M. Ghirardi • A. Grosso • G. Perboli

ESERCIZI SVOLTI
DI
RICERCA OPERATIVA

PROGETTO LEONARDO
ESCULAPIO · BOLOGNA

ISBN 978-88-7488-322-6

Prima edizione: Settembre 2005
Seconda edizione: Agosto 2009
Ristampa: Settembre 2010

Responsabile produzione: *Alessandro Parenti*
Redazione: *Gabriella Gatti e Giancarla Panigali*
Stampa e confezione: Global Print - Gorgonzola (MI)

SOCIETÀ EDITRICE ESCULAPIO

COLLANA PROGETTO LEONARDO

40131 Bologna - Via U. Terracini 30 - Tel. 051-63.40.113 - Fax 051-63.41.136

www.editrice-esculapio.it

Indice

Il presente volume raccoglie in forma organica il materiale prodotto dagli autori durante alcuni anni accademici per le esercitazioni di corsi di Ricerca Operativa e di Ottimizzazione Combinatoria svolti presso le facoltà di Ingegneria del Politecnico di Torino. Esso è soprattutto un eserciziario e quindi, a parte alcuni brevi richiami, non fornisce un impianto teorico completo per le tecniche trattate e non può chiaramente sostituire un libro di testo. Notazioni e convenzioni sono largamente tratte da Elementi di Ricerca Operativa *di F. Della Croce e R. Tadei, del quale il presente libro è un complemento.*

Parte del materiale, in particolare del capitolo 1, è stato elaborato a partire da un insieme di esercizi sviluppati in precedenza da F. Della Croce, con il quale gli autori sono in debito. Un sentito ringraziamento anche a R. Tadei, per il continuo supporto ed i consigli forniti.

Si ringraziano anticipatamente coloro che, rilevando questi errori, volessero gentilmente segnalarli, in modo da poterli eliminare in una eventuale riedizione del libro.

Marco Ghirardi	Andrea Grosso	Guido Perboli
marco.ghirardi@polito.it	grosso@di.unito.it	guido.perboli@polito.it

Progetto Leonardo – Editrice Esculapio – Bologna

1

Modelli di programmazione lineare

Introduzione

Molti problemi di interesse pratico si prestano ad essere descritti e risolti come *modelli di programmazione matematica*. Un modello (o *programma*) è la descrizione di un problema che richiede di massimizzare (o minimizzare) una funzione di costo o profitto su un certo dominio. La scrittura usuale è

$$\max \; z = f(x) \qquad \text{(oppure: } \min \; z = f(x)) \tag{1.1}$$

s.t.

$$g_i(x) \begin{cases} \leq b_i \\ = b_i \\ \geq b_i \end{cases} \qquad i = 1, \ldots, m, \tag{1.2}$$

$$x = (x_1, \ldots, x_n) \in X \subseteq \mathbb{R}^n. \tag{1.3}$$

In un modello sono presenti:

- una serie di *variabili di controllo* in funzione delle quali viene formulato ogni altro elemento del modello; queste variabili, almeno in parte, corrispondono alle quantità agendo sulle quali la soluzione verrà implementata;
- una *funzione obiettivo* $f(x)$ che determina un costo o profitto legato alla soluzione;
- una o più serie di *vincoli,* che correlano tra loro i valori delle variabili, imponendo condizioni di fisica realizzabilità e/o requisiti particolari richiesti alla soluzione.

Tra i modelli di programmazione matematica hanno particolare rilievo i modelli di *programmazione lineare,* nei quali la $f(x)$ e le $g_i(x)$ sono espressioni *lineari.* Un modello di programmazione lineare è quindi esprimibile sempre come

$$\max\ z = \sum_{j=1}^{n} c_j x_j \tag{1.4}$$

s.t.

$$\sum_{j=1}^{n} a_{ij} x_j \begin{cases} \leq b_i \\ = b_i \\ \geq b_i \end{cases} \qquad i = 1, \ldots, m, \tag{1.5}$$

$$x = (x_1, \ldots, x_n) \in X \subseteq \mathbb{R}^n. \tag{1.6}$$

I campi di esistenza delle variabili x_j sono di solito di tipo continuo (spesso non negativo), intero non negativo ($x_j \in \mathbb{Z}^+$), oppure binario ($x_j \in \{0,1\}$) a seconda del tipo di decisione che tali variabili modellano.
La particolarità dei modelli lineari è legata alla loro maggiore semplicità, che li rende più facilmente risolvibili rispetto ai modelli non lineari; in effetti sono ormai disponibili pacchetti software commerciali in grado di risolvere in modo efficiente programmi lineari di notevoli dimensioni (intese come quantità di variabili e di vincoli). Questo rende spesso preferibile, per la risoluzione di un problema, lo sviluppo di un modello lineare anche quando un modello non lineare potrebbe essere più compatto.
Lo sviluppo di un modello di programmazione lineare parte dall'analisi di una situazione reale (più o meno schematizzata) e, in modo simile a quanto accade nello sviluppo di una procedura software, richiede di identificare le variabili di controllo ed i rispettivi domini, i vincoli e la funzione obiettivo. Non ci sono regole rigide da seguire: il modello finale nasce spesso — in particolare nel caso di situazioni complesse — per raffinamenti successivi.

