\mathbf{cx} : \mathbf{Ax} = \mathbf{b} , \mathbf{x} \geq \mathbf{0}\} = 70$ se e solo se $\mathbf{max}\{\mathbf{ub} : \mathbf{A^Tu} \geq \mathbf{b} , \mathbf{u} \geq \mathbf{0}\} = 70$, mentre lascia le due rispettive soluzioni ottime non in relazione fra di loro. Nella figura di seguito, sempre con riferimento al caso (1) del Corollario 3.7, si può visualizzare, con riferimento al valore diciamo X sulla retta dei numeri reali, come: i valori delle soluzioni ammissibili del problema *duale* stanno alla sinistra di X, i valori delle soluzioni ammissibili del problema *primale* stanno alla destra di X, i valori delle due rispettive soluzioni ottime (del problema duale e del problema primale) coincidono in X.



<u>Utilità della Dualità in PL</u>

L'utilità della Dualità in PL sembra essere molteplice e per alcuni versi inaspettata.
Di seguito accenniamo solo a tre possibili utilità.
La prima utilità è legata al Teorema della Dualità forte: supponiamo che un decisore vuole risolvere il problema primale P; se è difficile risolvere P, allora [in base a tale teorema] si può provare a risolvere il problema duale D; in particolare, considerando la legge delle Dualità, il problema duale D potrebbe avere ad esempio meno variabili del problema primale P.
La seconda utilità è legata al Teorema della Dualità debole: supponiamo che un decisore vuole risolvere il problema primale P; se è difficile risolvere P, allora il decisore può sia ottenere un *upper bound* della soluzione ottima del problema P calcolando il valore di una qualsiasi soluzione ammissibile del problema duale P, sia ottenere un *lower bound* della soluzione ottima

del problema P calcolando il valore di una qualsiasi soluzione ammissibile del problema duale D [in base a tale teorema].

La terza utilità è nota come "interpretazione economica della Dualità"; in termini di applicazione, quando modelliamo un problema reale in termini di PL ottenendo un problema diciamo P, allora ogni variabile [decisionale] di P ha un significato; l' "interpretazione economica della Dualità" evidenzia la possibilità di dare un significato anche alle variabili del problema duale D di P, o più precisamente, di dare un significato al valore che ognuna delle del problema duale D di P assume nella soluzione ottima.

Esempi sull'interpretazione economica della Dualità

1) *Componendo risorse*

Un'azienda vinicola desidera produrre due tipi di vino: uno da tavola, uno da dessert. Il profitto che l'azienda trae dalla produzione di 1 unità di vino da tavola è 3, mentre dalla produzione di 1 unità di vino da dessert è 7. Tale produzione necessita di una particolare combinazione di due tipi di uva: diciamo di tipo A e di tipo B rispettivamente. Per produrre 1 unità di vino da tavola, si ha bisogno di 3 unità di uva di tipo A e di 2 unità di uva di tipo B. Per produrre 1 unità di vino da dessert, si ha bisogno di 1 unità di uva di tipo A e di 4 unità di uva di tipo B. Infine l'azienda ha a disposizione 1000 unità di uva di tipo A e 400 unità di uva di tipo B. Il problema è determinare le quantità di vino da tavola e da dessert da produrre in modo da massimizzare il profitto totale.

Variabili decisionali: x_1, x_2

x_1 = quantità di vino da tavola prodotta

x_2 = quantità di vino da dessert prodotta

$$
(1) \qquad \max \quad 3x_1 + 7x_2
$$
$$
\begin{aligned}
3x_1 + x_2 &\leq 1000 \\
2x_1 + 4x_2 &\leq 400 \\
x_1, x_2 &\geq 0
\end{aligned}
$$

La soluzione ottima è $x^*_1 = 0$, $x^*_2 = 100$, con valore 700.

Si consideri di seguito il problema duale del problema (1).

$$(2) \qquad \min \quad 1000\, y_1 + 400\, y_2$$
$$3y_1 + 2y_2 \geq 3$$
$$y_1 + 4y_2 \geq 7$$
$$y_1, y_2 \geq 0$$

La soluzione ottima è $y^*_1 = 0$, $y^*_2 = 7/4$, con valore 700.

Sia y_j^* una componente della soluzione ottima del problema duale. Ricordando che a ogni vincolo del problema primale è associato una variabile del problema duale, sia b_j il termine noto del vincolo del problema primale associato alla variabile y_j, cioè la disponibilità della risorsa j. In particolare b_jy_j è un termine della funzione obiettivo del problema duale: all'ottimo esso vale $b_jy^*_j$.
Ora si assuma di poter far variare b_j : sia $b_j + \Delta b_j$.

- Se $\Delta b_j > 0$ (che significa un <u>acquisto</u> della risorsa j), allora per il Teorema della Dualità Forte il profitto ottimo varia di $y^*_j\Delta b_j \geq 0$ (nel caso di acquisto gratuito); ma in generale varia di $y^*_j\Delta b_j - y'_j\Delta b_j$, dove y'_j è il prezzo unitario di acquisto della risorsa j; perciò tale acquisto è conveniente solo se $y^*_j\Delta b_j - y'_j\Delta b_j > 0$, cioè solo se $y'_j < y^*_j$.
- Se $\Delta b_j < 0$ (che significa una <u>vendita</u> della risorsa j), allora per il Teorema della Dualità Forte il profitto ottimo varia di $y^*_j\Delta b_j \leq 0$ (nel caso di vendita gratuita); ma in generale varia di $y^*_j\Delta b_j + y'_j|\Delta b_j| = y^*_j\Delta b_j - y'_j\Delta b_j$, dove y'_j è il prezzo unitario di vendita della risorsa j; perciò tale vendita è conveniente solo se $y^*_j\Delta b_j - y'_j\Delta b_j > 0$, cioè solo se $y'_j > y^*_j$ (essendo $\Delta b_j < 0$).

Quindi la soluzione ottima del problema duale $(y^*_1, \ldots, y^*_m)$ rappresenta il valore intrinseco di ogni risorsa j, per $j = 1, \ldots, m$. Tale valore è detto anche *prezzo ombra*, nel senso che:

- è conveniente <u>acquistare</u> la risorsa j (entro un certo limite di quantità), se la si acquista a un prezzo unitario che è <u>al di sotto del prezzo ombra y_j^*</u>.
- è conveniente <u>vendere</u> la risorsa j (entro un certo limite di quantità), se la si vende a un prezzo unitario che è <u>al di sopra del prezzo ombra y_j^*</u>.

<u>Osservazione</u>: i limiti di quantità entro i quali conviene acquistare/vendere una risorsa dipendono dal fatto che la soluzione ottima di base può cambiare per variazioni della disponibilità della risorsa, cioè del termine noto corrispondente; tali limiti sono calcolabili mediante semplici operazioni riportate

nell'analisi di sensitività (o di post-ottimalità) e comunque sono calcolati dai software che risolvono i problemi di PL.

Tornando al nostro esempio si ha che:
y^*_1 = prezzo ombra dell'uva A
y^*_2 = prezzo ombra dell'uva B

Consideriamo ad esempio il prezzo ombra $y^*_2 = 7/4$ dell'uva di tipo B.
E' conveniente acquistare tale risorsa (entro un certo limite di quantità) ad un prezzo unitario inferiore a 7/4, dato che ciò garantisce un aumento del profitto totale.
E' conveniente vendere tale risorsa (entro un certo limite di quantità) ad un prezzo unitario superiore a 7/4, dato che ciò garantisce un aumento del profitto totale.
E' indifferente acquistare o vendere tale risorsa (entro un certo limite di quantità) ad un prezzo unitario pari a 7/4, dato che ciò non cambia il profitto totale.

2) *Decomponendo risorse*

Una persona vuole fare una dieta. In particolare deve assumere due tipi di sostanze, cioè proteine e vitamine, che può ricavare comprando tre tipi di alimenti, cioè frutta, latte, uova. In dettaglio:
1 unità di frutta contiene: 0 u. di proteine, 7 u. di vitamine;
1 unità di latte contiene: 2 u. di proteine, 3 u. di vitamine;
1 unità di uova contiene: 5 u. di proteine, 1 u. di vitamine.
Il costo unitario della frutta, del latte e delle uova è rispettivamente di 10, 20, 10.
La dieta richiede di assumere almeno 15 unità di proteine e 25 unità di vitamine.
Il problema è determinare le quantità di frutta, latte, uova da acquistare, soddisfacendo la richiesta di sopra, in modo da minimizzare il costo totale.

Variabili decisionali: x_F, x_L, x_U
x_F = quantità di frutta da acquistare
x_L = quantità di latte da acquistare
x_U = quantità di uova da acquistare

$$(1) \qquad \min \quad 10x_F + 20x_L + 10x_U$$
$$0x_F + 2x_L + 5x_U \geq 15$$

$$7x_F + 3x_L + 1x_U \geq 25$$
$$x_F, x_L, x_U \geq 0$$

La soluzione ottima è $x^*_F = 3{,}14$, $x^*_L = 0$, $x^*_U = 3$, con valore 61,42.

Si consideri di seguito il problema duale del problema (1).

(2) max $15\, y_1 + 25\, y_2$

$$0y_1 + 7y_2 \leq 10$$
$$2y_1 + 3y_2 \leq 20$$
$$5y_1 + 1y_2 \leq 10$$
$$y_1, y_2 \geq 0$$

La soluzione ottima è $y^*_1 = 1{,}71$ e $y^*_2 = 1{,}42$, con valore 61,42.

Con un argomentazione simile a quella dell'Esempio 1, si può verificare che la soluzione ottima del problema duale ($y^*_1, \ldots, y^*_m$) rappresenta il valore intrinseco y_j^* di ogni produzione j, per $j = 1, \ldots, m$ (proteine, vitamine). Tale valore è detto anche *prezzo ombra*, nel senso che:

• è conveniente <u>acquistare</u> la produzione j (entro un certo limite di quantità), se la si acquista a un prezzo unitario che è <u>al di sotto del prezzo ombra y_j^*.</u>

<u>Osservazione</u>: i limiti di quantità entro i quali conviene acquistare una produzione dipendono dal fatto che la soluzione ottima di base può cambiare per variazioni della richiesta della produzione, cioè del termine noto corrispondente; tali limiti sono calcolabili mediante semplici operazioni riportate nell'analisi di sensitività (o di post-ottimalità) e comunque sono calcolati dai software che risolvono i problemi di PL.

Tornando al nostro esempio si ha che:
y^*_1 = prezzo ombra delle proteine
y^*_2 = prezzo ombra delle vitamine

Consideriamo ad esempio il prezzo ombra $y^*_2 = 1{,}42$ delle vitamine.

E’ conveniente acquistare tale produzione – cioè vitamine, ad esempio sotto forma di integratore alimentare – (entro certi limiti di quantità) ad un prezzo unitario inferiore a 1,42, dato che ciò garantisce una diminuzione della spesa totale.
E’ indifferente acquistare tale produzione (entro un certo limite di quantità) ad un prezzo unitario pari a 1,42, dato che ciò non cambia la spesa totale. □

4. Cenni sulla Programmazione Lineare Intera (PLI)

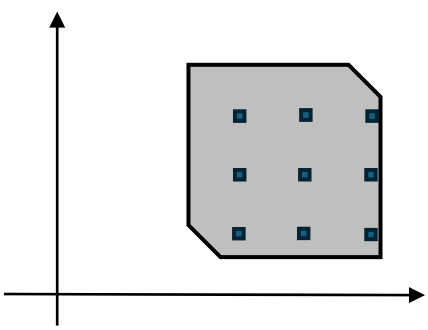
Un problema di Programmazione Lineare Intera, in breve di PLI, può essere inteso come una variante di un problema di PL dove in aggiunta tutte le variabili o solo alcune variabili sono vincolate ad assumere valori interi.

Un problema di PLI, in forma CANONICA, è definito come segue:

$$
\begin{aligned}
Z_{PLI} = \quad & \mathbf{min} \ \ \mathbf{cx} \\
& \mathbf{Ax} \geq \mathbf{b} \\
& \mathbf{x} \geq \mathbf{0} \\
& \mathbf{x} \ \ \text{INTERO}
\end{aligned}
$$

dove Z_{PLI} rappresenta il valore della soluzione ottima del problema (cioè il valore che la funzione obiettivo assume in corrispondenza della soluzione ottima del problema).
Nel caso in cui solo alcune variabili sono vincolate ad assumere valori interi si parla di Programmazione Lineare Intera Mista.

Denotiamo con **P** il problema di PLI formulato come sopra; in particolare diciamo che
$P = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} \geq \mathbf{b} \ , \ \mathbf{x} \geq \mathbf{0}\}$ è il *poliedro associato al problema* **P**
con formulazione dipendente dal poliedro P.
Allora (in termini di visualizzazione grafica) l'insieme delle soluzioni ammissibili del problema **P**, con formulazione dipendente dal poliedro P, è dato da tutti i punti di P che sono interi, cioè, è dato da $P \cap \mathbb{Z}^n$. Nella figura di sotto, P è il poliedro in grigio, mentre $P \cap \mathbb{Z}^n$ è l'insieme dei quadratini cioè delle soluzioni intere all'interno di P.



Allora si può osservare che il problema **P** può ammettere diverse formulazioni dipendenti da diversi poliedri, cioè, da qualsiasi altro poliedro P' tale che $P \cap \mathbb{Z}^n = P' \cap \mathbb{Z}^n$: denotiamo l'insieme di tali poliedri come l'*insieme dei poliedri associati al problema* **P**.
Nel seguito consideriamo problemi di PLI i cui poliedri associati sono dei politopi, cioè, problemi di PLI che ammettono (in caso) una soluzione ottima finita.

Un primo approccio

Un primo approccio per tentare di risolvere un problema **P** di PLI, come definito sopra, è il seguente procedimento: eliminare i vincoli di interezza e poi risolvere il problema di PL così ottenuto – detto *rilassamento continuo* del problema originale e riportato di seguito:

$$
\begin{aligned}
Z_{PL} = \quad \min \quad & \mathbf{cx} \\
& \mathbf{Ax} \geq \mathbf{b} \\
& \mathbf{x} \geq \mathbf{0}
\end{aligned}
$$

dove Z_{PL} rappresenta il valore della soluzione ottima del problema (cioè il valore che la funzione obiettivo assume in corrispondenza della soluzione ottima del problema).

Osservazione
La relazione fra il problema **P** di PLI e il suo rilassamento continuo è in questi due punti:
- $Z_{PL} \leq Z_{PLI}$;

infatti, ricordando la notazione $P = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} \geq \mathbf{b} \text{ , } \mathbf{x} \geq \mathbf{0}\}$, si ha che $P \supseteq P \cap \mathbb{Z}^n$,
così si ottiene $Z_{PL} = \mathbf{min}\{\mathbf{cx} : \mathbf{x} \in P\} \leq \mathbf{min}\{\mathbf{cx} : \mathbf{x} \in P \cap \mathbb{Z}^n\} = Z_{PLI}$.
- se la soluzione ottima sia $\mathbf{x^*}$ del rilassamento continuo di **P** è intera (cioè ha tutte le componenti intere), allora $\mathbf{x^*}$ è soluzione ottima anche del problema **P** , cioè $\mathbf{cx^*} = Z_{PLI}$;

infatti, da un lato si ha $\mathbf{cx^*} = Z_{PL} \leq Z_{PLI}$, da un altro lato si ha $\mathbf{cx^*} \geq Z_{PLI}$ essendo $\mathbf{x^*} \in P \cap \mathbb{Z}^n$. □

Tale procedimento per tentare di risolvere un problema **P** di PLI, come definito sopra, non può garantire di trovare una soluzione ottima per il problema **P**; in particolare, i vertici di un poliedro P associato al problema **P** possono essere anche non-interi, così che la soluzione ottima del rilassamento continuo al problema **P** può essere anche non-intera [ricordando i risultati introdotti per la Programmazione Lineare]. Comunque vale il seguente risultato teorico:

***Teorema* 4.1**: Dato un problema **P** di PLI, come definito sopra, esiste un politopo $P' = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A'x} \geq \mathbf{b'} \text{ , } \mathbf{x} \geq \mathbf{0}\}$ associato al problema **P** tale che:
:: P' ha tutti i vertici interi;

:: **min**$\{\mathbf{cx} : \mathbf{x} \in P\}$ = **min**$\{\mathbf{cx} : \mathbf{x} \in P'\}$ per ogni vettore **c** (di numeri razionali). □

In altri termini il Teorema 4.1 afferma che, in teoria, esiste un politopo associato al problema **P** di PLI che ha tutti i vertici interi – diciamo un politopo associato *ideale*. Questo implica che, in teoria, ogni problema di PLI è equivalente a un problema di PL. Tuttavia, in partica, tale politopo associato *ideale* può essere complicato da determinare e/o può definire un sistema enorme di disequazioni lineari.

Totale unimodularità

Questa sezione introduce alcuni risultati che permettono di individuare alcuni problemi di PLI per i quali il politopo associato *ideale* è facilmente determinabile.
Notazione: data una matrice **A**, di dimensione $n \times m$, indichiamo con $\{a_{ij}\}$ (per $i = 1, \ldots, n$ e per $j = 1, \ldots, m$) gli elementi di **A** e diciamo che **A** è *intera* se tutti gli elementi di **A** sono interi; inoltre ricordiamo che un matrice **A**, di dimensione $n \times m$, è detta *quadrata* se $n = m$.

Definizione: Una matrice intera **A**, di dimensione $n \times m$, si dice *totalmente unimodulare* (in breve, *TUM*) se $\det(\mathbf{Q}) \in \{-1, 0, 1\}$ per ogni sottomatrice quadrata **Q** di **A**. □

Si noti che, per definizione, se **A** è una matrice TUM allora per ogni elemento a_{ij} di **A** si ha che $a_{ij} \in \{-1, 0, 1\}$; infatti ogni elemento di **A** è una sottomatrice quadrata di dimensione 1×1.

***Teorema* 4.2:** Sia **A** una matrice TUM di dimensione $n \times m$ e sia **b** un vettore intero con m componenti. Allora sia il poliedro $P^{=} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} = \mathbf{b} , \mathbf{x} \geq \mathbf{0}\}$ sia il poliedro $P^{\geq} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} \geq \mathbf{b} , \mathbf{x} \geq \mathbf{0}\}$ hanno tutti i vertici interi. □

Allora il teorema di sopra implica il seguente corollario di carattere generale.

***Corollario* 4.3:** Dato un problema **P** di PLI come segue

$$\begin{aligned} \mathbf{min}\ \ & \mathbf{cx} \\ & \mathbf{Ax} \geq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \\ & \mathbf{x}\ \text{INTERO} \end{aligned}$$

se **A** è una matrice TUM e se **b** è un vettore intero, allora il vincolo **x** INTERO può essere eliminato, cioè il problema **P** di PLI può essere risolto come un problema di PL risolvendo il rilassamento continuo del problema **P**. □

Allora il corollario di sopra implica il seguente corollario di natura applicativa, con riferimento ad alcuni problemi che abbiamo introdotto come problemi di PLI, evidenziando così che tali problemi sono in finale *facili* in accordo con Appendice 1.