1.1 Modelli di base: mix e allocazione

ESEMPIO 1.1. L'acciaieria PLASTIK deve evadere un ordine di 1000 tonnellate di acciaio INOX. Per questa produzione servono manganese (almeno l'1% in peso), cromo (almeno il 18%) e molibdeno (almeno il 2%).
I fornitori di metalli non ferrosi vendono — per esigenze di mercato — questi prodotti in tre tipi di confezioni differenti. La prima confezione contiene 2 Kg. di manganese, 2 Kg. di cromo e 1 Kg. di molibdeno e costa 10 euro. La seconda confezione contiene 2 Kg. di manganese, 3 Kg. di cromo e 1 Kg. di molibdeno e costa 15 euro. La terza

confezione contiene 1 Kg. di manganese, 2 Kg. di cromo e 5 Kg. di molibdeno e costa 20 euro.
Formulare il modello di Programmazione Lineare per minimizzare il costo di acquisto delle confezioni.

Soluzione Le variabili più naturali sono x_1, x_2, x_3, dove x_i = numero di confezioni di tipo i acquistato. A volte può non essere evidente quale sia la scelta di variabili "più naturale". Una buona regola euristica è spesso la seguente: una definizione di variabili è soddisfacente quando essa permette di scrivere in modo *semplice* la funzione obiettivo (o comunque i vincoli più significativi) del modello. Ad esempio, in questo caso usare variabili che rappresentano le quantità di materiali (manganese, cromo, molibdeno) acquistate anziché le confezioni non sarebbe soddisfacente, in quanto la funzione obiettivo risulterebbe molto difficile da esprimere.
Con la scelta di variabili indicata invece, si ottiene il modello

$$\min\ 10x_1 + 15x_2 + 20x_3 \tag{1.7}$$

s.t.

$$2x_1 + 2x_2 + x_3 \geq 10000 \tag{1.8}$$

$$2x_1 + 3x_2 + 2x_3 \geq 180000 \tag{1.9}$$

$$x_1 + x_2 + 5x_3 \geq 20000 \tag{1.10}$$

$$x_1, x_2, x_3 \in \mathbb{Z}^+, \tag{1.11}$$

dove i vincoli (1.8), (1.9) e (1.10) rappresentano i quantitativi minimi di manganese, cromo e molibdeno da garantire, rispettivamente.

ESEMPIO 1.2. Un'azienda produce tre modelli 1, 2 e 3 di un certo prodotto. Ciascun modello richiede due tipi di materiali grezzi (A e B) di cui sono disponibili rispettivamente 4000 e 6000 unità. In particolare, per produrre una unità del modello 1 sono necessarie 2 unità di A e 4 unità di B; per una unità del modello 2 sono necessarie 3 unità di A e 2 unità di B; per una unità del modello 3 sono necessarie 5 unità di A e 7 di B. Il modello 1 richiede, per ogni unità prodotta, il doppio di forza lavoro rispetto al modello 2 e il triplo rispetto al modello 3. La forza lavoro presente in azienda è in grado di produrre al massimo l'equivalente di 700 unità/giorno del modello 1. Il settore marketing dell'azienda ha reso noto che la domanda minima per ciascun modello è rispettivamente di 200, 200 e 150 unità. Il profitto unitario di ogni modello è di 30, 20 e 50 euro, rispettivamente.
Formulare il programma lineare per pianificare la produzione giornaliera massimizzando il profitto.

Soluzione Le variabili x_1, x_2, x_3 sono sufficienti a modellare il problema, con $x_i =$ numero di unità di tipo i prodotte. Quindi si ha

$$\max\ 30x_1 + 20x_2 + 50x_3 \tag{1.12}$$

s.t.

$$\begin{aligned} 2x_1 + 3x_2 + 5x_3 &\leq 4000 \\ 4x_1 + 2x_2 + 7x_3 &\leq 6000 \end{aligned} \tag{1.13}$$

$$x_1 + \frac{1}{2}x_2 + \frac{1}{3}x_3 \leq 700 \tag{1.14}$$

$$x_1 \geq 200,\ x_2 \geq 200,\ x_3 \geq 150 \tag{1.15}$$

$$x_1, x_2, x_3 \in \mathbb{Z}^+, \tag{1.16}$$

con i vincoli (1.13), (1.14) e (1.15) che rappresentano rispettivamente i vincoli sulla disponibilità di materie prime, sulla forza lavoro disponibile e sui requisiti minimi di produzione stabiliti dal marketing.

ESEMPIO 1.3. La casa editrice ANALFABETA pubblica un quotidiano che viene distribuito da quattro centri di smistamento S_1, S_2, S_3, S_4 che richiedono rispettivamente almeno 100000, 150000, 50000 e 75000 copie. Il giornale viene stampato in tre tipografie T_1, T_2, T_3 che producono rispettivamente al massimo 125000, 180000 e 70000 copie.
I costi per la spedizione sono di 2 euro/Km. per giornale e le distanze tra le tipografie ed i centri di smistamento sono di 20, 25, 15 e 5 Km. per la prima tipografia, di 12, 14, 18 e 30 Km per la seconda e di 19, 11, 40 e 12 Km per la terza.
Formulare il modello di Programmazione Lineare per pianificare le spedizioni a costo totale minimo.

Soluzione Per come sono specificati i costi di spedizione, la scelta "naturale" per la definizione delle variabili di controllo è la seguente:

$$x_{ij} = \text{numero di copie spedite da } T_i \text{ a } S_j.$$

In questo modo il modello diventa

$$\min \sum_{i=1}^{3} \sum_{j=1}^{4} c_{ij}x_{ij} \tag{1.17}$$

s.t.

$$\begin{aligned} x_{11} + x_{12} + x_{13} + x_{14} &\leq 125000 \\ x_{21} + x_{22} + x_{23} + x_{24} &\leq 180000 \\ x_{31} + x_{32} + x_{33} + x_{34} &\leq 70000 \end{aligned} \tag{1.18}$$

$$\begin{aligned} x_{11} + x_{21} + x_{31} &\geq 100000 \\ x_{12} + x_{22} + x_{32} &\geq 150000 \\ x_{13} + x_{23} + x_{33} &\geq 50000 \\ x_{14} + x_{24} + x_{34} &\geq 75000 \end{aligned} \tag{1.19}$$

$$x_{ij} \in \mathbb{Z}^+, \forall i, j. \tag{1.20}$$

I vincoli (1.18) e (1.19) impongono che ogni tipografia spedisca non più giornali di quanti ne stampa (per la realizzabilità "fisica" della soluzione) e che ogni centro ne riceva una quantità pari almeno al proprio fabbisogno. I costi c_{ij} sono ricavati dalla matrice delle distanze indicata dal testo come $c_{ij} = 2\times$(distanza T_i-S_j).

1.2 Modelli min-max e min-abs

ESEMPIO 1.4. Si riconsideri l'esercizio precedente, cambiando l'obiettivo del modello come segue.
Si definisca il costo di approvvigionamento di un centro di smistamento come il costo totale delle spedizioni verso quel centro; si vuole creare un piano di trasporto che consenta un ragionevole "bilanciamento" di tali costi tra i centri. Si chiede quindi di formulare il modello di Programmazione Lineare che *minimizza il massimo costo di approvvigionamento.*

Soluzione La definizione di variabili rimane, in massima parte, la stessa dell'esercizio precedente.

$$x_{ij} = \text{numero di copie spedite da } T_i \text{ a } S_j.$$

L'obiettivo specificato pone la necessità di scrivere un programma di minimo con una funzione obiettivo del tipo

$$\max_j \Big\{ \sum_{i=1}^{3} c_{ij} x_{ij} \Big\}.$$

Tale espressione è però *non lineare* e quindi proibita nel tipo di modelli qui trattato. Per conservare la linearità del modello, occorre introdurre una variabile ausiliaria y ed una serie di vincoli come segue.

$$\min \; y \tag{1.21}$$

s.t.

$$\begin{aligned} x_{11} + x_{12} + x_{13} + x_{14} &\leq 125000 \\ x_{21} + x_{22} + x_{23} + x_{24} &\leq 180000 \\ x_{31} + x_{32} + x_{33} + x_{34} &\leq 70000 \end{aligned} \tag{1.22}$$

$$\begin{aligned} x_{11} + x_{21} + x_{31} &\geq 100000 \\ x_{12} + x_{22} + x_{32} &\geq 150000 \\ x_{13} + x_{23} + x_{33} &\geq 50000 \\ x_{14} + x_{24} + x_{34} &\geq 75000 \end{aligned} \tag{1.23}$$

$$\sum_{i=1}^{3} c_{ij} x_{ij} \leq y \qquad j = 1, \ldots, 4 \tag{1.24}$$

$$x_{ij} \in \mathbb{Z}^+, \forall\, i, j, \qquad y \geq 0 \tag{1.25}$$

La variabile ausiliaria y ed i vincoli (1.24) permettono di gestire l'obiettivo "min / max" conservando la linearità del modello: in ogni soluzione *ottima* di questo programma lineare, il valore assunto da y coincide esattamente con $\max_j \{\sum_{i=1}^{n} c_{ij} x_{ij}\}$. I vincoli (1.22) e (1.23) hanno il ruolo già noto.

ESEMPIO 1.5. Scrivere il modello in programmazione lineare del seguente problema. Un caporeparto di un'officina di un'azienda meccanica deve pianificare l'esecuzione di cinque lotti su di una macchina della durata rispettivamente di 5 minuti, 7 minuti, 4 minuti, 7 minuti e 10 minuti. La sequenza di esecuzione $(1, 2, 3, 4, 5)$ è data e non ci può essere sovrapposizione temporale fra i lotti. Il primo lotto ha come ora di consegna desiderata le 10.32, il secondo le 10.38, il terzo le 10.42, il quarto le 10.52 ed il quinto le 10.57. Sia l'errore di un lotto pari al valore assoluto della differenza tra il suo tempo di fine lavorazione e l'ora di consegna. Si vuole minimizzare la somma degli errori dei lotti, supponendo che il reparto cominci a lavorare alle 8.30.

Soluzione Il problema richiede di scrivere un modello in grado di determinare gli istanti di inizio lavorazione dei lotti in esame. Assumendo come “zero” del tempo l'ora 8:30, i lotti hanno date di scadenza (espresse in minuti) di 122, 128, 132, 142 e 147. Una serie di variabili è necessaria per rappresentare i tempi di inizio lavorazione:

$$t_i = \text{istante di lavorazione (in minuti dalle 8:30) del lotto } i.$$

Inoltre l'errore del lotto i è dato da $\Delta_i = |t_i + p_i - d_i|$, dove p_i e d_i indicano rispettivamente il tempo di lavorazione e la scadenza del lotto. La funzione obiettivo è quindi del tipo

$$\sum_{i=1}^{5} |t_i + p_i - d_i|.$$

Questo genere di funzione è *non lineare*, quindi occorre nuovamente ricorrere ad un espediente per rappresentare i valori assoluti in un modello lineare. Ricordando che $|x| = \max(x, -x)$, si può pensare di utilizzare la stessa tecnica usata per obiettivi di tipo “min / max”. Si introducono quindi le variabili

$$\Delta_i = \text{errore del lotto } i.$$

Il modello è il seguente.

$$\min \ \Delta_1 + \Delta_2 + \Delta_3 + \Delta_4 + \Delta_5 \tag{1.26}$$

s.t.