***Corollario* 4.4:** I seguenti problemi possono essere risolti come problemi di PL:
Assegnamento,
Matching,
Trasporti,
Cammino di costo minimo,
Massimo flusso. □

La dimostrazione del Corollario 4.4 la vedremo, tramite esempi, dopo aver introdotto alcune proprietà delle matrici TUM.

Ora torniamo al Corollario 4.3, il quale evidenzia che – in accordo con Appendice 1 – può valere la pena studiare come riconoscere le matrici TUM; infatti ciò permette di verificare se un problema di ottimizzazione, anche se formulato come PLI (risultando quindi fino a prova contraria *difficile*), può essere alla fine formulato come PL (risultando quindi *facile*).

Come preliminare consideriamo la seguente domanda:
Data una matrice **A**, come si può riconoscere [in modo efficiente] se **A** è TUM ?
Un metodo naturale [ma inefficiente] sarebbe enumerare tutte le possibili sottomatrici quadrate **Q** di **A** e poi verificare se det(**Q**) $\in \{-1, 0, 1\}$. Esistono metodi efficienti per determinare se una matrice è TUM, tuttavia questi metodi metodi sono sofisticati, così nel seguito vedremo dei semplici risultati parziali (o condizione necessaria o condizione sufficiente) che comunque sono molto utili.

Una <u>condizione necessaria</u> (come già osservato in precedenza) è che ogni per elemento a_{ij} di **A** si ha che $a_{ij} \in \{-1, 0, 1\}$; infatti ogni elemento di **A** è una sottomatrice quadrata di dimensione 1×1. Tale condizione necessaria non è comunque sufficiente (si possono costruire facilmente delle matrici, che hanno tutti gli elementi in $\{-1, 0, 1\}$, ma che non sono TUM).

Una <u>condizione sufficiente</u> è data dal seguente teorema.

***Teorema* 4.5:** Sia **A** una matrice tale che per elemento a_{ij} di **A** si ha che $a_{ij} \in \{-1, 0, 1\}$.
A è TUM se valgono le seguenti due condizioni:
:: ogni colonna di **A** ha al più due elementi non-nulli;
:: esiste una partizione $\{I_1, I_2\}$ dell'insieme delle righe di **A** tale che per ogni colonna con esattamente due elementi non-nulli si ha che: questi due elementi non-nulli appartengono allo

stesso insieme I_k (per $k \in \{1, 2\}$) se e solo se questi due elementi non-nulli sono concordi in segno. □

Tale condizione sufficiente non è comunque necessaria (si possono costruire facilmente delle matrici, che sono TUM e quindi con tutti gli elementi in $\{-1, 0, 1\}$, ma che non soddisfano almeno una delle due condizioni).

La seguente proposizione fornisce altre utili proprietà delle matrici TUM.

***Proposizione* 4.6:** Sia **A** una matrice intera. Le seguenti affermazioni sono equivalenti:
- **A** è TUM;
- $\mathbf{A}^{\mathbf{T}}$ è TUM;
- la matrice ottenuta da **A** permutando e/o cambiando di segno ad alcune colonne e/o righe è TUM;
- la matrice ottenuta da **A** aggiungendo una colonna o una riga composta da tutti elementi uguali a 0 tranne al più un elemento uguale a ±1 (cioè in $\{-1, 1\}$) è TUM. □

Di seguito proviamo a dimostrare, solo tramite esempi, il Corollario 4.4; in particolare tratterremo solo il problema dei Trasporti e il problema del Cammino di costo minimo; il problema di Assegnamento e il problema del Matching possono essere trattati in modo simile al problema dei Trasporti; il problema del Massimo flusso può essere trattato in modo simile al problema del Cammino di costo minimo.

<u>Esempio</u>: problema dei Trasporti

Ci sono s località sorgente e t località destinazione. Ogni località sorgente $i \in \{1, \ldots, s\}$ ha a disposizione $d_i \geq 0$ unità di un certo tipo di merce, e ogni destinazione $j \in \{1, \ldots, t\}$ richiede almeno $r_j \geq 0$ unità della stessa merce. Per ogni coppia (i, j), per $i \in \{1, \ldots, s\}$ e $j \in \{1, \ldots, t\}$, sono inoltre stabiliti il costo unitario di trasporto c_{ij} e la quantità massima trasportabile q_{ij}. Pianificare i trasporti in modo da minimizzare il costo totale.

Variabili decisionali: x_{ij} per $i = 1, \ldots, s$ e $j = 1, \ldots, t$;
x_{ij} indica la quantità trasportata dalla sorgente i alla destinazione j.

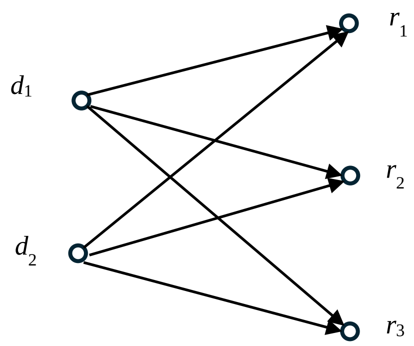
Modello: min $\displaystyle\sum_{i=1}^{s}\sum_{j=1}^{t} c_{ij}x_{ij}$

$\displaystyle\sum_{j=1}^{t} x_{ij} \leq d_i$ per $i = 1, \ldots, s$ (vincoli di disponibilità)

$\displaystyle\sum_{i=1}^{s} x_{ij} \geq r_j$ per $j = 1, \ldots, t$ (vincoli di richiesta)

$$
\begin{array}{lll}
x_{ij} \le q_{ij} & \text{per } i = 1, \dots, s \text{ e } j = 1, \dots, t & \text{(vincoli di capacità)} \\
x_{ij} \ge 0 & \text{per } i = 1, \dots, s \text{ e } j = 1, \dots, t & \\
x_{ij}\ \underline{\text{intero}} & \text{per } i = 1, \dots, s \text{ e } j = 1, \dots, t & \text{(se necessario)}
\end{array}
$$

Notiamo che, il fatto che le variabili sono vincolate a essere intere implica che le unità del bene da trasportare sono indivisibili, così si può assumere che i termini noti d_i e r_j sono interi.
Assumiamo (per ora) che $q_{ij} = \infty$.
Così la matrice **A** riguarderà solamente i vincoli di disponibilità e i vincoli di richiesta.
Consideriamo la seguente istanza del problema.



costi c_{ij}

SORGENTI DESTINAZIONI

Allora vediamo di seguito, in modo esplicito, le disequazioni relative alla matrice **A**.

$$
\begin{array}{ll}
x_{11} + x_{12} + x_{13} & \le d_1 \\
x_{21} + x_{22} + x_{23} & \le d_2 \\
x_{11} + x_{12} & \ge r_1 \\
x_{12} + x_{22} & \ge r_2 \\
x_{13} + x_{23} & \ge r_3
\end{array}
$$

Esse possono essere visualizzate come di seguito così da evidenziare (meglio) la matrice **A**.

$$
\begin{array}{lllll}
x_{11} + x_{12} + x_{13} & & & & \le d_1 \\
 & x_{21} & + x_{22} & + x_{23} & \le d_2 \\
x_{11} & + x_{21} & & & \ge r_1 \\
x_{12} & & + x_{22} & & \ge r_2 \\
x_{13} & & & + x_{23} & \ge r_3
\end{array}
$$

In particolare, con riferimento alle prime due disequazioni, moltiplichiamo entrambi i "lati" di tali disequazioni per −1, così da preparare la formulazione in accordo al Corollario 4.3:

$$
\begin{array}{rrrrrrl}
-x_{11} & -x_{12} & -x_{13} & & & & \geq d_1 \\
 & & & -x_{21} & -x_{22} & -x_{23} & \geq d_2 \\
x_{11} & & & +x_{21} & & & \geq r_1 \\
 & x_{12} & & & +x_{22} & & \geq r_2 \\
 & & x_{13} & & & +x_{23} & \geq r_3
\end{array}
$$

A questo punto, con riferimento al Corollario 4.3, la matrice **A** è la seguente:

$$
\begin{array}{rrrrrr}
-1 & -1 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & -1 & -1 \\
1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 1
\end{array}
$$

Ora applichiamo il Teorema 4.5 per verificare che **A** è TUM. Si noti che: per elemento a_{ij} di **A** si ha che $a_{ij} \in \{-1, 0, 1\}$; ogni colonna di **A** ha al più due elementi non-nulli; esiste una partizione $\{I_1, I_2\}$ dell'insieme delle righe, sia R, di **A** come indicato nel Teorema 4.4 infatti tale partizione è data da $I_1 = R$ e $I_2 = \emptyset$.

Osserviamo infine che, anche assumendo che $q_{ij} \geq 0$, cioè anche considerando i vincoli $x_{ij} \leq q_{ij}$ ($i = 1, \ldots, s$, e $j = 1, \ldots, t$), si ha che la matrice **A*** così estesa è TUM: infatti i vincoli $x_{ij} \leq q_{ij}$ ($i = 1, \ldots, s$, e $j = 1, \ldots, t$) vanno a estendere la matrice di partenza **A** con una *matrice identità* [che ha gli elementi sulla diagonale uguali 1 e ha i rimanenti elementi uguali a 0], così che si può applicare la Proposizione 4.6.

Questo dimostra il Corollario 4.4 nel caso del problemi dei Trasporti, dato che **A** è TUM e dato che i termini noti d_i e r_j sono interi [cioè dato che il "vettore **b**" è intero].

<u>Esempio</u>: problema del Cammino di costo minimo
Sia $G = (V, A)$ un grafo orientato e siano s e t due vertici di G. Ad ogni arco $(i, j) \in A$ è associato un costo $q_{ij} \geq 0$. Il costo di un cammino da s a t è dato dalla somma dei costi degli archi che formano il cammino. Determinare un cammino da s a t di costo minimo.

Variabili decisionali: x_{ij} per ogni arco $(i, j) \in A$;
x_{ij} avrà valore 1 se l'arco (i, j) sta nel cammino;
x_{ij} avrà valore 0 se l'arco (i, j) non sta nel cammino.

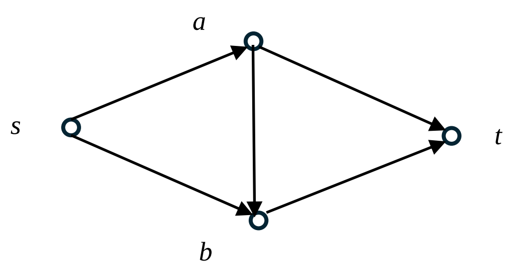
Modello: min $\sum_{(i,j)\in A} q_{ij} x_{ij}$

$$\sum_{(s,j)\in A} x_{sj} = 1$$

$$\sum_{(i,j)\in A} x_{ij} - \sum_{(j,i)\in A} x_{ji} = 0 \quad \text{per ogni vertice } i \neq s, t \text{ di } G$$

$x_{ij} \geq 0$ per ogni $(i, j) \in A$
$x_{ij} \leq 1$ per ogni $(i, j) \in A$
x_{ij} <u>intero</u> per ogni $(i, j) \in A$

Consideriamo la seguente istanza del problema.



$$x_{sa} + x_{sb} = 1$$
$$-x_{sa} + x_{ab} + x_{at} = 0$$
$$-x_{sb} + x_{bt} = 0$$

Esse possono essere visualizzate come di seguito così da evidenziare (meglio) la matrice **A**.

$$\begin{array}{rrrrrl} x_{sa} & +x_{sb} & & & & = 1 \\ -x_{sa} & & +x_{ab} & +x_{at} & & = 0 \\ & -x_{sb} & & & +x_{bt} & = 0 \end{array}$$

A questo punto, con riferimento al Corollario 4.3, la matrice **A** è la seguente:

$$\begin{array}{rrrrr} 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 & 0 \\ 0 & -1 & 0 & 0 & 1 \end{array}$$

Ora applichiamo il Teorema 4.5 per verificare che **A** è TUM. Si noti che: per elemento a_{ij} di **A** si ha che $a_{ij} \in \{-1, 0, 1\}$; ogni colonna di **A** ha al più due elementi non-nulli; esiste una partizione $\{I_1, I_2\}$ dell'insieme delle righe, sia R, di **A** come indicato nel Teorema 4.5 infatti tale partizione è data (anche questa volta) da $I_1 = R$ e $I_2 = \emptyset$.

Questo dimostra il Corollario 4.4 nel caso del problema del Cammino di costo minimo, dato che **A** è TUM e dato che i termini noti sono interi.

Metodo del "Branch and Bound"

Il metodo del "Branch and Bound" è un metodo per risolvere un problema di PLI.
Due considerazioni preliminari: (1) tale metodo non è l'unico per risolvere un problema di PLI, in particolare in [Fischetti] è riportato il metodo dei "Piani di Taglio", che è un metodo geometrico che consiste nel risolvere iterativamente il rilassamento continuo del problema corrente, andando in ogni iterazione [nel caso in cui la soluzione ottima, diciamo **x**, del rilassamento continuo non sia intera] ad aggiungere una nuova disequazione lineare detta "piano di taglio" al problema corrente, in modo da escludere la soluzione **x** nell'iterazione successiva senza escludere le soluzioni ammissibili; (2) il metodo del "Branch and Bound" non è un metodo geometrico, potremmo dire che è un metodo di buon senso basato su osservazioni di buon senso, che guida la ricerca di una soluzione ottima in un contesto in cui l'unica strategia sicura sarebbe quella di effettuare una ricerca a tappeto (cioè esaustiva).

Di seguito indichiamo un problema di PLI del tipo **min** $\{\mathbf{cx} : \mathbf{x} \in S\}$ [dove S è l'insieme delle soluzioni ammissibili di tale problema, cioè, S è un insieme di punti interi all'interno di un poliedro] come $(S, \mathbf{c})$ e assumiamo che S sia finito.

Il metodo del "Branch and Bound" consiste nell'*interazione* di due fasi:
la fase di BRANCH e la fase di BOUND.

• La fase di BRANCH si basa sulla seguente osservazione:
se il problema $(S, \mathbf{c})$ è complicato da risolvere, allora *si definisce una partizione* di S, sia $\{S_1, \ldots, S_t\}$ con $t \geq 2$; per definizione di partizione, gli insiemi $S_1, \ldots, S_t$ sono mutuamente disgiunti, ma allo stesso tempo l'unione degli insiemi $S_1, \ldots, S_t$ è l'insieme S; in particolare siano:
z_i il valore della soluzione ottima del sotto-problema $(S_i, \mathbf{c})$, per $i \in \{1, \ldots, t\}$, e
z il valore della soluzione ottima del problema $(S, \mathbf{c})$;
allora si ha che

$$z = \min \{z_i : i \in \{1, \ldots, t\}\}.$$

Tale procedimento può essere iterato per ciascun (S_i, **c**) … Si può osservare che una iterazione spropositata di tale procedimento potrebbe portare a un numero intrattabile di sotto-problemi. Al fine di evitare ciò, risulta utile la fase di BOUND, descritta di seguito.

• La fase di BOUND si basa sulla seguente osservazione:
sia $\check{z}$ il valore della migliore soluzione trovata fino a un certo punto, cioè, $\check{z}$ è un *ottimo corrente*; per ogni sotto-problema (S_i, **c**) si calcola un *lower bound* del valore della soluzione ottima di (S_i, **c**), sia L_i, così che vale la relazione $L_i \leq z_i$; si osservi che:

se $\check{z} \leq L_i$, allora il sotto-problema (S_i, **c**) *si può chiudere*;

infatti, dato che $\check{z} \leq L_i \leq z$, si avrebbe $\check{z} \leq z_i$; cioè si avrebbe che nel sotto-problema (S_i, **c**) non si sono soluzioni che hanno valore più basso (e quindi migliore) di $\check{z}$ che è l'ottimo corrente.

L'efficacia del metodo del "Branch and Bound" dipende proprio dalle:
<u>strategie di separazione</u>, cioè, il modo in cui si partizionano via via i sotto-problemi;
<u>strategie di soluzione</u>, cioè, il modo in cui si calcolano via via i *lower bound*.

Un esempio di strategia di separazione è il seguente: assumiamo che le variabili del sotto-problema (S_i, **c**), siano $x_1, \ldots, x_n$, siano binarie cioè con valori in {0, 1}; allora una possibile partizione di S_i si può ottenere fissando una variabile, sia x_j , e definendo due sottoinsiemi di S come segue: $\{\mathbf{x} \in S_i : = 1\}$ e $\{\mathbf{x} \in S_i : = 0\}$.

Un esempio di strategia di soluzione – il più usuale sembra – è il seguente: si risolve il rilassamento continuo del sotto-problema (S_i, **c**); allora, come osservato al momento di introdurre il *rilassamento continuo* di un problema di PLI, la soluzione ottima sia **x** di tale rilassamento continuo è proprio un *lower bound* del valore della soluzione ottima di (S_i, **c**); inoltre, se **x** è intera, allora **x** è soluzione ottima anche di (S_i, **c**).

5. Alcuni problemi specifici con metodi di soluzione specifici

In questo capitolo focalizzeremo i seguenti cinque problemi:
--- Problema del Cammino di costo minimo,
--- Problema della Pianificazione di progetti,
--- Problema del Massimo flusso,
--- Problema della Programmazione della produzione,
--- Problema di Localizzazione di impianti.

Con riferimento all'Appendice 1:

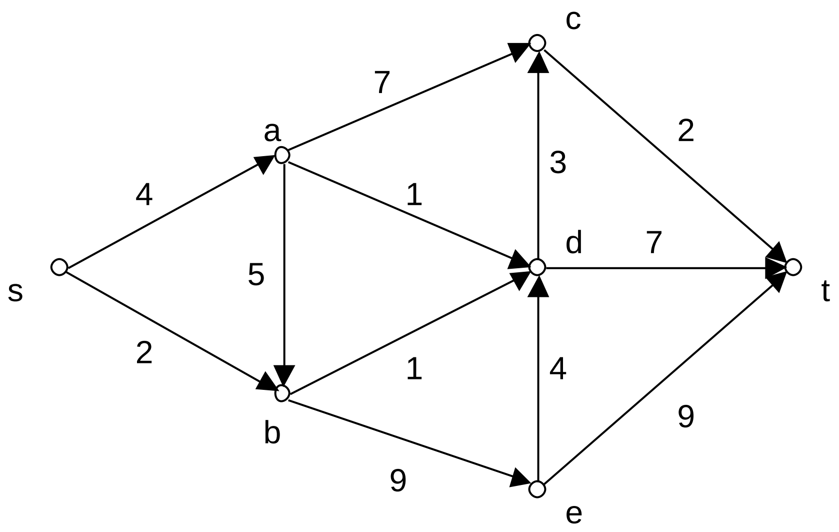
:: i primi quattro problemi sono problemi *facili*; in particolare essi sono formulabili in termini di PL (riguardo il quarto ciò è vero sotto alcune ipotesi), così in generale già sappiamo come risolverli con il metodo del Simplesso, tuttavia introduciamo per ognuno di questi quattro problemi alcune rispettive proprietà che "caratterizzano" le soluzioni ottime e che permettono di definire metodi di soluzione "ad hoc" (più veloci rispetto al metodo del Simplesso);

:: il quinto problema è un problema *difficile*; in particolare esso sono formulabili in termini di PLI, così in generale già sappiamo come risolverli con il metodo del "Branch and Bound", tuttavia introduciamo un "metodo euristico" per trovare una soluzione presumibilmente buona ma non necessariamente ottima del problema.

5.1 Problema del Cammino di costo minimo

Definizione del problema:

Sia $G = (V, A)$ un grafo orientato e siano s e t due vertici di G. Ad ogni arco $(i, j) \in A$ è associato un costo $q_{ij} \geq 0$. Il costo di un cammino da s a t è dato dalla somma dei costi degli archi che formano il cammino. Il problema è determinare un cammino da s a t di costo minimo.



Osservazioni

:: Ricordiamo che tale può essere formulato come un problema di PLI (cfr. Capitolo 2, esempio C1).

:: Ricordiamo che tale problema, anche se formulato inizialmente come un problema di PLI, può essere formulato (e quindi considerato) come un problema di PL; infatti come abbiamo visto riguardo le matrici TUM, è possibile eliminare [nella formulazione in termini di PLI] i vincoli di interezza, per poi risolvere il rilassamento continuo [di tale formulazione]; in particolare il problema del Cammino minimo può essere risolto con il metodo del Simplesso.

:: La formulazione come problema di PLI (in generale) è adattabile a varianti come di seguito. Ad esempio: ad ogni arco $(i, j) \in A$ può essere associato un tempo $t_{ij} \geq 0$, e si può stabilire un tempo massimo sia T per arrivare da s a t; il tempo di un cammino da s a t è dato dalla somma dei tempi degli archi che formano il cammino; allora il problema può essere: determinare un cammino da s a t di costo minimo che abbia un tempo non superiore a T; per modellare tale variante basta aggiungere [nella formulazione in termini di PLI] il vincolo $\sum_{(i,j)\in A} t_{ij}\, x_{ij} \leq T$ □

Al fine di introdurre un metodi di soluzione "ad hoc" per il problema del Cammino di costo minimo – oltre al metodo del Simplesso come osservato sopra – vediamo due proprietà.

Per brevità diciamo che una soluzione ottima del problema del cammino di costo minimo è un *cammino ottimo*. Vale il seguente risultato:

***Teorema* 5.1.1:** Un cammino è ottimo se e solo se è formato da sotto-cammini ottimi. □

Questo risultato può essere compreso visualizzando ad esempio il cammino ottimo che è indicato da un "navigatore" per un cammino ottimo da Pescara a Roma; allora tale cammino indica implicitamente i rispettivi cammini ottimi da Pescara a tutte le località (tipo Avezzano, Carsoli, …) che fanno parte di tale cammino. La dimostrazione del teorema di sopra è piuttosto intuitiva, per assurdo, basata sul fatto che i costi sono ≥ 0. A questo punto si può intuire che dovrebbe esistere un metodo combinatorio che permette di calcolare un cammino ottimo. Il seguente teorema descrive appunto il nucleo di un metodo combinatorio che permette di calcolare un cammino ottimo.

***Teorema* 5.1.2:** Sia L_v il costo del cammino di costo minimo da s a v , per ogni $v \in V$, e sia $\{W, Z\}$ una partizione di V tale che $L_w \leq L_z$ per ogni $(w, z) \in W \times Z$. Sia $(w^*,z^*) = \text{argmin}\ \{L_w + q_{wz} : (w, z) \in W \times Z\}$. Allora $L_{z^*} \leq L_z$ per ogni $z \in Z$, cioè si potrà aggiungere z^* in W, e si ha $L_{z^*} = L_{w^*} + q_{w^*z^*}$. □

Algoritmo di Dijkstra

Input: $G = (V, E)$ con $|V| = n$; $s, t \in V$; $q_{ij} \geq 0$ per ogni $(i,j) \in E$.

Output: un cammino di costo minimo da s a t

<u>Inizializzazione</u>

:: $W := \{s\}$

:: costruisci un vettore α con n componenti, una per ogni vertice v di G, come segue:

$\alpha(v) = q_{sv}$, se esiste l'arco (s, v)

$\alpha(v) = \infty$, altrimenti

<u>Inizializzazione</u>

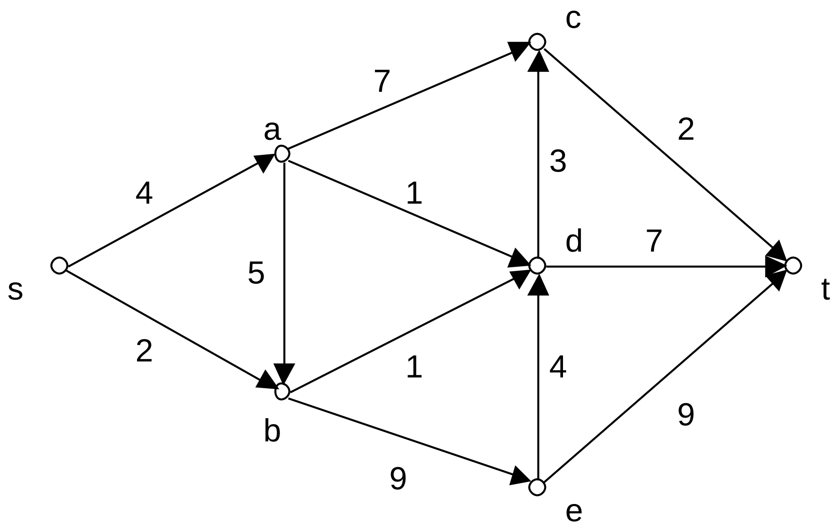
:: scegli un vertice $z^* \in V \setminus W$, tale che $\alpha(z^*)$ sia minimo;

:: inserisci z^* in W;

:: aggiorna il vettore α ponendo $\alpha(z) := \min\ \{\alpha(z) , \alpha(z^*) + q_{z^*z}\}$ per ogni $z \in V \setminus W$. □

Esercizio svolto

Calcolare mediante il metodo di Djikstra un cammino da s a t di costo minimo nel seguente grafo orientato $G = (V, A)$, in cui per ogni arco (i, j) è indicato il costo associato $q_{ij} \geq 0$.



Soluzione

Il problema è determinare un cammino di costo minimo da s a t, dove il costo di un cammino è dato dalla somma dei costi associati agli archi del cammino (ad esempio, il cammino $s \to a \to d \to t$ ha costo 4 + 1 + 7 = 12). Il metodo di soluzione è basato sul Teorema 5.1.2.

Il metodo costruisce iterativamente tale insieme W, fino ad inserire t in W.

Inizializzazione

$W := \{s\}$. Costruisci il vettore α, che ha una componente per ogni vertice v del grafo, con $z(v)$ uguale: a q_{sv} se esiste l'arco sv, a ∞ altrimenti.

s	a	b	c	d	e	t
0	4	2	∞	∞	∞	∞

Passo generico

- scegli un vertice $z^* \in V \setminus W$, tale che $\alpha(z^*)$ sia minimo;
- inserisci z^* in W;
- aggiorna il vettore α ponendo $\alpha(z) := \min \{\alpha(z), \alpha(z^*) + q_{z^*z}\}$ per ogni $z \in V \setminus W$. □

Passo 1

- scegli b
- $W := W \cup \{b\}$
- aggiorna il vettore α come di seguito

s	*A*	*b*	*c*	*d*	*e*	*t*
0	4	**2**	∞	3	11	∞

Passo 2

- scegli d
- $W := W \cup \{d\}$
- aggiorna il vettore α come di seguito

s	*A*	*b*	*C*	*d*	*e*	*t*
0	4	**2**	6	**3**	11	10

Passo 3

- scegli a
- $W := W \cup \{a\}$
- aggiorna il vettore α come di seguito

s	*A*	*b*	*C*	*d*	*e*	*t*
0	**4**	**2**	6	**3**	11	10

Passo 4

- scegli c
- $W := W \cup \{c\}$
- aggiorna il vettore α come di seguito

s	*A*	*b*	*C*	*d*	*e*	*t*
0	**4**	**2**	**6**	**3**	11	8

Passo 5

- scegli t
- $W := W \cup \{t\}$
- Il metodo termina dato che t è stato inserito in W

Il numero $z(t) = 8$ indica il costo di un cammino di costo minimo da s a t. Per individuare tale cammino, si può procedere andando a ritroso da t verso s in questo modo:

$\alpha(t)$ è stato aggiornato l'ultima volta quando c è entrato in W;

$\alpha(c)$ è stato aggiornato l'ultima volta quando d è entrato in W;

$\alpha(d)$ è stato aggiornato l'ultima volta quando b è entrato in W;

$\alpha(b)$ è stato aggiornato l'ultima volta quando s è entrato in W.

Si ottiene il cammino: $s \to b \to d \to c \to t$ (con costo = 8) □

Commenti finali sulle applicazioni

1) Il problema del Cammino di costo minimo ha molte applicazioni di natura pratica e di natura teorica; ne riportiamo, di seguito, solo alcune:

--- Programmazione Dinamica: il Teorema 5.1.1 [cioè: un cammino è ottimo se e solo se è formato da sotto-cammini ottimi, detto più in generale, una soluzione è ottima se e solo se è formata da sotto-soluzioni ottime] è un risultato che caratterizza i problemi così detti di Programmazione Dinamica; così il problema del Cammino di costo minimo è un problema di Programmazione Dinamica, in particolare ha inoltre un ruolo speciale in questo contesto, nel senso che ogni problema di Programmazione Dinamica può essere visualizzato come una istanza del problema del Cammino di costo minimo;

--- reti stradali: il problema di trovare un cammino di costo minimo (o di tempo minimo) fra due punti di una rete stradale è evidentemente una istanza del problema del Cammino di costo minimo; così il "navigatore", nella sua funzione basilare, risolve proprio questo problema;

--- localizzazione: trovare il *centro* di una rete può significare trovare un punto della rete che minimizza la somma dei cammini minimi da quel punto a ciascun punto della rete;

--- sotto-procedura: il problema del Cammino di costo minimo può essere utilizzato come sott-procedura nella risoluzione di altri problemi, ad esempio, il problema delle Commissioni che abbiamo introdotto all'inizio di questi appunti.

2) Cosa succede se il problema del Cammino di costo minimo non dovesse valere il fatto che $q_{ij} \geq 0$, cioè, se i costi degli archi del grafo G potessero essere anche negativi? Facendo sempre riferimento al libro [Fischetti], in breve, diciamo che in tal caso: se G ha un "ciclo negativo" (cioè se G ha un ciclo tale che la somma dei costi degli archi del ciclo è negativa), allora il problema è illimitato; se G non ha un “ciclo negativo”, allora si può comunque trovare un cammino ottimo in modo efficiente, mediante l’Algoritmo di Floyd-Warshall [il quale, inoltre, trova in simultanea un cammino ottimo fra ogni coppia di vertici del grafo G]. $\square$

5.2 Problema delle Pianificazione di progetti

<u>Definizione del problema</u>: Un *progetto* è un insieme di n attività A_i, per $i \in \{1, \ldots, n\}$, ciascuna con una durata $d_i \geq 0$ nota. Fra alcune attività sono specificate relazioni di precedenza $A_i < A_j$ ad indicare che l'istante di completamento dell'attività A_i deve precedere l'istante di inizio dell'attività A_j. Il problema è pianificare le attività in modo da minimizzare il tempo di completamento del progetto.

Osservazioni

:: Ricordiamo che tale può essere formulato come un problema di PL (cfr. Capitolo 2, esempio C2).

:: La formulazione come problema di PL (in generale) è adattabile a varianti come di seguito. Supponiamo di inserire altre relazioni fra le attività, diciamo, una *relazione di incompatibilità*, che stabilisce che fra due attività che fra due attività c'è una relazione di incompatibilità se esse devono essere svolte in intervalli di tempo disgiunti (un esempio: consideriamo il "progetto" basato sulle attività che una persona svolge da quando si sveglia fino a quando esce di casa; allora ci sono attività che sono fra loro in relazione di precedenza, tipo il fare colazione e il lavarsi i denti, ma ci sono anche attività che sono fra loro in relazione di incompatibilità, tipo il fare colazione e il vestirsi); allora per modellare la presenza di attività che sono fra loro incompatibili, si può utilizzare l'artificio che abbiamo visto nel Capitolo 2, esempio D3 (vincoli disgiuntivi), che permette appunto di stabilire una precedenza fra due attività incompatibili. □

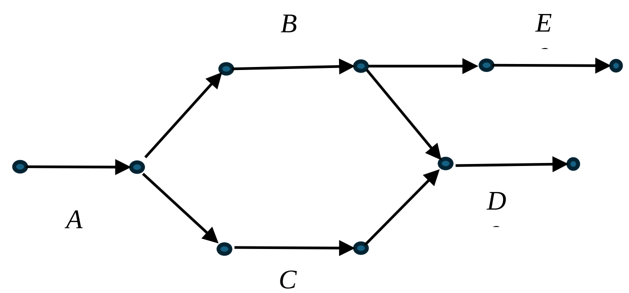
Al fine di introdurre un metodo di soluzione "ad hoc" per il problema della Pianificazione di progetti – oltre al metodo del Simplesso come osservato sopra – introduciamo una costruzione.

Costruiamo un grafo orientato, sia G, in modo che:
per ogni attività A_h, con durata d_h, esiste un arco (x, y) con durata $d_{xy} = d_h$;
per ogni relazione di precedenza $A_i < A_j$, esiste un arco fittizio (x, y) con durata $d_{xy} = 0$.

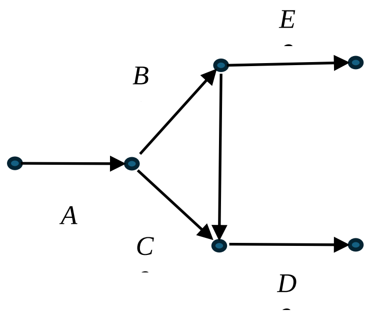
<u>Esempio</u>

Attività: A, B, C, D, E

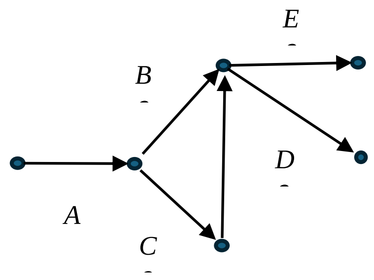
Relazioni di precedenza: $A < B, A < C, B < D, B < E, C < D$



Si può semplificare "contraendo" alcuni archi fittizi, ma senza aggiungere/togliere precedenze.



SI'



NO (si aggiunge $C < E$)

Proprietà 1: Il grafo G è aciclico: infatti, non esistono attività tali, diciamo $A_1, \ldots, A_k$, tali che $A_1 < \ldots < A_k < A_1$, sarebbe una inconsistenza logica. □

Sia V l'insieme dei vertici di G e siano:

$V^- = \{v \in V : \text{in } v \text{ non "entra" nessun arco}\}$

$V^+ = \{ v \in V : \text{da } v \text{ non "esce" nessun arco}\}$

Proprietà 2: $V^- \neq \emptyset$ e $V^+ \neq \emptyset$: tale proprietà è una conseguenza della Proprietà 1. □

A questo punto estendiamo il grafo G come segue:

:: aggiungi un vertice s e aggiungi gli archi (s, v), per ogni $v \in V^-$, con durata $d_{sv} = 0$;

:: aggiungi un vertice t e aggiungi gli archi (v, t), per ogni $v \in V^+$, con durata $d_{vt} = 0$.

Ricapitolando, G è un grafo aciclico, con un vertice s in cui non "entra" nessun arco e con un vertice t da cui non "esce" nessun arco; inoltre ad ogni arco di G è associata una "durata"; in particolare la durata di una cammino è data dalla somma delle durate degli archi che lo compongono. Dalla costruzione di sopra, si può intuire che la durata minima del progetto non può essere maggiore della durata del cammino di durata massima da s a t in G, poiché un tale cammino [massimo] è comunque un limite inferiore nel contesto del progetto; per la precisione, vale il seguente risultato, che è alla base del metodo "ad hoc" che stiamo introducendo.

***Teorema* 5.2.1:** La durata del cammino di durata massima da s a t in G è uguale al tempo minimo di completamento del progetto. □

A questo punto, in senso tecnico, il problema consiste nel determinare un cammino di durata (che, in generale, può essere di costo, di distanza, ecc.) massima fra due vertici in un grafo orientato. Tuttavia il metodo PERT che stiamo per vedere, oltre a risolvere tale problema, fornisce altre indicazioni che possono essere utili al decisore; introduciamo alcuni preliminari.

<u>Ordinamento topologico in un grafo orientato aciclico G.</u>

1. Assegna il numero 1 a un qualunque vertice $v \in V^-$.
2. Elimina v, con gli archi che incidono su v, dal grafo.
3. Ripeti il procedimento per assegnare, in sequenza, i numeri 2, 3, …, n, ai vertici rimanenti. □

<u>Eventi, $T_{min}[h]$, $T_{max}[h]$</u>

Mentre ogni arco di G è inteso come attività (reale o fittizia), ogni vertice di G può essere inteso come un *evento*; in particolare, per ogni vertice h di G, cioè per ogni evento h siano:

$T_{min}[h]$ = istante minimo di inizio a partire dal quale l'evento h può accadere;

$T_{max}[h]$ = istante massimo di inizio entro il quale l'evento h deve accadere (al fine di non generare ritardi nel tempo minimo di completamento del progetto).

Algoritmo PERT

<u>Input</u>: grafo $G = (V, E)$ con $|V| = n$; $s, t \in V$; durata $d_{ij} \geq 0$ per ogni $(i, j) \in E$.

<u>Ouput</u>: un cammino di durata massima da s a t, cioè, il tempo minimo di completamento del progetto uguale a $T_{min}[n]$; altre informazioni utili per il decisore.

Step 1) Effettua un ordinamento topologico in G.

Step 2) $T_{min}[1] := 0$.

Step 3) per $h = 2, \dots, n$, poni $T_{min}[h] := \max \{ T_{min}[i] + d_{ih} : (i, h) \in E\}$

Step 4) $T_{max}[n] := T_{min}[n]$.