$$\begin{aligned}
t_1 + 5 &\le t_2 && (1 \to 2) \\
t_2 + 7 &\le t_3 && (2 \to 3) \\
t_3 + 4 &\le t_4 && (3 \to 4) \\
t_4 + 7 &\le t_5 && (4 \to 5)
\end{aligned} \tag{1.27}$$

$$
\begin{aligned}
\Delta_1 &\geq \quad t_1 + 5 - 122 \\
\Delta_1 &\geq -(t_1 + 5 - 122) \\
\Delta_2 &\geq \quad t_2 + 7 - 128 \\
\Delta_2 &\geq -(t_2 + 7 - 128) \\
\Delta_3 &\geq \quad t_3 + 4 - 132 \\
\Delta_3 &\geq -(t_3 + 4 - 132) \\
\Delta_4 &\geq \quad t_4 + 7 - 142 \\
\Delta_4 &\geq -(t_4 + 7 - 142) \\
\Delta_1 &\geq \quad t_5 + 10 - 147 \\
\Delta_1 &\geq -(t_5 + 10 - 147)
\end{aligned}
\tag{1.28}
$$

$$
t_i \geq 0, \quad \Delta_i \geq 0, \qquad i = 1, \ldots, 5. \tag{1.29}
$$

I vincoli (1.27) garantiscono il rispetto della sequenza di lavorazione che, secondo il testo, è predeterminata, mentre i vincoli (1.28) vincolano le variabili Δ_i ad assumere il valore assoluto di $t_i + p_i - d_i$.

ESEMPIO 1.6. Di un fenomeno fisico che coinvolge due variabili x e y sono state effettuate quattro osservazioni.

Oss.	x	y
1	0.0	1.0
2	2.5	3.0
3	5.0	3.2
4	6.0	4.3

Si suppone che la legge fisica osservata abbia una forma lineare

$$
y = ax + b.
$$

Scrivere il programma lineare per determinare i coefficienti a e b per i quali la retta $y = ax + b$ minimizza lo scarto assoluto medio rispetto alle osservazioni riportate.

Soluzione Si vogliono determinare a e b per i quali è minimo il valore di

$$
z = \sum_{i=1}^{4} |(ax_i + b) - y_i|
$$

dove le $x_1, \ldots, x_4$, $y_1, \ldots, y_4$ sono le osservazioni riportate in tabella. La funzione da minimizzare è una somma di valori assoluti, ma si può ricorrere ad una tecnica analoga a quella utilizzata nell'esempio precedente. Si introducono le variabili $\Delta_1, \ldots, \Delta_4$ con

$$\Delta_i = \text{scarto assoluto dell'osservazione } i\text{-esima.}$$

Il modello risultante è il seguente, nelle variabili a, b, $\Delta_1, \ldots, \Delta_4$.

$$\min \ \Delta_1 + \Delta_2 + \Delta_3 + \Delta_4 \tag{1.30}$$

s.t.

$$\begin{aligned}
\Delta_1 &\geq b - 1 \\
\Delta_1 &\geq 1 - b \\
\Delta_2 &\geq 2.5a + b - 3 \\
\Delta_2 &\geq -2.5a - b + 3 \\
\Delta_3 &\geq 5a + b - 3 \\
\Delta_3 &\geq -5a - b + 3 \\
\Delta_4 &\geq 6a + b - 4.3 \\
\Delta_4 &\geq -6a - b + 4.3
\end{aligned} \tag{1.31}$$

$$\Delta_1, \ldots, \Delta_4 \geq 0, \ \ a, b \text{ libere.} \tag{1.32}$$

I soli vincoli qui sono gli (1.31), che servono a modellare i termini della funzione obiettivo che contengono un valore assoluto.

1.3 Uso di variabili binarie e vincoli logici

ESEMPIO 1.7. Un'azienda agricola produce mais, soia e grano in tre tenute A, B, C. La tenuta A dispone di 600 ettari di terreno e di una riserva di $8 \times 10^6\, m^3$ di acqua. La tenuta B ha 700 ettari di terreno e $5 \times 10^6\, m^3$ di acqua. La terza dispone di 450 ettari e di $6 \times 10^6\, m^3$. Le produzioni di mais, soia e grano garantiscono rispettivamente profitti di 5, 7 e 6 migliaia di euro per ettaro. I consumi di acqua sono di 20000 m^3/ha per il mais, 10000 m^3/ha per la soia e 10000 m^3/ha per il grano. Le direttive della comunità europea richiedono che:

- almeno una tenuta lasci 200 ettari di terreno incolto;
- l'estensione complessiva del terreno coltivato a soia dall'azienda non superi il 40% del totale del suolo coltivato.

Formulare il programma lineare per la massimizzazione del profitto.

Soluzione Lo scenario descrive tre tenute (A, B, C) e tre coltivazioni (M, S, G) (**M**ais, **S**oia, **G**rano). Una scelta naturale di variabili che descrive la coltivazioni da effettuare è

$$x_{AM}, x_{AS}, x_{AG},$$
$$x_{BM}, x_{BS}, x_{BG},$$
$$x_{CM}, x_{CS}, x_{CG},$$

dove x_{ij} = ettari dedicati alla coltivazione j nella tenuta i. Tenuto conto dei profitti espressi in Meuro/ha la funzione obiettivo è

$$f(x) = 5(x_{AM} + x_{BM} + x_{CM}) + 7(x_{AS} + x_{BS} + x_{CS}) + 6(x_{AG} + x_{BG} + x_{CG}).$$

I vincoli descritti dal testo richiedono di:

- rispettare le disponibilità di acqua di ogni tenuta;
- rispettare l'estensione territoriale della tenuta;
- rispettare le norme CEE.

Il rispetto della norma sul terreno incolto viene gestito con tre variabili logiche y_A, y_B, y_C, dove $y_i = 1$ se la tenuta i lascia 200 ha di terreno incolto. Il modello risultante è

$$\max\ 5(x_{AM} + x_{BM} + x_{CM}) + 7(x_{AS} + x_{BS} + x_{CS}) + 6(x_{AG} + x_{BG} + x_{CG})$$

s.t.
(Approvvigionamento idrico — migliaia di litri)

$$20x_{AM} + 10x_{AS} + 10x_{AG} \leq 8000$$

$$20x_{BM} + 10x_{BS} + 10x_{BG} \leq 5000$$

$$20x_{CM} + 10x_{CS} + 10x_{CG} \leq 6000$$

(Terreno disponibile)

$$x_{AM} + x_{AS} + x_{AG} \leq 600$$

$$x_{BM} + x_{BS} + x_{BG} \leq 700$$

$$x_{CM} + x_{CS} + x_{CG} \leq 450$$

(Al più 40% del terreno a soia)

$$x_{AS} + x_{BS} + x_{CS} \leq 0.4(x_{AM} + x_{AS} + x_{AG} + x_{BM} + x_{BS} +$$

$$x_{BG} + x_{CM} + x_{CS} + x_{CG})$$

(Vincolo logico)

$$y_A + y_B + y_C \geq 1$$

$$x_{AM} + x_{AS} + x_{AG} \leq 600 - 200y_A$$

$$x_{BM} + x_{BS} + x_{BG} \leq 700 - 200y_B$$

$$x_{CM} + x_{CS} + x_{CG} \leq 450 - 200y_C$$

$$x_{ij} \geq 0, \forall i, j \qquad y_A, y_B, y_C \in \{0, 1\}.$$

Nota: gli ultimi tre vincoli rendono superflui i precedenti sul terreno disponibile.

ESEMPIO 1.8. Una ditta ha la possibilità di attivare, per l'anno corrente, la produzione di quattro tipi di prodotti A, B, C e D. Per ogni tipo di produzione, se attivata, la ditta si impegna a produrre un quantitativo minimo pari rispettivamente a 1000, 1500, 3000 e 2000 unità. La produzione di A, B, C e D richiede un costo fisso per l'attivazione delle rispettive linee di produzione ed una quantità di forza lavoro per ogni unità prodotta, ed ogni unità venduta fornisce un profitto, come specificato dalla seguente tabella (in euro).

Prodotto	**Costo fisso**	**Forza lavoro unit.**	**Profitto unit.**
A	14500	10	50
B	10000	15	60
C	8000	5	55
D	9000	14	80

La ditta dispone per l'anno in corso di 200000 unità complessive di forza lavoro. Inoltre i committenti per la quale essa lavora richiedono che nel caso venga attivata la produzione di A venga anche prodotto almeno uno tra C o D, almeno nei quantitativi minimi sopra indicati.
Formulare il programma lineare per decidere le produzioni da attivare e pianificarne i quantitativi al fine di massimizzare il saldo costi-profitti.

Soluzione Il problema proposto riguarda la pianificazione di certi tipi di produzione con costi fissi imputabili alla preparazione degli impianti produttivi e costi variabili legati alle quantità prodotte. La decisione di attivare o meno la produzione di A, B, C o D è di tipo vero/falso, e quindi si può modellare con variabii binarie

$$y_i = 1 \qquad \text{iff si attiva la produzione di } i \in \{A, B, C, D\}.$$

Occorre poi determinare anche i volumi prodotti, rappresentabili mediante un'altra serie di variabili

$$x_i = \quad \text{numero di unità di tipo } i \text{ prodotte, } i \in \{A, B, C, D\}.$$

Tenuto conto dei dati su quantitativi minimi, profitti unitari e forza lavoro dati dal testo il modello risulta come segue.

$$\begin{aligned} \max \quad & 50x_A + 60x_B + 55x_C + 80x_D \\ & - (14500y_A + 10000y_B + 8000y_C + 9000y_D) \end{aligned} \tag{1.33}$$

soggetto a

$$\begin{aligned} x_A &\geq 1000y_A \\ x_B &\geq 1500y_B \\ x_C &\geq 3000y_C \\ x_D &\geq 2000y_D \end{aligned} \tag{1.34}$$

$$\begin{aligned} x_A &\leq My_A \\ x_B &\leq My_B \\ x_C &\leq My_C \\ x_D &\leq My_D \end{aligned} \tag{1.35}$$

$$10x_A + 15x_B + 5x_C + 14x_D \leq 2000000 \tag{1.36}$$

$$y_A \leq y_C + y_D \tag{1.37}$$

$$x_i \in \mathbb{Z}_+, y_i \in \{0, 1\}, \qquad i \in \{A, B, C, D\}. \tag{1.38}$$

I vincoli (1.34) impongono il rispetto dei quantitativi minimi qualora un tipo di produzione venga attivato, mentre i vincoli (1.35) assicurano che non vengano pianificati quantitativi per produzioni non attivate. Il vincolo (1.36) impone di non eccedere la quantità di forza lavoro disponibile; il vincolo (1.37) modella l'implicazione logica

$$y_A \implies y_C \vee y_D,$$

come richiesto dal testo (considerare la tabella di verità dell'implicazione in esame).