Step 5) per $h = n-1, \dots, 1$, poni $T_{max}[h] := \min \{ T_{max}[j] + d_{hj} : (h, j) \in E\}$. □

Alcuni commenti sull'algoritmo PERT:

l'algoritmo può essere visto in due fasi; prima una fase di "andata" da *s* a *t*, in cui i valori $T_{min}[h]$ (per ogni evento *h*) vengono determinati, e in cui il valore $T_{min}[n]$ indicherà la durata minima del progetto; poi una fase di "ritorno" da *s* a *t*, in cui i valori $T_{max}[h]$ (per ogni evento *h*) vengono determinati, fornendo così altre informazioni al decisore (ricordando il significato del valore $T_{max}[h]$ introdotto sopra);

in dettaglio: una attività $(i, j) \in E$ è *critica* quando $T_{min}[i] = T_{min}[j]$, $T_{max}[i] = T_{max}[j]$, $T_{min}[i] + d_{ij} = T_{max}[j]$; un cammino da *s* a *t* è *critico* se è formato da attività critiche; vale la ***proprietà*** che *ogni progetto ammette almeno un cammino critico*; si ha che la durata di un cammino critico è uguale al tempo minimo di completamento del progetto; in particolare, al fine di garantire il tempo minimo di completamento del progetto, le attività critiche sono quelle che dovrebbero scorrere in modo perfetto (cioè, iniziare appena possibile, terminare senza ritardo);

concludiamo menzionando che una volta eseguito l'algoritmo PERT, come illustrato in [Fischetti], il decisore potrà visualizzare tutte le attività nel così detto *diagramma di Gantt* lungo una linea temporale: tale diagramma evidenzia, oltre a un cammino critico, la possibilità di "parallelizzare" le varie attività.

Esercizio svolto

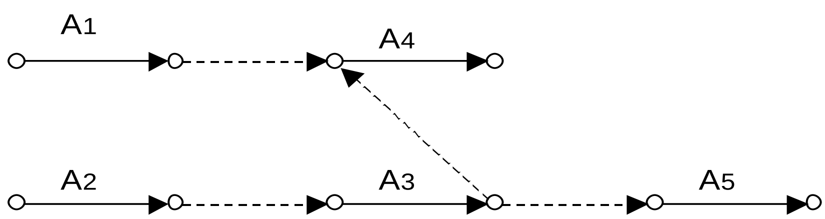
Calcolare mediante il metodo PERT il tempo minimo di completamento, le attività critiche e i massimi slittamenti ammessi (per ogni attività) nel seguente progetto:

le attività sono 5: $A_1, \ldots, A_5$; ogni attività A_i ha durata d_i, dove d = [5, 7, 2, 4, 3];

le relazioni di precedenza sono: $A_1 < A_4$; $A_2 < A_3$; $A_3 < A_4$; $A_3 < A_5$.

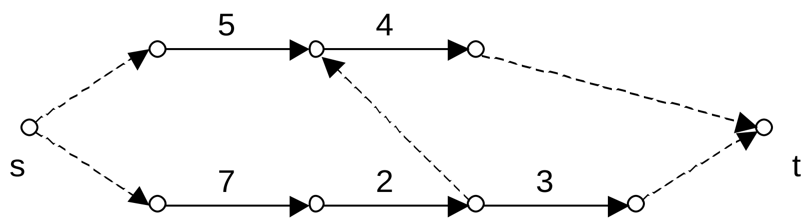
<u>Soluzione</u>

- Costruisci un grafo orientato *G* associando ad ogni attività un arco e introducendo le precedenze con archi fittizi.

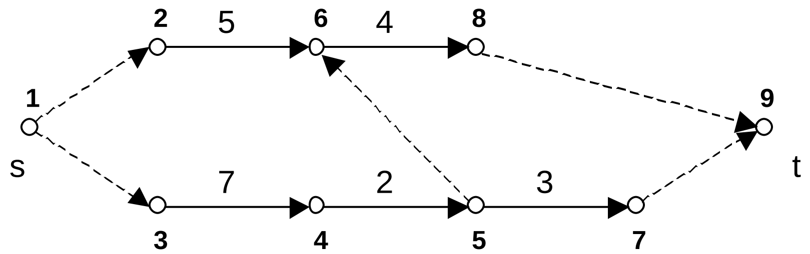


- Costruisci un grafo *G'* modificando il grafo *G* come segue: (i) elimina gli archi fittizi che sono obsoleti – tale operazione non deve né togliere né aggiungere precedenze; (ii) aggiungi un vertice *s* e gli archi fittizi *sv* per ogni vertice *v* di *G* in cui non "entra" alcun arco; (iii) aggiungi un vertice t e gli archi fittizi vt per ogni vertice v di G da cui non “esce” alcun arco; (iv) associa ad ogni attività A_i il valore d_i (per i = 1, . . ., 5); (v) associa ad ogni arco fittizio il valore 0 (non riportato in figura).

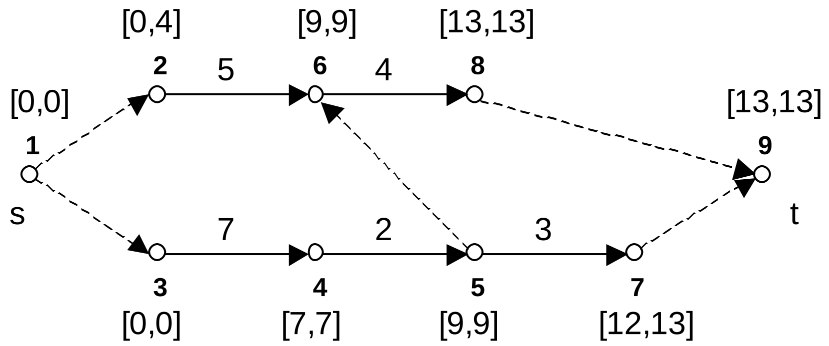
Osservazione: i valori d_i delle attività e i valori 0 degli archi fittizi possono essere visti come “durate” d_{ij} associate agli archi di G' ; così, in accordo con il Teorema 5.2.1, la durata di un cammino di durata massima da s a t in G' è pari al tempo minimo di completamento del progetto.



- Ordina topologicamente i nodi di G' [cioè, associa una numerazione progressiva ai nodi, in modo che se il numero associato a un nodo x è minore di quello associato a un nodo y allora non esiste un cammino in G' da y a x].

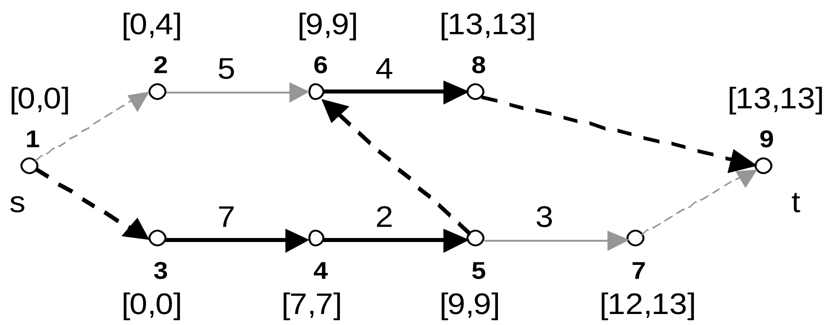


- Associa ad ogni vertice una doppia etichetta [T_{min} , T_{max}], cioè per ogni vertice h di G' calcola due valori $T_{min}(h)$ e $T_{max}(h)$ come segue: (i) $T_{min}(1) = 0$; (ii) per h = 2, …, 9, poni $T_{min}(h) = \max\{T_{min}(i) + q_{ih}$: esiste arco $(i, h)\}$; (iii) $T_{max}(9) = T_{min}(9)$; (iv) per h = 8, …, 1, poni $T_{max}(h) = \min\{T_{max}(i) + q_{hi}$: esiste arco $(h, i)\}$.



Il tempo minimo di completamento del progetto è dato da $T_{min}(9) = 13$.

Le *attività critiche* (cioè gli archi (i, j) tali che: $T_{min}(i) = T_{max}(i)$; $T_{min}(j) = T_{max}(j)$; $T_{max}(j) - T_{max}(i) = q_{ij}$) sono A_2, A_3, A_4. Tali attività sono quelle il cui eventuale ritardo di attuazione genera direttamente un ritardo sul tempo minimo di completamento del progetto. Esse individuano almeno un *cammino critico*, insieme con archi fittizi corrispondenti ad attività critiche fittizie, come evidenziato in figura sottostante.



□

5.3 Problema del Massimo flusso

Definizione del problema: Sia $G = (V, A)$ un grafo orientato e siano s e t due vertici di G. Ad ogni arco $(i, j) \in A$ è associata una capacità $b_{ij} \geq 0$. Un *flusso* in G (da s a t) è un'assegnazione di valori $x_{ij} \geq 0$ per ogni arco $(i, j) \in A$ tale che:

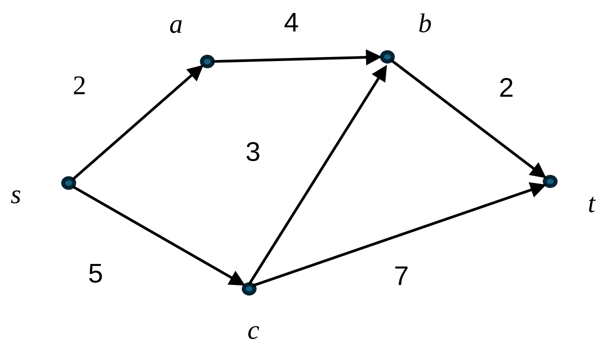
:: $\sum_{(i,j)\in A} x_{ij} - \sum_{(j,i)\in A} x_{ji} = 0$ per ogni vertice $i \neq s, t$ di G (cioè per ogni vertice $i \neq s, t$ di G, la somma di ciò che entra è uguale alla somma di ciò che esce);

:: $x_{ij} \leq b_{ij}$ per ogni arco $(i, j) \in A$.

Il *valore di un flusso* da s a t è $\sum_{(s,j)\in A} x_{sj}$.

Il problema è determinare un flusso da s a t di valore massimo. □

Notazione: Un grafo orientato G definito come sopra, evidenziando due vertici di G (i vertici s, t) ed associando una capacità ad ogni arco di G (i valori b_{ij}) è detto *rete* : così il problema definito sopra sarà di seguito richiamato anche come il problema di trovare un massimo flusso in una rete G. □



Due esempi per focalizzare il problema:

:: la rete G può essere immaginata come una rete idrica, ogni arco è una tubatura con una sua capacità, ogni nodo è un raccordo fra più tubature; un flusso in G indica la quantità di acqua che scorre in ogni tubatura, tale che, per ogni raccordo, la quantità totale di acqua che entra è uguale alla quantità di acqua che esce; così, un flusso di valore massimo indica la massima quantità di acqua che può essere immessa in s e fatta scorrere – a regime – fino a t (come se la rete fosse un'unica tubatura);

:: la rete G può essere immaginata come una rete di aeroporti; in particolare assumiamo che un volo dall'aeroporto s all'aeroporto t sia stato soppresso e che la Compagnia di volo studia comunque il modo di far arrivare più persone possibili da s a t passando per questa rete G (in giornata); ogni vertice della rete diverso da s e t è un possibile aeroporto di transito, ogni arco (i, j) rappresenta un volo fra l'aeroporto i e l'aeroporto j, in particolare la capacità b_{ij} rappresenta il numero di posti disponibili su tale volo; così, un flusso di valore massimo indica la massima quantità di persone che la Compagnia può comunque far arrivare a t.

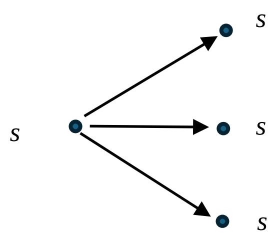
Osservazioni

:: Ricordiamo che tale può essere formulato come un problema di PLI (cfr Capitolo 2, esempio C3).

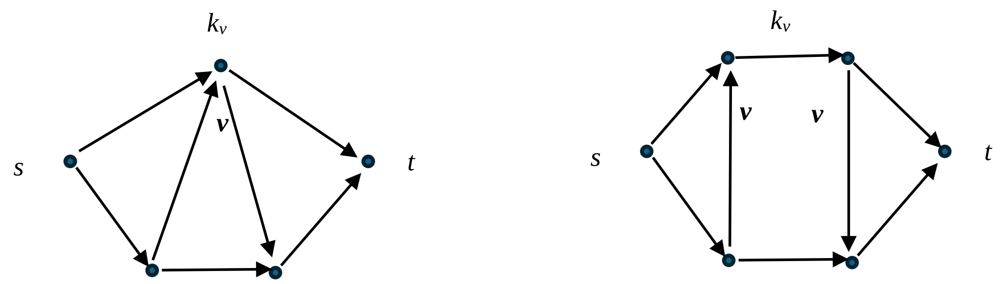
:: Ricordiamo che tale problema, anche se formulato inizialmente come un problema di PLI – notiamo che, il fatto che le variabili sono vincolate a essere intere implica che le unità del bene da trasportare sono indivisibili, così si può assumere che anche i termini noti b_{ij} siano interi – può essere formulato (e quindi considerato) come un problema di PL; infatti come abbiamo visto riguardo le matrici TUM, è possibile eliminare [nella formulazione in termini di PLI] i vincoli di interezza, per poi risolvere il rilassamento continuo [di tale formulazione]; in particolare il problema del Massimo flusso può essere risolto con il metodo del Simplesso.

:: La formulazione come problema di PLI (in generale) è adattabile a varianti come di seguito:

:: :: se ci sono più sorgenti, allora ci si può ricondurre al caso con una sola sorgente, come segue:



:: :: se c'è un nodo v a cui è assegnata una "capacità" k_v, allora ci si può ricondurre al caso in cui solo agli archi sono assegnate le "capacità", come segue:

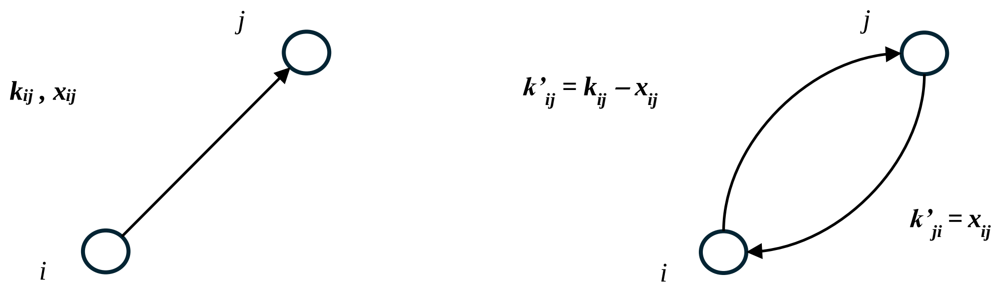


***Definizione*:** Sia **x** un flusso in una rete $G = (V, A)$. La *rete incrementale* $G' = (V, A')$ *rispetto a* **x** è ottenuta dalla rete originale $G = (V, A)$ sostituendo ogni arco $(i, j) \in A$ con due archi:

• un *arco diretto* (i, j) di capacità residua $b'_{ij} = b_{ij} - x_{ij}$

• un *arco inverso* (j, i) di capacità residua $b'_{ji} = x_{ij}$

ed eliminando poi gli archi di capacità nulla. □



***Teorema* 5.3.1:** Un flusso **x** in una rete $G = (V, A)$ è ottimo se e solo se nella rete incrementale $G' = (V, A')$ rispetto a **x** non esiste un cammino da s a t. □

<u>PARENTESI</u> : <u>max flow / min cut</u>

***Definizione*:** Data una rete $G = (V, A)$ [con $s, t \in V$] si dice:

:: *sezione* (o *cut*) *di G* una partizione, sia $\{S, T\}$, di G tale che $s \in S$ e $t \in T$;

:: *capacità della sezione* $\{S, T\}$ la quantità $K(S, T) = \sum_{\substack{i \in S \\ j \in T}} k_{ij}$. □

***Teorema* 5.3.2:** Data una rete $G = (V, A)$ [con $s, t \in V$] si ha che: il valore di un flusso in G di valore massimo è uguale alla capacità di una sezione di G di valore minimo. □

Da un lato, si può verificare (anche solo intuitivamente) che il valore di un flusso in G di valore massimo è minore o uguale alla capacità di una qualsiasi sezione di G la quale costituisce comunque un limite, e in particolare è minore o uguale alla capacità di una sezione di G di valore minimo. Il Teorema 5.3.2 afferma che vi è proprio una uguaglianza. In altri termini, il teorema permette di immaginare la rete come se fosse un unico ipotetico arco (una sola tubatura come visualizzato sopra), calcolando così la capacità ipotetica di questo unico ipotetico arco la quale è naturalmente uguale al valore di un flusso di valore massimo in questo unico ipotetico arco. [Fine PARENTESI]

Algoritmo di Ford-Furlkerson

Input: una rete G cioè: un grafo orientato $G = (V, E)$; $s, t \in V$; $q_{ij} \geq 0$ per ogni $(i,j) \in E$.

Output: un flusso di valore massimo nella rete G.

Step 1) $\mathbf{x} := \mathbf{0}$, cioè, $x_{ij} := 0$ per ogni $(i, j) \in E$;

Step 2) costruisci la rete incrementale rispetto a $\mathbf{x}$, sia $G' = (V, A')$;

Step 3) se in $G' = (V, A')$ non esiste un cammino da s a t, allora STOP [$\mathbf{x}$ è ottimo];

altrimenti:

--- "satura" un cammino da s a t in $G' = (V, A')$;

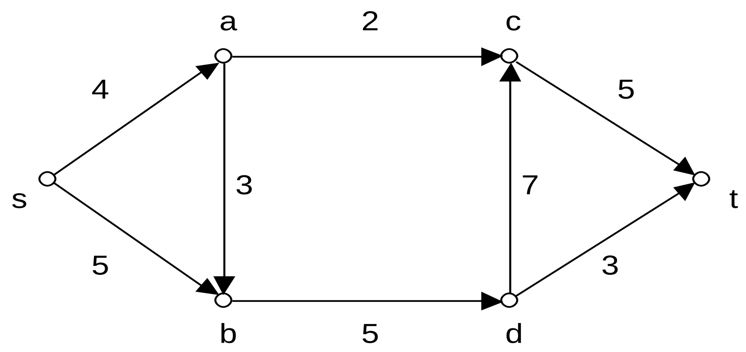
--- aggiorna $\mathbf{x}$ in G ;

--- va allo Step 2 . □

Per illustrare il significato di "saturare" un cammino da s a t in $G' = (V, A')$, immaginiamo prima che $G' = (V, A')$ sia esattamente un cammino da s a t; allora il significato è incrementare il flusso corrente lungo tale cammino il più possibile; in particolare tale incremento massimo sarà uguale alla capacità residua dell'arco di tale cammino che ha minima capacità residua, cioè al *collo di bottiglia* di tale cammino. Nel caso in cui $G' = (V, A')$ non è esattamente un cammino, si può comunque fissare un cammino da s a t, per poi procedere come sopra.

Esercizio svolto

Calcolare mediante l'algoritmo di Ford-Fulkerson un flusso da s a t di valore massimo nella seguente rete, in cui accanto ad ogni arco è indicata la corrispondente capacità.