ESEMPIO 1.9. Una ditta che si occupa di riparazioni deve pianificare le assunzioni per i prossimi 5 mesi. All'inizio la ditta dispone di 20 operai esperti; ogni operaio esperto fornisce 150 ore di lavoro al mese e percepisce uno stipendio mensile di 1000 euro. Un operaio neoassunto, durante il primo mese di servizio percepisce uno stipendio di 500 euro e non fornisce in pratica lavoro utile; per questo primo mese gli viene invece affiancato un lavoratore esperto per insegnargli il mestiere. Ogni lavoratore esperto che svolge affiancamento rende per 70 ore di lavoro al mese (anziché 150). Dopo il mese di apprendistato i lavoratori neoassunti diventano esperti, con pari abilità lavorativa e stipendio. Le quantità di ore/lavoro da coprire per i prossimi 5 mesi sono rispettivamente di 2000, 4000, 7000, 3000, 3500 ore. Infine, se si assumono almeno 10 persone nel corso dei primi due mesi, l'azienda puó incassare un contributo statale di 100000 euro. Formulare il programma lineare che consente di pianificare le assunzioni riducendo al minimo i costi del personale nei prossimi cinque mesi.

Soluzione Il problema richiede in pratica di gestire un pool di assunti/neoassunti che varia di mese in mese e si chiede di pianificare la gestione del personale per un numero finito di mesi. I modelli nei quali viene pianificata un'attività lungo un numero fissato di intervalli temporali sono anche detti *problemi multiperiodali*. Una soluzione è un "piano di assunzioni" che permetta di coprire comunque il monte-ore richiesto nei vari mesi compatibilmente con lo svolgimento dell'affiancamento da parte degli esperti. Possiamo modellare la situazione con dieci variabili:

$$y_1, y_2, y_3, y_4, y_5,$$

$$x_1, x_2, x_3, x_4, x_5,$$

dove y_i = disponibilità di esperti al mese i e x_i = numero di persone assunte al mese i. Modelliamo con una variabile logica z la scelta di ottenere o non ottenere il contributo statale. Come ulteriore considerazione, si noti che la x_5 è superflua in quanto assumere all'ultimo mese è pienamente un costo: non fornisce forza lavoro sfruttabile entro l'orizzonte temporale coperto dal modello e non influisce sulla possibilità di ottenere il contributo statale, per cui in ogni soluzione ottima si avrà $x_5 = 0$.
In base alle variabili definite, il costo del personale (assunzioni e stipendi) totale nei cinque mesi è:

$$f(x) = 500(x_1 + x_2 + x_3 + x_4 + x_5) + 1000(y_1 + y_2 + y_3 + y_4 + y_5) - 100000z.$$

Il modello complessivo risulta essere:

$$\min \; 500(x_1 + x_2 + x_3 + x_4 + x_5) + 1000(y_1 + y_2 + y_3 + y_4 + y_5) - 100000z$$

s.t.
(Mese 1)

$$y_1 = 20$$

$$x_1 \leq y_1$$

$$150(y_1 - x_1) + 70x_1 \geq 2000$$

(Mese 2)

$$y_2 = y_1 + x_1$$

$$x_2 \leq y_2$$

$$150(y_2 - x_2) + 70x_2 \geq 4000$$

(Mese 3)

$$y_3 = y_2 + x_2$$

$$x_3 \leq y_3$$

$$150(y_3 - x_3) + 70x_3 \geq 7000$$

(Mese 4)

$$y_4 = y_3 + x_3$$

$$x_4 \leq y_4$$

$$150(y_4 - x_4) + 70x_4 \geq 3000$$

(Mese 5)

$$y_5 = y_4 + x_4$$

$$x_5 \leq y_5$$

$$150(y_5 - x_5) + 70x_5 \geq 3500$$

(Vincolo logico)

$$x_1 + x_2 \geq 10z$$

$$x_i, y_i \in \mathbb{Z}^+ \, \forall i, \quad z \in \{0, 1\}$$

ESEMPIO 1.10. Una raffineria produce benzina verde e super a partire da due tipi di greggio A e B, usando tre impianti. Il primo impianto può produrre 2 barili di verde e 3 di super a partire da 4 barili di greggio tipo A e 3 barili di greggio tipo B. Il secondo impianto può produrre 4 barili di verde e 2 di super a partire da 3 barili di A e 4 barili di B. Il terzo può produrre 2 barili di verde e 2 barili di super a partire da 3 barili di A e 3 barili di B.
Gli impianti lavorano *sempre* con le proporzioni specificate. La benzina verde viene venduta a 40 euro/barile, la super a 50 euro/barile. Sono disponibili per questo

mese 5000 barili di greggio A e 6000 barili di greggio B. Per esigenze legate ad altre lavorazioni, almeno uno tra gli impianti deve produrre non più di 1000 barili.
Formulare il programma lineare per massimizzare il profitto legato alla produzione mensile.

Soluzione 1 Il problema richiede di formulare un piano di produzione di benzina verde/super con l'obiettivo di massimizzare il profitto senza sforare dalle scorte di magazzino e garantendo (particolare importante) l'alimentazione degli impianti con il corretto mix di greggio ed il limite di 1000 barili per almeno un impianto (vincolo logico). Una possibile scelta di variabili è:

$$x_{v1}, x_{v2}, x_{v3},$$

$$x_{s1}, x_{s2}, x_{s3},$$

$$y_{A1}, y_{A2}, y_{A3},$$

$$y_{B1}, y_{B2}, y_{B3},$$

dove x_{ij} = barili di verde e super prodotti dall'impianto j e y_{ij} = numero di barili di greggio di tipo i che vanno ad alimentare l'impianto j. Le variabili x_{ij} permettono di scrivere in modo immediato la funzione obiettivo, le altre servono per gestire l'alimentazione degli impianti. Tre variabili logiche z_1, z_2, z_3 decideranno quali impianti dovranno produrre meno di 1000 barili. Il profitto totale è:

$$f(x) = 40(x_{v1} + x_{v2} + x_{v3}) + 50(x_{s1} + x_{s2} + x_{s3}),$$

quindi il modello complessivo risulta:

$$\max\ 40(x_{v1} + x_{v2} + x_{v3}) + 50(x_{s1} + x_{s2} + x_{s3})$$

s.t.
(Scorte di magazzino)

$$y_{A1} + y_{A2} + y_{A3} \leq 5000$$

$$y_{B1} + y_{B2} + y_{B3} \leq 6000$$

(Alimentazione impianti)

$$\begin{aligned}
3y_{A1} &= 4y_{B1} && \text{ingresso 1: } y_{A1} : y_{B1} = 4:3 \\
3x_{v1} &= 2x_{s1} && \text{uscita 1: } x_{v1} : x_{s1} = 2:3 \\
2y_{A1} &= 4x_{v1} && \text{ingresso/uscita 1: } y_{A1} : x_{v1} = 4:2 \\
4y_{A2} &= 3y_{B2} && \text{ingresso 2} \\
2x_{v2} &= 4x_{s2} && \text{uscita 2} \\
4y_{A2} &= 3x_{v2} && \text{ingresso/uscita 2} \\
y_{A3} &= y_{B3} && \text{ingresso 3} \\
x_{v3} &= x_{s3} && \text{uscita 3} \\
2y_{A3} &= 3x_{v3} && \text{ingresso/uscita 3}
\end{aligned}$$

(Vincolo logico)

$$\begin{aligned}
z_1 + z_2 + z_3 &\geq 1 \\
x_{v1} + x_{s1} &\leq 1000 + M(1 - z_1) \\
x_{v2} + x_{s2} &\leq 1000 + M(1 - z_2) \\
x_{v3} + x_{s3} &\leq 1000 + M(1 - z_3) \\
x_{ij}, y_{ij} \geq 0\, \forall i, j &\quad z_1, z_2, z_3 \in \{0, 1\}
\end{aligned}$$

Soluzione 2 Per questo problema una scelta di variabili leggermente meno ovvia permette di ottenere un modello drasticamente meno complesso. La definizione delle variabili si basa proprio sull'osservazione che gli impianti lavorano sempre con rapporti nei mix di ingresso e di uscita ben determinati. Siano

$$x_1, x_2, x_3$$

il numero (anche frazionario) di "lavorazioni" svolte agli impianti 1, 2 e 3 rispettivamente. Queste tre variabili, unite alle tre variabili logiche z_1, z_2 e z_3 sono sufficienti a modellare il problema. Il modello è il seguente:

$$\max\ 40(2x_1 + 4x_2 + 2x_3) + 50(3x_1 + 2x_2 + 2x_3)$$

s.t.
(Scorte di magazzino)

$$4x_1 + 3x_2 + 3x_3 \leq 5000$$

$$3x_1 + 4x_2 + 3x_3 \leq 6000$$

(Vincolo logico)

$$z_1 + z_2 + z_3 \geq 1$$

$$5x_1 \leq 1000 + M(1 - z_1)$$

$$6x_2 \leq 1000 + M(1 - z_2)$$

$$4x_3 \leq 1000 + M(1 - z_3)$$

$$x_1, x_2, x_3 \geq 0, \quad z_1, z_2, z_3 \in \{0, 1\}$$

Nota: in questo modello il rispetto delle proporzioni nell'alimentazione e nell'uscita degli impianti è garantito automaticamente dalla definizione delle variabili.

ESEMPIO 1.11. Uno stabilimento chimico produce per conto terzi tre tipi di polveri A, B e C. Tutta la produzione è acquistata in blocco dal committente a fine mese. Lo stabilimento ha tre impianti 1, 2 e 3 con caratteristiche diverse. L'impianto 1 produce polveri B e C nella proporzione di 2:3 fino a un totale di 4 tonnellate di polvere al mese; l'impianto 2 produce fino a 2 tonn./mese di polveri A e C nella proporzione 1:2; l'impianto 3 produce polveri A e B fino ad un totale di 5 tonn./mese, in proporzione 1:3. Per lo stoccaggio della produzione mensile sono disponibili due magazzini M1 ed M2, della capacità di 4 e 5 tonnellate rispettivamente. Per ragioni di sicurezza le polveri A e C non possono essere stoccate nello stesso magazzino. Formulare il programma lineare per massimizzare la produzione complessiva di polveri.

Soluzione Analogamente al modello precedente, la definizione delle variabili che produce un modello semplice è:

$$x_1, x_2, x_3,$$

dove x_i = totale di polvere (di qualunque tipo) prodotta dall'impianto i (altre scelte, equivalenti, sono possibili); inoltre per gestire lo stoccaggio nei magazzini (dove le polveri vanno trattate in modo distinto) introduciamo:

$$x_{A1}, x_{A2}, x_{B1}, x_{B2}, x_{C1}, x_{C2}$$

dove x_{ij} = quantità di polvere di tipo i stoccata al magazzino j. Due variabile logiche y_1, y_2 discriminano il magazzino nel quale viene stoccata la polvere di tipo A ($y_i = 1$ se e solo se A va al magazzino i).