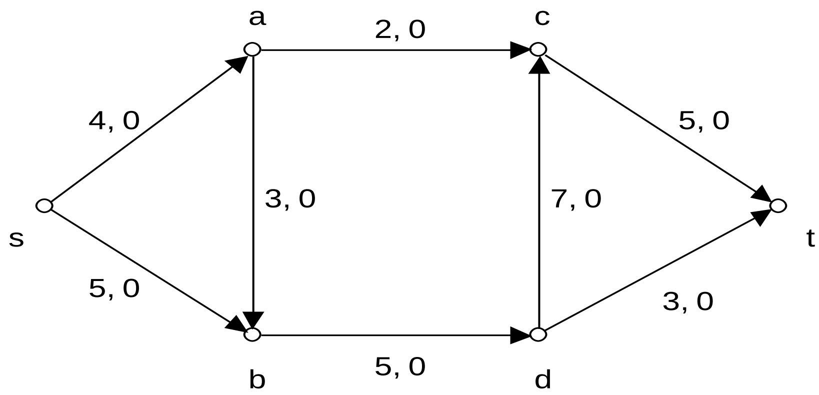


Soluzione

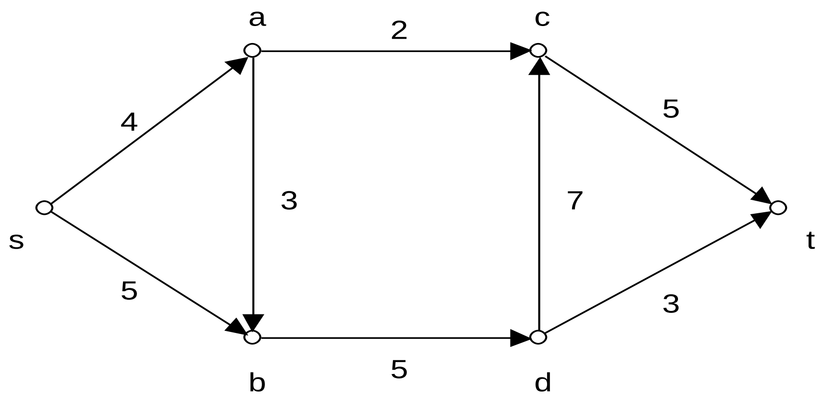
Applichiamo l'algoritmo di Ford-Fulkerson, nelle pagine seguenti.

<u>Inizializzazione</u>

- poni pari a 0 ogni componente del flusso su ogni arco

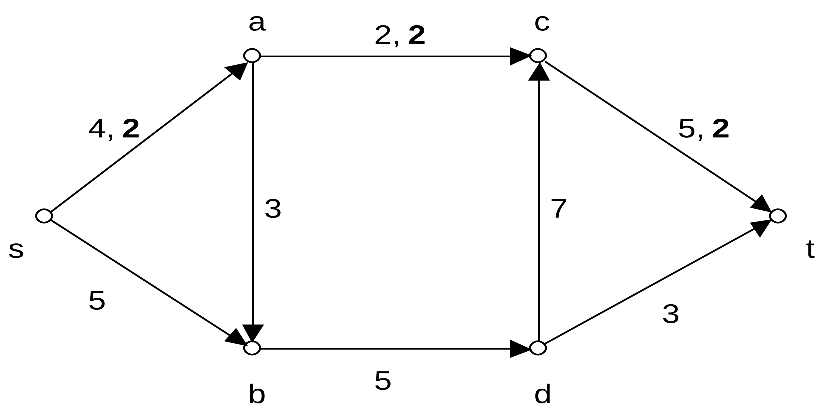


- costruisci rete incrementale rispetto al flusso



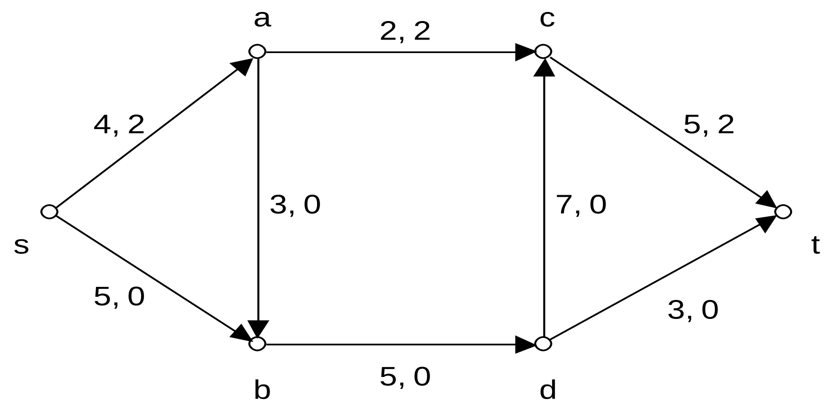
- individua cammino da *s* a *t* nella rete incrementale, e saturalo; un cammino è *s*, *a*, *c*, *t*;

lo si satura "inviando" più flusso possibile, cioè 2 (l'arco *ac* è collo di bottiglia)

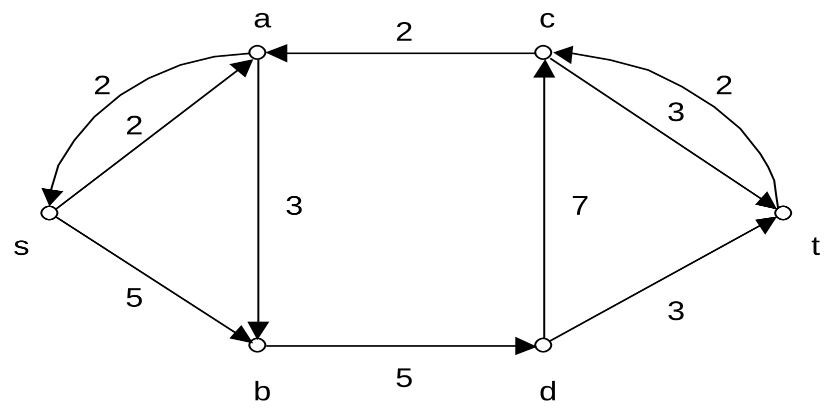


Iterazione 1

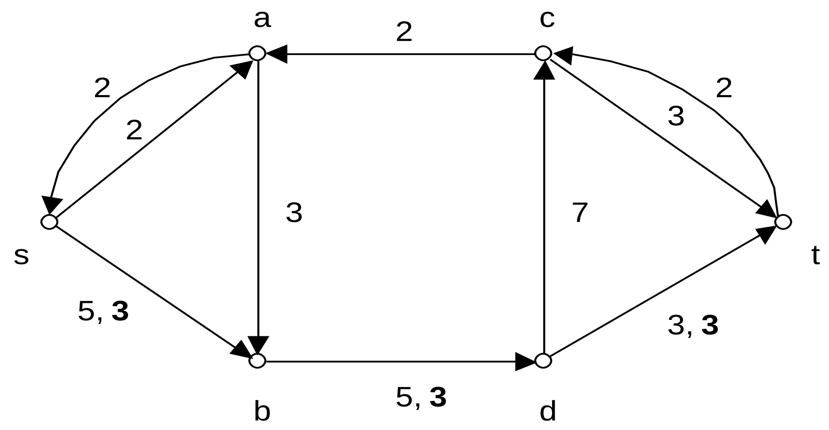
- aggiorna il flusso (in base al cammino aumentante trovato)



- costruisci la rete incrementale relativa al flusso

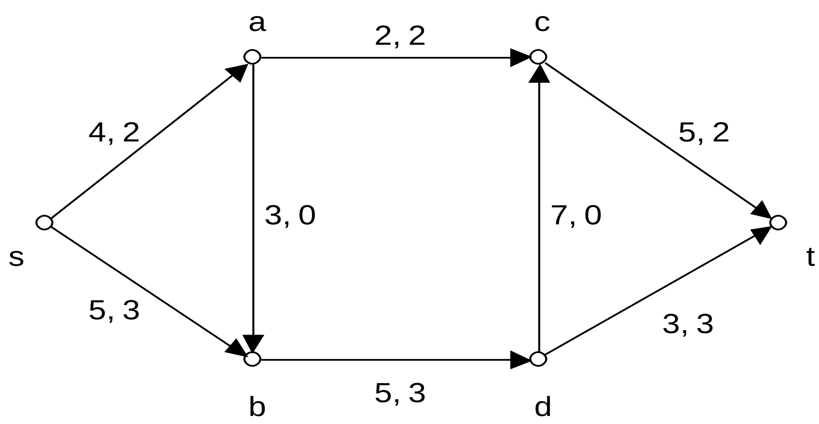


- individua cammino da *s* a *t* nella rete incrementale, e saturalo; un cammino è *s*, *b*, *d*, *t*; lo si satura "inviando" più flusso possibile, cioè 3 (l'arco *dt* è collo di bottiglia)

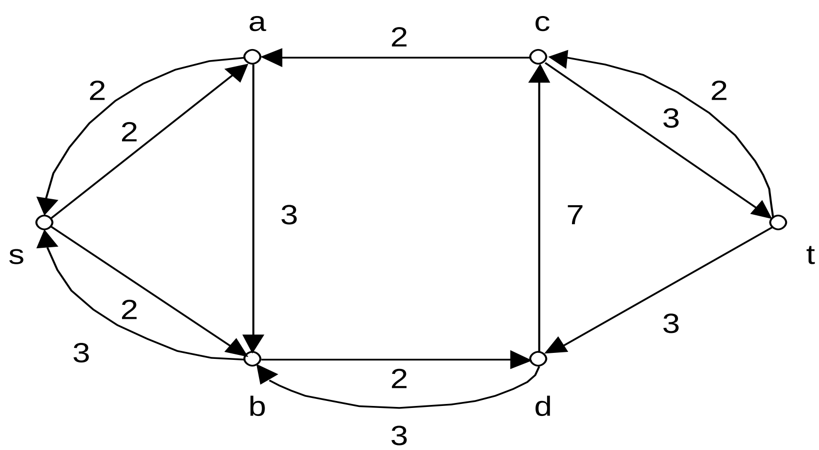


Iterazione 2

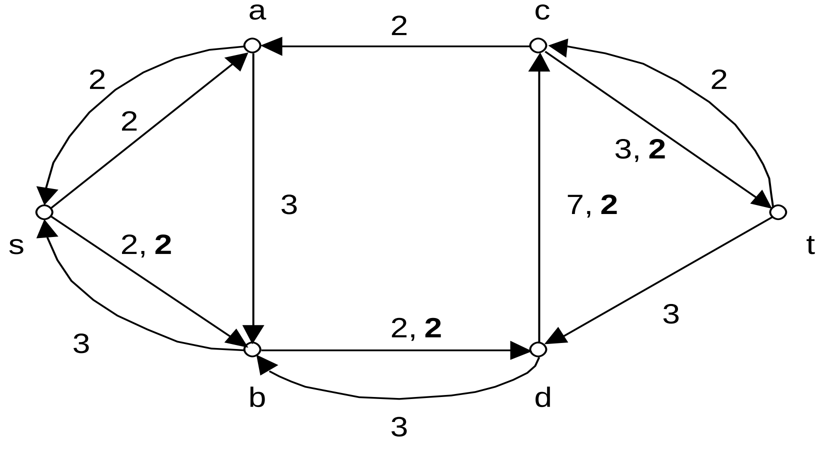
- aggiorna il flusso (in base al cammino aumentante trovato)



- costruisci la rete incrementale relativa al flusso

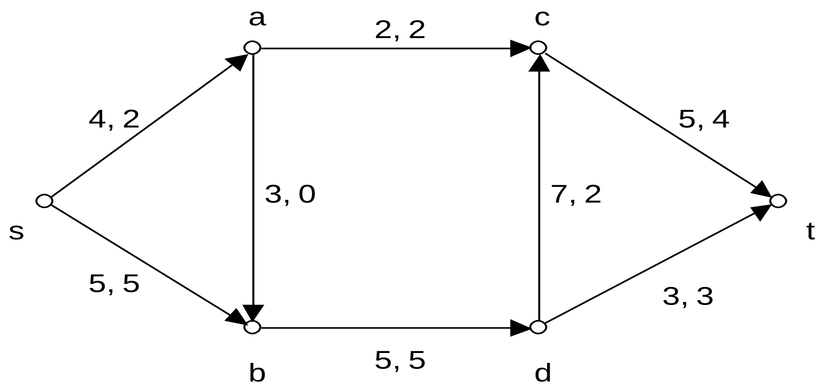


- individua cammino da *s* a *t* nella rete incrementale, e saturalo; un cammino è *s*, *b*, *d*, *c*, *t*; lo si satura “inviando” più flusso possibile, cioè 2 (l’arco *sb* è collo di bottiglia)

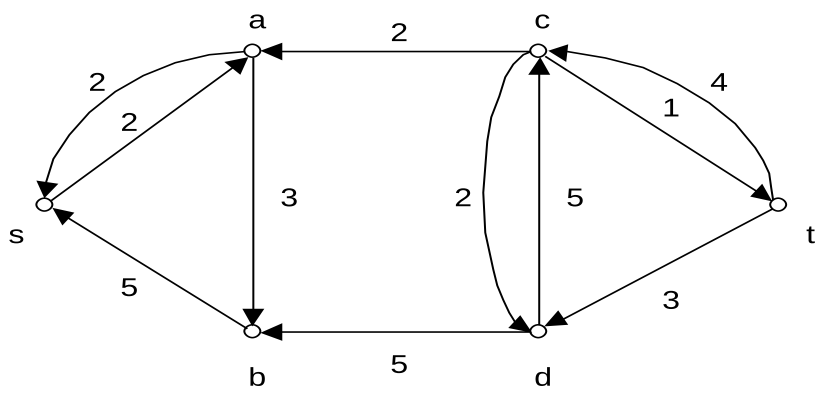


Iterazione 3

- aggiorna il flusso (in base al cammino aumentante trovato)

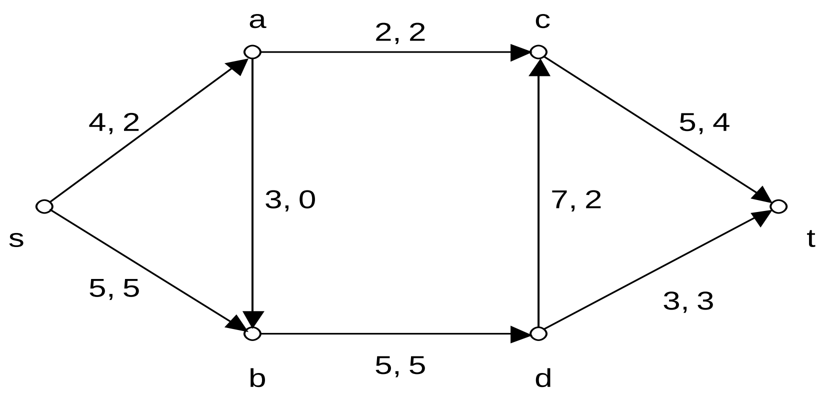


- costruisci la rete incrementale relativa al flusso



- individua cammino da *s* a *t* nella rete incrementale, e saturalo.

in questo caso non esiste alcun cammino da *s* a *t* nella rete incrementale; quindi il flusso corrente (cioè quello relativo all'ultimo aggiornamento – nella figura sotto) è ottimo; il suo valore è dato dalla somma delle componenti del flusso relative agli archi uscenti da *s*, cioè, 2 + 5 = 7.



□

Commenti finali sulle applicazioni

Il Teorema 5.3.2 risulta essere il più generale fra i così detti "teoremi Mengeriani", con riferimento al Teorema di Menger, dello studioso Karl Menger. Nel secolo scorso diversi teoremi [che introducevano relazioni di relazioni min-max] furono introdotti su diversi argomenti da diversi studiosi. A un certo punto si è verificato che tale gruppo di teoremi più o meno si implicavano a vicenda, in particolare, il Teorema 5.3.2 li implicava tutti compreso il Teorema di Menger.

Comunque riportiamo tale teorema (dove # = numero) soprattutto per la sua valenza applicativa:

<u>Teorema di Menger</u>: Data un rete semplice G [cioè: un grafo orientato $G = (V, E)$; $s, t \in V$; $q_{ij} = 1$ per ogni $(i,j) \in E$], si ha che:

:: max # di cammini arco-disgiunti da s a t = min # archi la cui rimozione disconnette s e t ;

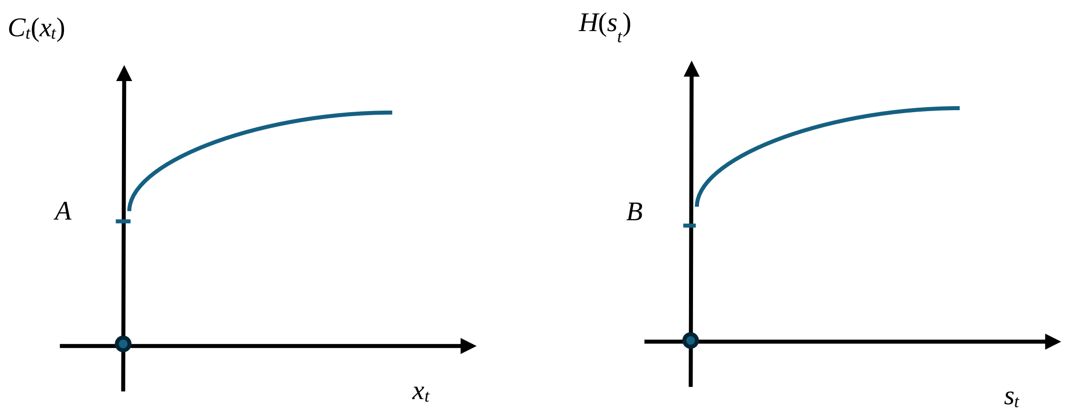
:: max # di cammini nodo-disgiunti da s a t = min # nodi la cui rimozione disconnette s e t. □

Si noti che il Teorema di Menger sembra avere una naturale applicazione nel controllo di una rete semplice, ad esempio, qual è il minimo numero di archi o di nodi da "monitorare" (ad esempio in cui fissare un posto di blocco) per controllare tutto ciò che scorre nella rete G da s a t. Un'altra simile applicazione è quella di calcolare il min # di archi o il min # di nodi bisognerebbe eliminare per disconnettere una rete semplice (per fare questo si possono considerare tutte le coppie di nodi di G, applicare per ognuna di esse il l'algoritmo di Ford-Fulkerson, per poi scegliere la "migliore" soluzione trovata). In accordo con il Teorema 5.3.2, tutti i valori indicato nel Teorema di Menger possono essere determinati risolvendo il problema del Massimo flusso in G [nel caso particolare in cui $q_{ij} = 1$ per ogni $(i,j) \in E$], tramite l'algoritmo di Ford e Fulkerson: in particolare, una volta determinato un flusso di valore massimo, allora si può facilmente individuare una sezione di capacità minima. □

5.4 Problema della Programmazione della produzione

<u>Definizione del problema</u>: Un'azienda deve produrre un bene, per soddisfare un piano di vendita, in un certo orizzonte temporale (ad esempio un mese) diviso in periodi (ad esempio settimane), che stabilisce le vendite da effettuare per ogni periodo. La capacità produttiva può variare da periodo a periodo, così come il costo unitario di produzione e il costo unitario di giacenza alla fine di ogni periodo nel magazzino di cui l'azienda può servirsi. Non ci sono giacenze in magazzino all'inizio, né si desidera averne alla fine dell'orizzonte temporale.
In dettaglio:

:: sia $\{1, \ldots, T\}$ l'insieme finito dei periodi dell'orizzonte temporale [noto] ;
:: per ogni $t \in \{1, \ldots, T\}$, sia d_t la vendita da effettuare nel periodo t [nota / stimata] ;
:: per ogni $t \in \{1, \ldots, T\}$, sia x_t il livello di produzione nel periodo t [da stabilire] ;
:: per ogni $t \in \{0, 1, \ldots, T\}$, sia s_t la giacenza alla fine del periodo t [da stabilire] ;
:: per ogni $t \in \{1, \ldots, T\}$, sia $C_t(x_t) = A_t\eta(x_t) + c_t(x_t)$ la *funzione di costo di produzione* nel periodo t [nota / stimata], dove: A_t è una costante che rappresenta un costo fisso [cioè, $\eta(x_t) = 0$ se $x_t = 0$ mentre $\eta(x_t) = 1$ se $x_t > 0$], $c_t(x_t)$ è una funzione *concava* in x_t ;
:: per ogni $t \in \{1, \ldots, T\}$, sia $H_t(x_t) = B_t\eta(s_t) + h_t(s_t)$ la *funzione di costo di stoccaggio* alla fine del periodo t [nota / stimata], dove: B_t è una costante che rappresenta un costo fisso [cioè, $\eta(s_t) = 0$ se $s_t = 0$, mentre $\eta(s_t) = 1$ se $s_t > 0$], $h_t(s_t)$ è una funzione *concava* in s_t .



Il problema è determinare le quantità x_t e s_t , per $t \in \{1, \ldots, T\}$, con il vincolo di garantire le vendite da effettuare per ogni periodo, in modo da minimizzare $\sum_{t=1}^{T}[\,C_t(x_t) + H_t(s_t)]$.

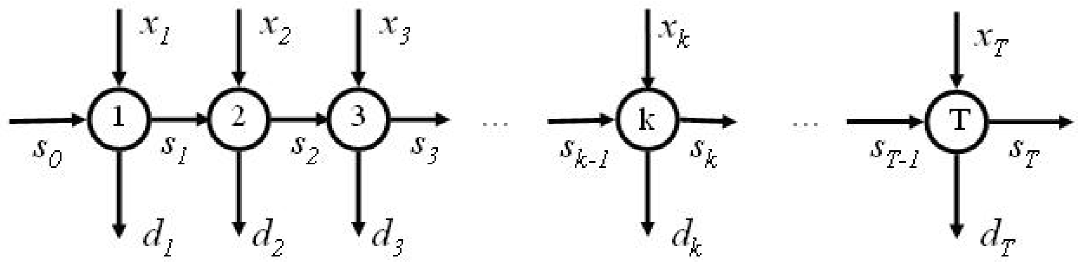
Nota bene: come specificato sopra, non ci sono giacenze in magazzino all'inizio, né si desidera averne alla fine dell'orizzonte temporale; così si può fissare $s_0 = 0$ e $s_T = 0$. □

Osservazioni

:: Tale problema non è formulabile come un problema di PL(I) dato che la funzione obiettivo comunque sarebbe una funzione non-lineare. Comunque osserviamo che: se $A_t = 0$ e se $c_t(x_t)$ è una funzione *lineare* (che è un caso specifico della convessa), allora $C_t(x_t)$ è una funzione lineare; se $B_t = 0$ e se $h_t(s_t)$ è una funzione *lineare* (che è un caso specifico della convessa), allora $H_t(x_t)$ è una funzione lineare; concludendo, se tali ipotesi valgono, allora tale problema è formulabile come un problema di PL (cfr. Capitolo 2, esempio C4).

:: Nella definizione di tale problema si considera un bene da "produrre", tuttavia, tale problema può considerare allo stesso modo un bene da "acquistare".

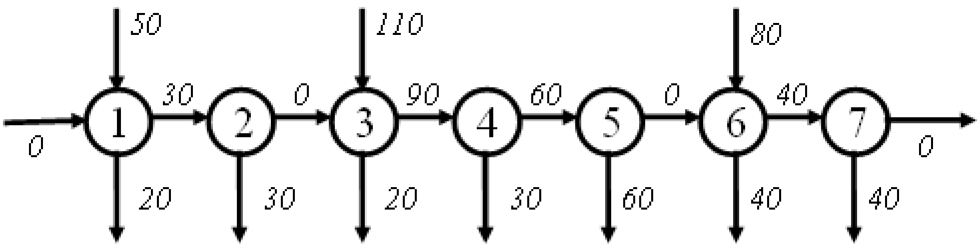
:: Infine osserviamo che, in realtà, le variabili da determinare sono solamente la variabili le quantità x_t (per $t \in \{1, \ldots, T\}$) poiché le quantità s_t (per $t \in \{1, \ldots, T\}$) sono univocamente determinate una volta determinate quantità x_t (per $t \in \{1, \ldots, T\}$); infatti, come anche illustrato di seguito, vale la relazione $s_{k-1} + x_k = s_k + d_k$, per ogni $k \in \{1, \ldots, T\}$. □



Il metodo si soluzione "ad hoc" che introduciamo è basato sulle seguenti due proprietà.

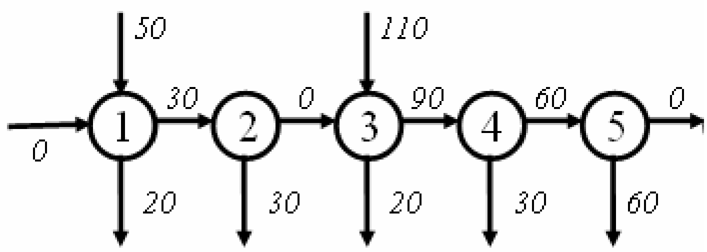
Proprietà 1: Esiste una soluzione ottima $\{x_1, \ldots, x_T\} \cup \{s_1, \ldots, s_{T-1}\}$ tale che, per ogni $t \in \{1, \ldots, T\}$, si ha che esattamente una delle due equazioni vale: $x_t = 0$ OPPURE $s_{t-1} = 0$. In altri termini esiste una soluzione ottima tale che i livelli di produzione si attivano (cioè sono > 0) se e solo se le rispettive giacenze precedenti sono nulle (cioè sono = 0). □

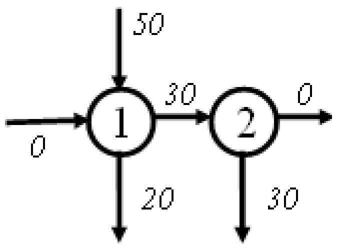
La seguente figura riporta una ipotetica soluzione ottima che soddisfa la Proprietà 1.



Proprietà 2: La soluzione ottima della Proprietà 1 è composta da sotto-soluzioni ottime. □

Per ogni $k \in \{1, \ldots, T\}$, sia $\mathbf{P}_k$ la versione del nostro problema ridotta all'orizzonte temporale $\{1, \ldots, k\}$, così che in particolare $\mathbf{P}_T$ è proprio il nostro problema.
Allora il significato della Proprietà 2 è il seguente. Si può osservare che la soluzione ottima di sopra partiziona l'insieme $\{1, \ldots, T\}$ in tre *blocchi*, cioè, $\{1, 2\} \cup \{3, 4, 5\} \cup \{6, 7\}$. Nello specifico, la soluzione ottima di sopra non solo fornisce la soluzione ottima per il problema $\mathbf{P}_7$, ma fornisce la soluzione ottima anche per i problemi $\mathbf{P}_5$ e $\mathbf{P}_2$. In questo senso la soluzione ottima della Proprietà 1 è composta da sotto-soluzioni ottime.

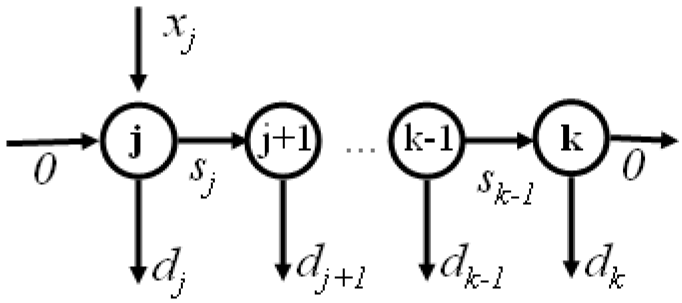




Per preparare il metodo di soluzione "ad hoc", introduciamo la seguente notazione, che focalizza i costi per ogni possibile blocco.
Per ogni j , $k \in \{1, \ldots, T\}$, con $j \leq k$, sia

$$M(j, k) = C_j(x_j) + \sum_{t=j}^{k} H_t(s_t) \text{ , con } x_j = \sum_{r=j}^{k} d_r \text{ e con } s_t = \sum_{r=t+1}^{k} d_r \text{ per } t \in \{j, \ldots, k-1\}$$

In altri termini, con riferimento al blocco $\{j, \ldots, k\}$, $M(j, k)$ è il costo necessario per produrre solo in j (partendo da giacenza nulla) e stare bene fino a k (arrivando a giacenza nulla).



Per ogni $k \in \{1, \ldots, T\}$, sia F_k il valore della soluzione ottima del problema $\mathbf{P}_k$.

Allora, dalla Proprietà 2, si ha che:

$$F_k = \min_{1 \leq j \leq k} \{F_{j-1} + M(j, k)\} \text{ , per ogni } k \in \{1, \ldots, T\}.$$

La precedente è una ***espressione ricorsiva*** del valore F_k , in funzione dei valori F_j , con $1 \leq j \leq k-1$. Così grazie a questa espressione ricorsiva, ponendo $F_0 = 0$, è possibile calcolare successivamente i valori $F_1, F_2, \ldots, F_T$. Questo è il nucleo del metodo “ad hoc”.

Algoritmo di Wagner-Whitin

L’algoritmo di Wagner-Whitin è in tre fasi:

FASE I : calcolo dei valori $M(j, k)$, per $j, k \in \{1, \ldots, T\}$, con $j < k$.

Tale calcolo si basa sulla formula introdotta sopra.

FASE II : calcolo dei valori $F_1, F_2, \ldots, F_T$

Tale calcolo si basa sulla espressione ricorsiva introdotta sopra.

FASE III : ricostruzione della soluzione ottima.

Tale ricostruzione su basa su un processo a ritroso dei calcoli effettuati nella Fase II; in particolare, si parte dal calcolo di F_T e si individua il periodo j tale che $F_T = F_{j-1} + M(j, T)$, individuando così il blocco $\{j , T\}$ (che è l’ultimo blocco) della soluzione ottima; dopo di che si itera questo procedimento, a partire dal periodo j, fino ad arrivare al periodo 1; in questa maniera si individuano tutti i blocchi della soluzione ottima, cioè, implicitamente tutti i valori della soluzione ottima $\{x_1, \ldots, x_T\} \cup \{s_1, \ldots, s_{T-1}\}$. □

Esercizio svolto

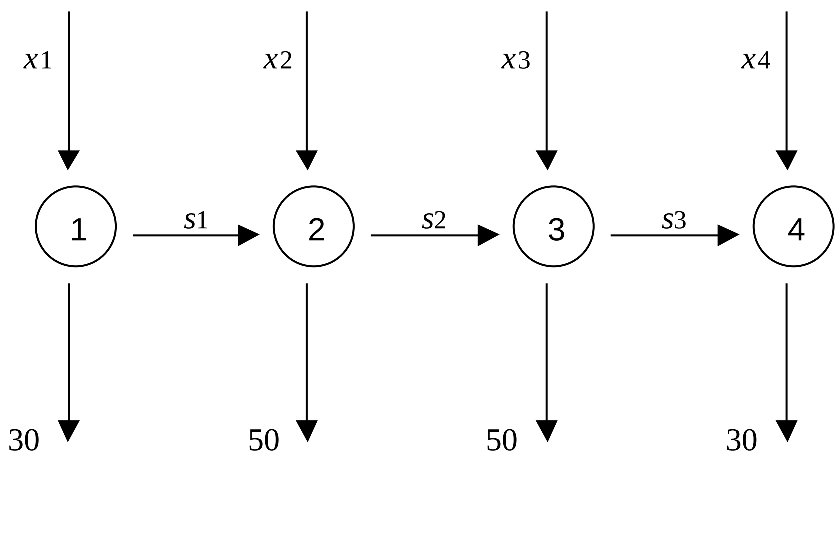
Un bene deve essere prodotto su un orizzonte temporale composto da 4 periodi. Per ogni periodo $t = 1,\ldots, 4$, è nota la domanda d_t da soddisfare nel periodo t. All’inizio del periodo 1 non ci sono giacenze in magazzino e non si vuole che ce ne siano alla fine del periodo 4.

Applicando il metodo di Wagner-Whitin, determinare per $t = 1,\dots, 4$, la quantità x_t prodotte nel periodo t e le quantità s_t in giacenza nel periodo t, in modo da minimizzare i costi totali e da soddisfare le domande al momento in cui vengono effettuate, sapendo che la funzione costo di produzione è $C_t = A_t w(x_t) + c_t x_t$, dove: $w(x_t) = 1$ se $x_t > 0$, e $w(x_t) = 0$ altrimenti; la funzione costo di stoccaggio è $H_t = h_t s_t$; inoltre in tabella sono riportati i dati relativi a quanto sopra.

Periodo	d_t	A_t	c_t	h_t
1	30	40	2	1
2	50	20	3	2
3	50	10	4	2
4	30	20	2	

Soluzione

La situazione può essere schematizzata come nella figura di seguito, in cui ad ogni periodo i sono associate le quantità d_i di domanda, x_i di produzione e s_i di stoccaggio (ciò che avanza a fine periodo). Quindi, note le quantità d_i e le funzioni costo, il problema è determinare le quantità x_i per ogni periodo $i = 1, \dots, 4$ e le quantità s_i per ogni periodo $i = 1, \dots, 3$.



Fase 1 : calcola i valori $M(i, j)$ per ogni coppia di periodi (i, j) con $i \leq j$

Il valore $M(i, j)$ è il costo totale che si dovrebbe pagare per soddisfare le domande dei periodi $i, i+1, \dots, j$, producendo solo nel periodo i (partendo da giacenza nulla e lasciando giacenza nulla).

$M(1,1) = 40 + 2 \cdot 30 = 100$

$M(1,2) = 40 + 2 \cdot (30+50) + 1 \cdot 50 = 250$

$M(1,3) = 40 + 2 \cdot (30+50+50) + 1 \cdot (50+50) + 2 \cdot 50 = 500$

$M(1,4) = 40 + 2 \cdot (30+50+50+30) + 1 \cdot (50+50+30) + 2 \cdot (50+30) + 2 \cdot 30 = 710$

$M(2,2) = 20 + 3 \cdot 50 = 170$

$M(2,3) = 20 + 3 \cdot (50+50) + 2 \cdot 50 = 420$

$M(2,4) = 20 + 3 \cdot (50+50+30) + 2 \cdot (50+30) + 2 \cdot 30 = 630$

$M(3,3) = 10 + 4 \cdot 50 = 210$

$M(3,4) = 10 + 4 \cdot (50+30) + 2 \cdot 30 = 390$

$M(4,4) = 20 + 2 \cdot 30 = 80$

<u>Fase 2</u> : calcola ricorsivamente F_k per $k = 1, \ldots, 4$

Il valore F_k è il valore di una soluzione ottima del problema ristretto ai primi k periodi. Quindi F_4 è il valore di una soluzione ottima cercata. Il calcolo avviene ricorsivamente, ponendo $F_0 = 0$ e applicando la formula $F_k = \min_{1 \leq j \leq k} \{F_{j-1} + M(j, k)\}$.

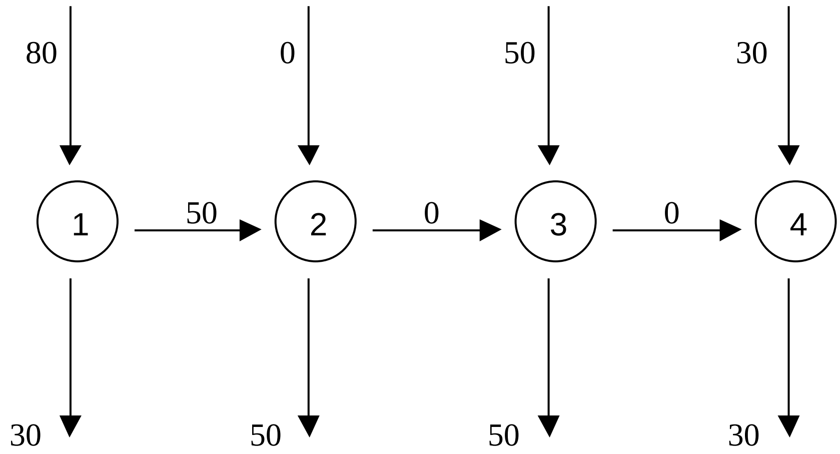
$F_0 = 0$
$F_1 = \min \{F_0 + M(1,1)\} = 100$
$F_2 = \min \{F_0 + M(1,2), F_1 + M(2,2)\} = \{250, 270\} = 250$
$F_3 = \min \{F_0 + M(1,3), F_1 + M(2,3), F_2 + M(3,3)\} = \{500, 520, 460\} = 460$
$F_4 = \min \{F_0 + M(1,4), F_1 + M(2,4), F_2 + M(3,4), F_3 + M(4,4)\} = \{710, 730, 640, 540\} = 540$

<u>Fase 3</u> : evidenzia la soluzione ottima trovata

Per fare ciò si parte da F_4 andando a ritroso, sostituendo (in base ai calcoli appena svolti) iterativamente i valori F_k fino ad arrivare a F_0, come di seguito:

$F_4 = F_3 + M(4,4) = F_2 + M(3,3) + M(4,4) = F_0 + M(1,2) + M(3,3) + M(4,4)$.

Si ottiene che la soluzione ottima è individuata da $M(1,2) + M(3,3) + M(4,4)$, cioè si produce: nel periodo 1 (soddisfacendo le domande dei periodi 1, 2), nel periodo 3 (soddisfacendo le domande del periodo 3), nel periodo 4 (soddisfacendo le domande del periodo 4). I valori ottimi di x_1, …, x_4 e s_1, …, s_3 possono quindi essere dedotti dal loro significato e dalle domande, come indicato in figura.



Commenti finali sulle applicazioni

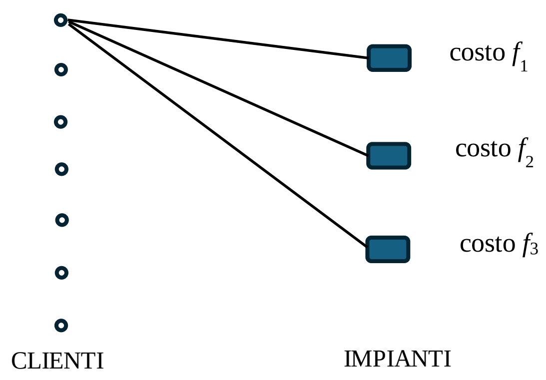
:: Il problema della Programmazione della produzione è un problema di Programmazione Dinamica, come si vede dalla Proprietà 2, in base a ciò che abbiamo commentato nel contesto del problema del Cammino di costo minimo. In particolare, si può osservare che l'ultima fase dell'algoritmo di Wagner-Whitin è dedicata alla ricostruzione della soluzione ottima, così come l'ultima fase dell'algoritmo di Dijkstra è dedicato alla ricostruzione della soluzione ottima.

:: Il problema della Programmazione della produzione, così come presentato sopra, non è una sorta di astrazione ideale, che però rimane lontana da applicazioni pratiche. Intanto sul riferimento [Sassano] si può trovare una simulazione in cui si evidenzia come l'applicazione del metodo di Wagner-Whitin avrebbe procurato un notevole risparmio. Inoltre, il problema della Programmazione della produzione rimane significativo anche se i costi di produzione e/o i costi di stoccaggio rimangono costanti per ogni periodo, e rimane significativo anche se la giacenza iniziale e la giacenza finale vengono fissati a un valore $Q > 0$ invece che al valore 0.

□

5.5 Problema della Localizzazione di impianti

Definizione del problema: Bisogna attivare un certo numero di impianti, da scegliere fra m potenziali impianti, che dovranno rifornire n clienti. Attivare il potenziale impianto j (per $j \in \{1, \ldots, m\}$) comporta un costo $f_j \geq 0$. Ogni impianto si intende abbia capacità di rifornimento illimitata. Ogni cliente sarà collegato ad un unico impianto: in particolare collegare un cliente i ad un impianto j comporta un costo c_{ij} (per $i \in \{1, \ldots, n\}$ e per $j \in \{1, \ldots, m\}$). Il problema è scegliere quali impianti attivare, collegando ogni cliente a un impianto, in modo da minimizzare i costi complessivi di attivazione degli impianti e di collegamento ai clienti.



Sia $N = \{1, \ldots, n\}$ l'insieme dei clienti e sia $M = \{1, \ldots, m\}$ l'insieme degli impianti.
Una soluzione del problema [candidata a essere soluzione ottima] può essere identificata da un sottoinsieme $T \subseteq M$, a cui si associa un collegamento di clienti a elementi di T automaticamente, cioè collegando ogni cliente $i \in N$ a un impianto $j \in T$ per il quale c_{ij} *è minimo*. In dettaglio, per ogni sottoinsieme $T \subseteq M$ e per ogni cliente $i \in N$, sia $c(T, i) = \min \{c_{ij} : j \in M\}$.
Allora il valore della soluzione identificata da $T \subseteq M$ è $Z(T) = \sum_{j \in T} d_j + \sum_{i \in N} c(T, i)$.

Esempio. Di seguito vediamo un'istanza del problema e un esempio di calcolo di $Z(T)$ per due possibili sottinsiemi $T \subseteq M$:

matrice c_{ij}

		IMPIANTI				
		1	2	3	4	5
	1	12	13	6	0	1
	2	8	4	9	1	2
	3	2	6	6	1	2
CLIENTI	4	3	5	2	10	8
	5	8	0	5	10	8
	6	2	0	3	4	1
vettore f		7	9	5	4	7

Assumiamo che $T = \{4, 5\}$, cioè, che vengono attivati solo gli impianti 4 e 5;
allora $Z(T) = (4 + 7) + (0 + 1 + 0 + 8 + 8 + 1) = 29$.
Assumiamo che $T = \{1, 3, 5\}$, cioè, che vengono attivati solo gli impianti 1, 4, 5;
allora $Z(T) = (4 + 1 + 7) + (1 + 2 + 1 + 2 + 5 + 1) = 24$. □

Metodi euristici: algoritmo greedy, algoritmo di ricerca locale

• Algoritmo greedy

Di seguito, applicheremo l' *algoritmo greedy* per la soluzione del nostri problema, e vedremo che l'algoritmo greedy è un caso particolare di *algoritmo di ricerca locale*.
Con riferimento ai problemi *difficili* (cfr. Appendice 1): i metodi euristici sono metodi che determinano una soluzione ammissibile di un problema in maniera abbastanza semplice, senza fornire in genere garanzie sulla "bontà" della soluzione ammissibile determinata, cioè sulla distanza dalla soluzione ottima; comunque i metodi euristici sono basati su considerazioni di buon senso, che possono dare garanzie parziali sulla "bontà" della soluzione ammissibile determinata, come vedremo di seguito per il caso dell'algoritmo di ricerca totale.
L'algoritmo greedy è definito (nel caso del nostro problema) come segue:

Algoritmo greedy

Inizializzazione: poni: $i := 1$; $T_0 := \emptyset$; $Z(T_0) := \infty$.

Iterazione i-esima

$u_i = \text{argmin } \{Z(T_{i-1} \cup \{u\}) : u \in M \setminus T_{i-1}\}$;

se $Z(T_{i-1} \cup \{u_i\}) \geq Z(T_{i-1})$, allora STOP [T_{i-1} è la *soluzione greedy*];

altrimenti poni $T_i := T_{i-1} \cup \{u_i\}$;

se $T_i = M$, allora STOP [T_{i-1} è la *soluzione greedy*];

altrimenti poni $i := i + 1$ e vai a Iterazione i-esima. □

Un commento: l'algoritmo greedy costruisce la soluzione greedy, sia T, aggiungendo ad ogni iterazione un elemento in T (partendo dall'insieme vuoto, cioè, da $T_0 := \emptyset$); ad ogni iterazione, l'algoritmo aggiunge alla soluzione corrente l'impianto che converrebbe aggiungere, nel caso in cui a tale soluzione corrente fosse possibile solo un unico impianto; in questo senso l'algoritmo è detto greedy, cioè "ingordo", esso sceglie ciò gli sembra meglio scegliere avendo una visione solo a breve termine (nel senso di un unico impianto); l'algoritmo si ferma quando la soluzione trovata, a una certa iterazione, non è migliore della soluzione trovata all'iterazione precedente; oppure l'algoritmo si ferma naturalmente nel caso in cui tutti gli impianti sono stati inseriti nella soluzione corrente.

Esercizio svolto

Bisogna attivare un certo numero di impianti, da scegliere fra m potenziali impianti, che dovranno rifornire n clienti. Attivare il potenziale impianto j costa f_j. Ogni impianto si intende abbia capacità di rifornimento illimitata. Ogni cliente sarà collegato ad un unico impianto; in particolare collegare un cliente i ad un impianto j comporta un costo c_{ij}. Il problema è scegliere quali impianti attivare in modo da minimizzare i costi complessivi di attivazione e di assegnazione ai clienti.

Calcolare una soluzione "greedy" di tale problema per la seguente istanza:

matrice c_{ij}

		IMPIANTI				
		1	2	3	4	5
	1	0	1	7	0	4
	2	1	0	3	0	3
	3	3	7	1	4	2
CLIENTI	4	1	7	4	0	7
	5	1	0	3	2	4
	6	4	4	1	4	3
	7	7	1	2	1	0
vettore f		7	9	5	4	7

Soluzione

Inizializzazione

$i = 1$; $T_0 = \emptyset$; $Z(T_0) = \infty$.

Iterazione 1

$Z(\{1\})$	$Z(\{2\})$	$Z(\{3\})$	$Z(\{4\})$	$Z(\{5\})$
24	29	26	15	30

Ad esempio $Z(\{1\}) = (0 + 1 + 3 + 1 + 1 + 4 + 7) + 7 = 24$.

Il minimo è $Z(\{4\}) = 15$;

$15 < \infty \Rightarrow i = 2$, $T_1 = \{4\}$.

Iterazione 2

$Z(\{4,1\})$	$Z(\{4,2\})$	$Z(\{4,3\})$	$Z(\{4,5\})$
20	22	14	18

Ad esempio $Z(\{4,1\}) = (0 + 0 + 3 + 0 + 1 + 4 + 1) + (7 + 4) = 20$.

Il minimo è $Z(\{4,3\}) = 14$;

$14 < 15 \Rightarrow i = 3$, $T_2 = \{4,3\}$.

Iterazione 3

$Z(\{4,3,1\})$	$Z(\{4,3,2\})$	$Z(\{4,3,5\})$
22	21	20

Ad esempio $Z(\{4,3,1\}) = (0 + 0 + 1 + 0 + 1 + 1 + 1) + (7 + 5 + 4) = 22$.

Il minimo è $Z(\{4,3,5\}) = 20$;

$20 \geq 14 \Rightarrow$ STOP

La soluzione greedy è $T_2 = \{4,3\}$, con $Z(\{4,3\}) = 14$. □

• **Algoritmo di ricerca locale**

L'algoritmo greedy fa parte delle così dette "euristiche di ricerca locale".

Definizione **(***problema di Ottimizzazione Combinatoria***)** : Dati un insieme E finito, una famiglia Φ di sottoinsiemi di E, e una funzione $\omega : \Phi \to \mathbb{R}$, trovare **min** $\{\omega(F) : F \in \Phi\}$. □

Osservazione

Un problema di PLI 0-1 (cioè, con variabili decisionali che sono variabili binarie, ad esempio il problema del Riempimento) è un problema di Ottimizzazione Combinatoria [mentre un problema di Ottimizzazione Combinatoria può non essere un problema di PLI 0-1, infatti, basata che la funzione ω di sopra non sia lineare]. Così, tutto quello che stiamo per vedere con riferimento a un problema di Ottimizzazione Combinatoria, vale anche per un problema di PLI 0-1. □

Un approccio generale per trovare una soluzione ammissibile di un problema di Ottimizzazione Combinatoria è il seguente:

per ogni elemento $F \in \Phi$, si definisce un *intorno di F* [all'interno di Φ], denotato con $\mathcal{N}(F)$, che è un insieme di elementi di Φ che sono diciamo "vicini" a F; da ciò si ottiene $\{ \mathcal{N}(F) : F \in \Phi\}$ che è detto *sistema di intorni* relativo agli elementi di Φ.

Una soluzione $F^* \in \Phi$ [del problema] si dice *ottimo locale* se $\omega(F^*) \leq \omega(F)$ per ogni $F \in \mathcal{N}(F^*)$.

Una soluzione ottima [del problema] si dice *ottimo globale*.

Allora, dato un sistema di intorni, si può stabilire che l'obiettivo è determinare un ottimo locale; in altri termini, se determinare una soluzione ottima del problema risulta particolarmente complicato [anche per quelle che sono le valutazioni del decisore], allora ci si può "accontentare" di determinare un ottimo locale. Ora un *algoritmo di ricerca locale* è un algoritmo che determina un ottimo locale.

Algoritmo di ricerca locale

Inizializzazione: poni: $i := 1$; scegli $F_0 \in \Phi$.

Iterazione i-esima

$F_i = \text{argmin}\,\{\omega(F) : F \in \mathcal{N}(F_{i-1})\}$;

se $\omega(F_i) \geq \omega(F_{i-1})$, allora STOP [T_{i-1} è *ottimo locale*];

altrimenti poni $i := i + 1$ e vai a Iterazione i-esima. □

La “bontà” di un algoritmo di ricerca locale dipende dalla scelta di $F_0 \in \Phi$ e dalla definizione del sistema di intorni. Si richiede in genere esperienza per ottenere buoni risultati. Comunque, in generale, il decisore si trova di fronte a un trade-off [cioè a un compromesso] fra la qualità dell’ottimo locale e la complessità della computazione.

Osservazione

Di seguito verifichiamo che l’algoritmo greedy è effettivamente un algoritmo di ricerca locale; a tal fine, basta verificare che esso è basato sul seguente sistema di intorni, definito come segue per ogni $T \subseteq M$:

$\mathcal{N}(T) = \{T \cup \{u\} : u \in M \setminus T\}$. □

Appendice 1: complessità di calcolo e supporto alle decisioni

Nota 1: Come abbiamo già visto – in modo approssimativo per il nostro solo scopo, rimandando il lettore a testi di Informatica, per una trattazione formale – ricordiamo di seguito alcuni cenni di complessità computazionale. I problemi di Programmazione Matematica sono divisi [nei limiti del possibile] in *facili* e *difficili*, in base alla complessità computazionale del migliore algoritmo di soluzione [definito a priori] noto per risolvere una generica istanza del problema:
I problemi *facili* – detti anche "polinomiali" – comprendono quelli di PL; ad esempio il problema della pianificazione dei progetti è facile; inoltre anche certi problemi, che non sono inizialmente modellati come PL, sono infine riconducibili alla PL [e quindi sono *facili*] tramite risultati sulle matrici TUM; come, ad esempio, i seguenti problemi: Assegnamento, Trasporti, Cammino di costo minimo, Massimo Flusso. Riguardo lo studio dei problemi di ottimizzazione formulabili come problemi di PL [i quali possono comunque essere risolti all'ottimo tramite metodi usuali per la PL, cioè, mediante l'Algoritmo del Simplesso], in genere vengono introdotti/studiati possibili "metodi esatti" per trovare una soluzione ottima del problema, senza ricorrere quindi ai metodi usuali per la PL; questo perché comunque è utile disporre di metodi veloci, soprattutto nel caso di grandi istanze del problema.
I problemi *difficili* – detti anche "NP-hard" – comprendono quelli di PLI; ad esempio il problema di localizzazione degli impianti è difficile. Riguardo lo studio dei problemi di ottimizzazione formulabili come problemi di PLI [i quali possono comunque essere risolti all'ottimo tramite metodi usuali per la PLI, cioè, mediante l'Algoritmo del Branch and Bound], in genere vengono introdotti/studiati "metodi euristici" per trovare una soluzione presumibilmente buona ma non necessariamente ottima del problema, senza ricorrere quindi ai metodi usuali per la PLI; questo poiché un metodo usuale per la PLI può essere lento, anche per piccole istanze del problema, e così il decisore (se lo ritiene) può utilizzare un "metodo euristico" che, pur garantendo una soluzione presumibilmente buona ma non necessariamente ottima del problema, ha il pregio di essere veloce.

Nota 2: Come abbiamo già visto, modellare e risolvere un problema di ottimizzazione come un problema di Programmazione Matematica ha come scopo principale quello di essere un "supporto alle decisioni"; infatti spesso il modello che si produce è solo una approssimazione della realtà; in particolare, la soluzione ottenuta può dare indicazioni sia su come procedere riguardo la decisione direttamente legata al problema, sia su come procedere riguardo decisioni indirettamente legate al problema [ad esempio abbiamo visto che i *prezzi ombra*, nel contesto di una produzione che avviene componendo risorse, individuano direzioni e condizioni per un eventuale sviluppo di quella produzione].

Appendice 2 : esempio di risoluzione di un problema di PL con Excel

Vediamo come risolvere un problema di PL con Excel. Riprendiamo un esercizio già visto.

Un'azienda vinicola desidera produrre due tipi di vino: uno da tavola, uno da dessert. Il profitto che l'azienda trae dalla produzione di una unità di vino da tavola è 3, mentre dalla produzione di una unità di vino da dessert è 7. Tale produzione necessita di una particolare combinazione di due tipi di uva: diciamo di tipo A e di tipo B rispettivamente. Per produrre 1 unità di vino da tavola, si ha bisogno di 3 unità di uva di tipo A e di 2 unità di uva di tipo B. Per produrre 1 unità di vino da dessert, si ha bisogno di 1 unità di uva di tipo A e di 4 unità di uva di tipo B. Infine l'azienda ha a disposizione 1000 unità di uva di tipo A e 400 di uva di tipo B. Il problema è determinare le quantità di vino da tavola e da dessert da produrre in modo da massimizzare il profitto totale.

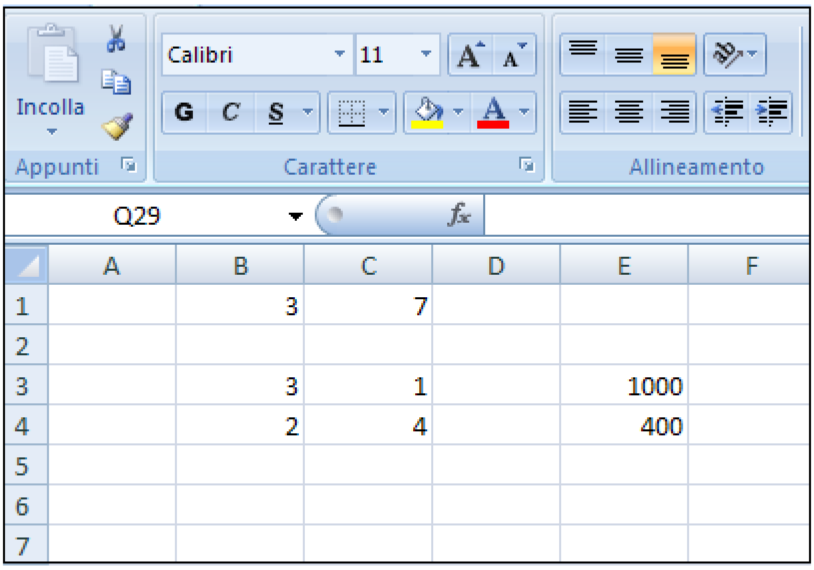
Variabili decisionali: x_1, x_2

x_1 = quantità di vino da tavola prodotta

x_2 = quantità di vino da dessert prodotta

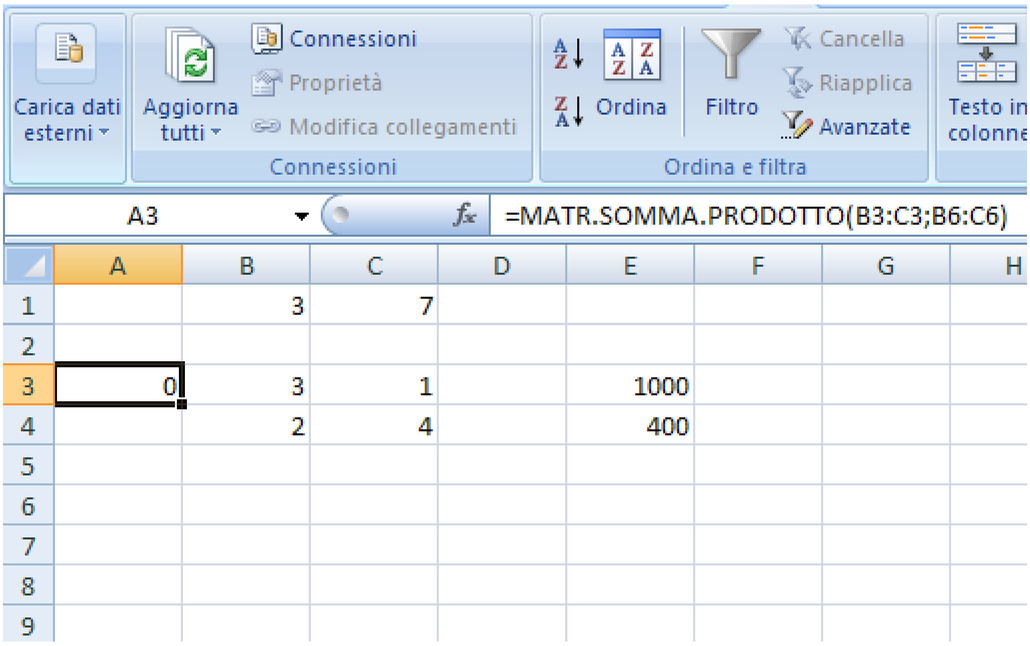
$$
(1) \qquad \max \quad 3x_1 + 7x_2
$$
$$
3x_1 + x_2 \leq 1000
$$
$$
2x_1 + 4x_2 \leq 400
$$
$$
x_1, x_2 \geq 0
$$

Inseriamo su Excel i dati del problema nel seguente modo, assegnando inoltre alle variabili le caselle B6 e C6.

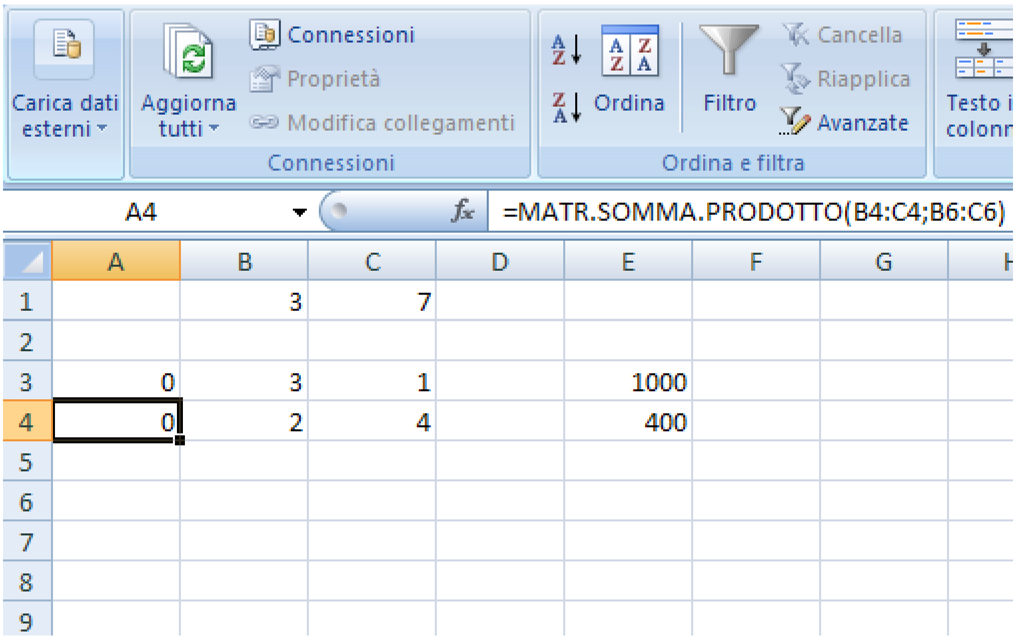


	A	B	C	D	E	F
1		3	7			
2						
3		3	1		1000	
4		2	4		400	
5						
6						
7						

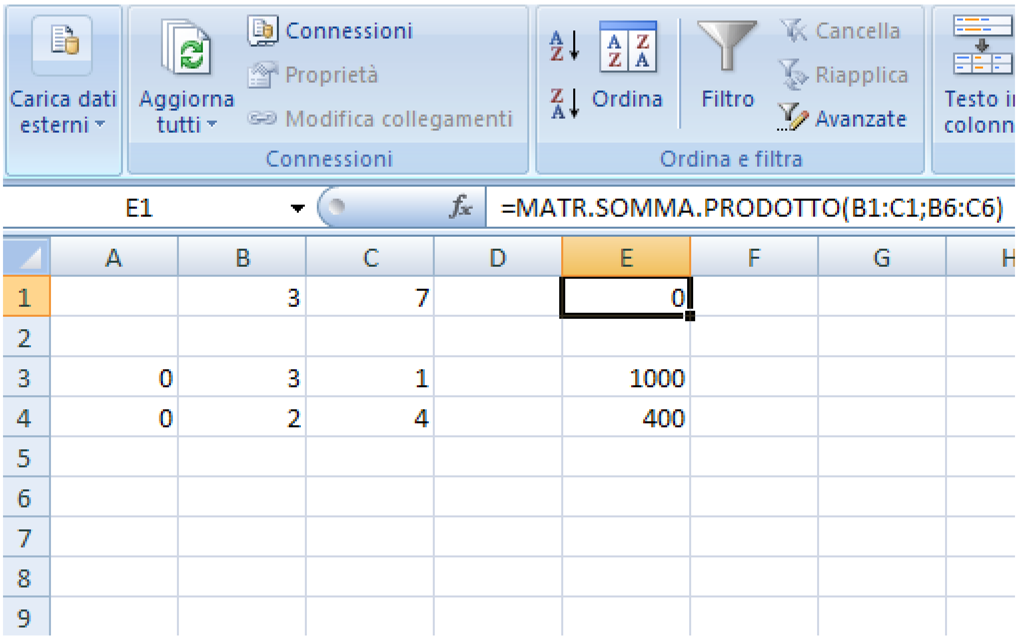
Nella cella A3 inseriamo la matrice somma prodotto (B3 :C3;B6 :C6), cioè il valore $3x_1 + x_2$ relativo alla prima disuguaglianza.



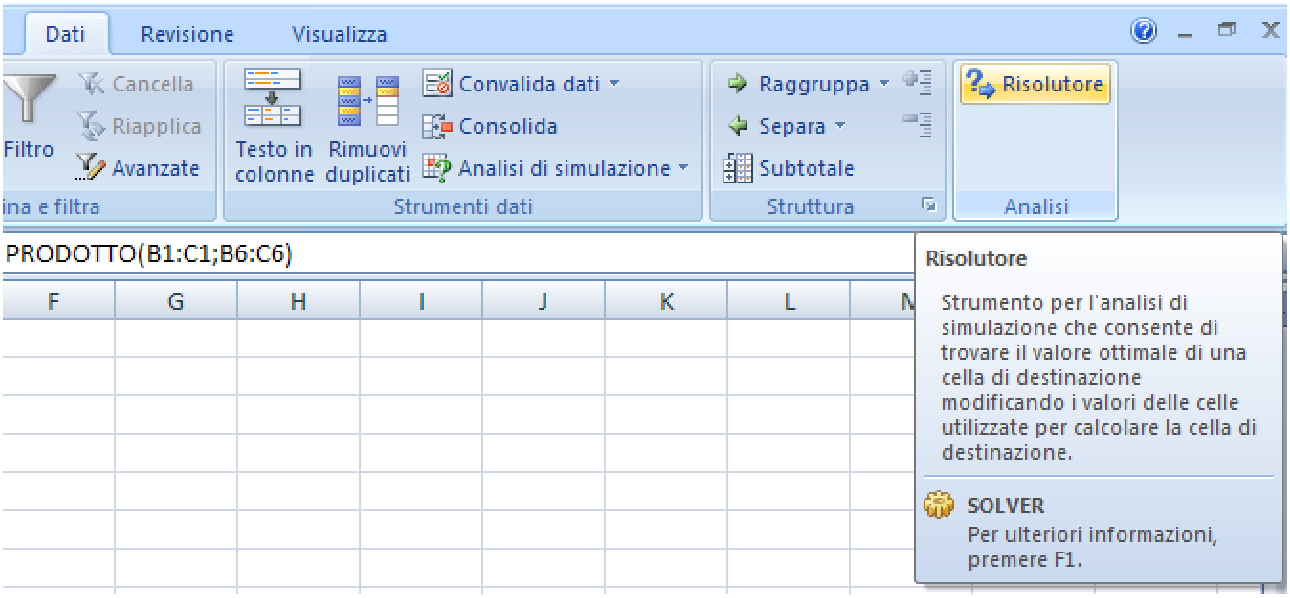
Nella cella A4 procediamo in modo analogo, con riferimento alla seconda disuguaglianza.



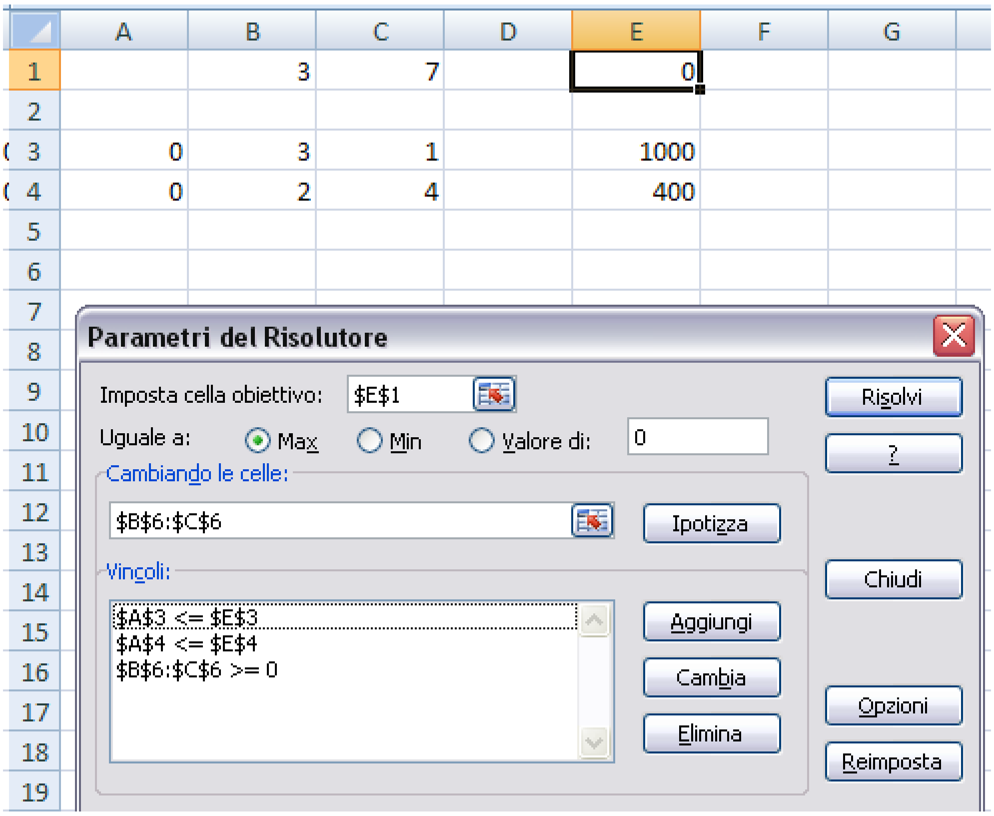
Nella cella E1 inseriamo infine la matrice somma prodotto (B1 :C1;B6 :C6), cioè il valore $3x_1 + 7x_2$ della funzione obiettivo.



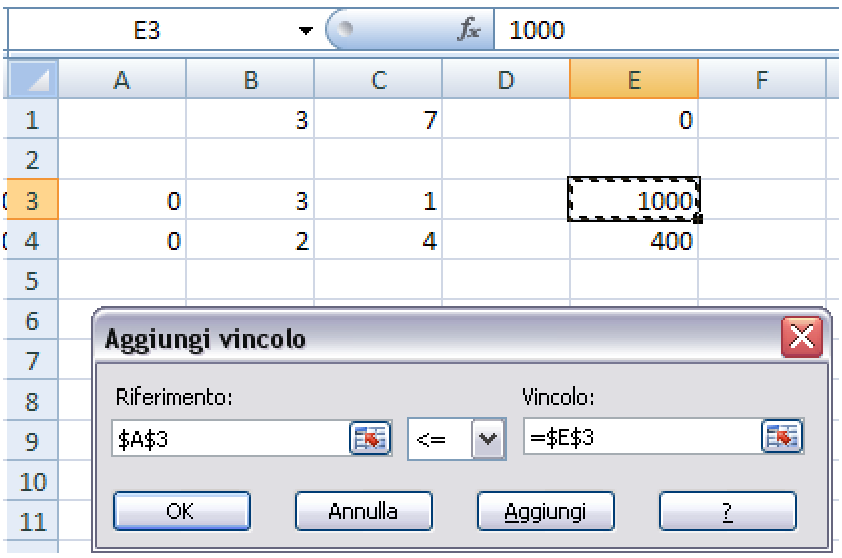
Ora possiamo utilizzare uno strumento fornito da Excel: il <u>Risolutore</u>.



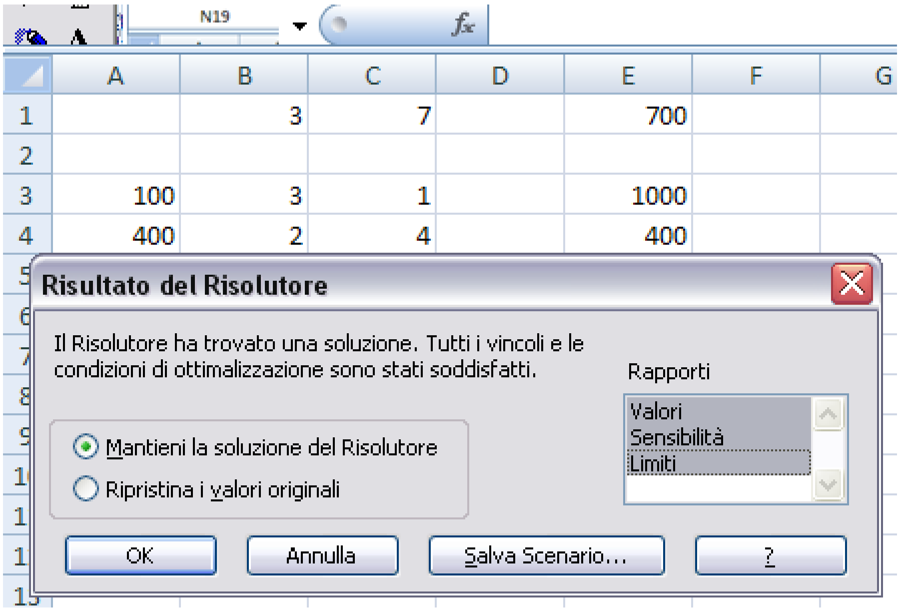
Cliccando su di esso è possibile inserire i parametri del problema :



In "Imposta cella obiettivo" (cioè la cella che indica il valore della funzione obiettivo) scriviamo la cella E1; in tale cella comparirà il valore ottimo della funzione obiettivo. Poi scegliamo l'opzione "max", dato che il problema è di "max". In "Cambiano le celle" (cioè le celle che indicano le variabili) scriviamo le celle B6 e C6; in tali celle compariranno i valori ottimi delle variabili, cioè i valori di x_1 e x_2. Infine in "Vincoli" inseriamo tutti i vincoli, mediante l'uso di "Aggiungi", come illustrato di seguito rispetto al solo primo vincolo.



Poi andiamo su “Opzioni” (tra i parametri del Risolutore), spuntiamo l’opzione “Presupponi modello lineare” e diamo l’OK.
Ora possiamo procedere al calcolo della soluzione, cliccando su “Risolvi”.
A questo punto è possibile anche richiedere il rapporto sui valori, sulla sensibilità e sui limiti (come illustrato di seguito) cliccando su ognuno di essi.



Diamo l’OK e finalmente abbiamo i risultati che attendevamo.

La soluzione ottima è: $x^*_1 = 0$, $x^*_2 = 100$, con valore 700,

	A	B	C	D	E	F	G	H
1		3	7		700			
2								
3	100	3	1		1000			
4	400	2	4		400			
5								
6		0	100					
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								
25								

Rapporto valori 1 | Rapporto sensibilità 1 | Rapporto limiti 1 | **Foglio1**

In basso possiamo cliccare sui vari rapporti per avere delle indicazioni più precise sui risultati. Vediamo in particolare il “Rapporto sensibilità”: in esso possiamo leggere quanto segue.

1. I prezzi ombra delle risorse, che sono la soluzione ottima del problema duale:

$$\begin{aligned} \min \quad & 1000\, y_1 + 400\, y_2 \\ & 3y_1 + 2y_2 \geq 3 \\ & y_1 + 4y_2 \geq 7 \\ & y_1, y_2 \geq 0 \end{aligned}$$

cioè, $y^*_1 = 0$, $y^*_2 = 1{,}75$, con valore 700.

Celle variabili

Cella	Nome	Valore finale	ridotto Costo	oggettivo Coefficiente	consentito Incremento	consentito Decremento
B6		0	-0,5	3	0,5	1E+30
C6		100	0	7	1E+30	1

Vincoli

Cella	Nome	Valore finale	ombra Prezzo	Vincolo a destra	consentito Incremento	consentito Decremento
A3		100	0	1000	1E+30	900
A4		400	1,75	400	3600	400

2. L'analisi di post-ottimalità relativa ai coefficienti della funzione obiettivo; la soluzione ottima di base resterà la stessa a condizione che: il prezzo unitario di vendita del vino da tavola aumenti al massimo di 0,5, e il prezzo unitario di vendita del vino da dessert diminuisca al massimo di 1.

Celle variabili

Cella	Nome	Valore finale	ridotto Costo	oggettivo Coefficiente	consentito Incremento	consentito Decremento
B6	Tavola	0	-0,5	3	0,5	1E+30
C6	Dess.	100	0	7	1E+30	0,999999999

Vincoli

Cella	Nome	Valore finale	ombra Prezzo	Vincolo a destra	consentito Incremento	consentito Decremento
A3	uva A	100	0	1000	1E+30	900
A4	uva B	400	1,75	400	3600	400

3. L'analisi di post-ottimalità relativa ai termini noti; la soluzione ottima di base resterà la stessa a condizione che: la disponibilità di uva A diminuisca al massimo di 900, e la disponibilità di uva B aumenti al massimo di 3600 e diminuisca al massimo di 400.

Celle variabili

Cella	Nome	Valore finale	ridotto Costo	oggettivo Coefficiente	consentito Incremento	consentito Decremento
B6	Tavola	0	-0,5	3	0,5	1E+30
C6	Dess.	100	0	7	1E+30	0,999999999

Vincoli

Cella	Nome	Valore finale	ombra Prezzo	Vincolo a destra	consentito Incremento	consentito Decremento
A3	uva A	100	0	1000	1E+30	900
A4	uva B	400	1,75	400	3600	400

In accordo con quanto introdotto nell'esempio di interpretazione economica della dualità, presentiamo una possibile parziale lettura di questi dati.

Consideriamo la risorsa uva B: il suo prezzo ombra è 1,75. Allora:

• è conveniente acquistare l'uva B a un prezzo unitario p < 1,75; ciò garantirebbe un aumento del profitto (cioè del valore della soluzione ottima) pari a 1,75 – p per unità di uva B; questo vale però solo per una quantità non superiore a 3600 unità, dato che superato tale valore la soluzione ottima di base cambia.

• è conveniente vendere l'uva B a un prezzo unitario p > 1,75; ciò garantirebbe un aumento del profitto (cioè del valore della soluzione ottima) pari a p – 1,75 per unità di uva B; questo vale però solo per una quantità non superiore a 400 unità, dato che superato tale valore la soluzione ottima di base cambia. □