Il problema chiede di produrre un piano di produzione e stoccaggio compatibile con le capacità dei due magazzini e con il vincolo di sicurezza.
La funzione obiettivo è data da:

$$f(x) = x_1 + x_2 + x_3,$$

ed il modello risulta:

$$\max \; x_1 + x_2 + x_3$$

s.t.
(Tutto ciò che viene prodotto va stoccato)

$$\frac{1}{3}x_2 + \frac{1}{4}x_3 = x_{A1} + x_{A2} \qquad \text{(polvere } A\text{)}$$

$$\frac{2}{5}x_1 + \frac{3}{4}x_3 = x_{B1} + x_{B2} \qquad \text{(polvere } B\text{)}$$

$$\frac{3}{5}x_1 + \frac{2}{3}x_2 = x_{C1} + x_{C2} \qquad \text{(polvere } C\text{)}$$

(Capacità dei magazzini)

$$x_{A1} + x_{B1} + x_{C1} \leq 4$$

$$x_{A2} + x_{B2} + x_{C2} \leq 5$$

(Capacità impianti)

$$x_1 \leq 4, \; x_2 \leq 2, \; x_3 \leq 5$$

(Vincolo logico)

$$x_{A1} \leq My_1 \qquad x_{A2} \leq My_2$$

$$x_{C1} \leq M(1 - y_1) \quad x_{C2} \leq M(1 - y_2)$$

$$x_i, x_{ij} \geq 0, \forall i, j \quad y \in \{0, 1\}$$

ESEMPIO 1.12. L'azienda PC4All produce pc e deve acquistare le scorte di materie prime necessarie per la produzione dei case. Per produrre i case nel mese corrente sono necessari i seguenti materiali:

- viti: 15000 unità;
- plastica: 1300 kg.;

- acciaio: 2900 kg.

Per effettuare gli acquisti l'azienda si può appoggiare a quattro fornitori, i quali le forniscono le materie prime in lotti contenenti le seguenti quantità di materiale:

	viti	plastica	acciaio
F1	50	3	5
F2	30	4	7
F3	25	1	3
F4	10	8	1

Nell'ottica di gestire al meglio il proprio magazzino, la PC4All intende avere, alla fine del mese, la minor quantità di materiale non utilizzato possibile e, a tal fine, è disposta anche a comprare una quantità di materie prime inferiore alle proprie necessità. Il costo per lo stoccaggio o per il mancato acquisto di una unità di materiale è il seguente:

Viti	0.2 euro/pezzo
Plastica	1 euro/kg.
Acciaio	3 euro/kg.

Per motivi commerciali l'azienda, se acquista dei lotti di materiale dal fornitore F1, è impossibilitata a rifornirsi dai fornitori F2 ed F4.
Formulare il modello di programmazione lineare che minimizzi i costi derivanti dallo scostamento tra le quantità di materiali acquistate e quelle necessarie, tenendo conto che non è possibile comprare porzioni di lotto di materiali.

Soluzione L'azienda in questione, a detta del testo, paga un costo sia per lo stoccaggio che per il mancato acquisto di materiali. Quindi, tenuto conto dei quantitativi richiesti e del fatto che si può anche comprare meno del fabbisogno, una funzione obiettivo possibile è:

$$0.2|V - 15000| + |P - 1300| + 3|A - 2900|,$$

dove V, P, A sono rispettivamente i quantitativi totali di viti, plastica e acciaio acquistati. I valori assoluti introducono caratteristiche di non linearità nel modello che devono essere opportunamente gestite.
Siccome i fornitori hanno a disposizione lotti dai contenuti standard, una scelta naturale di variabili per descrivere il "piano di approvvigionamento" è:

$$x_1, x_2, x_3, x_4,$$

dove x_i = numero di lotti acquistati dal fornitore F_i. Per gestire i valori assoluti sono necessarie tre variabili ausiliarie y_V, y_P, y_A. Si noti che $|X| = \max(X, -X)$, quindi i valori assoluti si gestiscono con tecniche simili a quelle usate per i problemi

min / max. La scelta tra i fornitori $F_2 + F_4$ e F_1 è modellata con una variabile logica z_1; $z_1 = 1$ se e solo se l'azienda si rifornisce presso F_1.
Il modello è il seguente:

$$\min\ 0.2y_V + y_P + 3y_A$$

s.t.
(Vincoli per gestire i valori assoluti)

$$50x_1 + 30x_2 + 25x_3 + 10x_4 - 15000 \leq y_V$$

$$-50x_1 - 30x_2 - 25x_3 - 10x_4 + 15000 \leq y_V$$

$$3x_1 + 4x_2 + x_3 + 8x_4 - 1300 \leq y_P$$

$$-3x_1 - 4x_2 - x_3 - 8x_4 + 1300 \leq y_p$$

$$5x_1 + 7x_2 + 3x_3 + x_4 - 2900 \leq y_A$$

$$-5x_1 - 7x_2 - 3x_3 - x_4 + 2900 \leq y_A$$

(Vincolo logico — grande M)

$$x_2 + x_4 \leq M(1 - z_1)$$

$$x_1 \leq Mz_1$$

$$x_1, x_2, x_3, x_4 \in \mathbb{Z}^+, \quad y_V, y_P, y_A \in \mathbb{Z}^+, \quad z_1 \in \{0, 1\}$$

ESEMPIO 1.13. Si consideri la situazione dell'esempio 3, modificato come segue. I quotidiani sono distribuiti da quattro centri di smistamento S_1, S_2, S_3, S_4 che richiedono rispettivamente almeno $b_1 = 100000$, $b_2 = 150000$, $b_3 = 50000$ e $b_4 = 75000$ copie. Il giornale viene stampato in tre tipografie T_1, T_2, T_3 che producono rispettivamente al massimo $a_1 = 125000$, $a_2 = 180000$ e $a_3 = 70000$ copie.
Le distanze d_{ij} tra le tipografie ed i centri di smistamento sono di 20, 25, 15 e 5 Km. per la prima tipografia, di 12, 14, 18 e 30 Km per la seconda e di 19, 11, 40 e 12 Km per la terza.
I trasportatori che occupano di recapitare i quotidiani dalle tipografie ai centri di smistamento chiedono su ogni tratta, come costo di trasporto 3 euro/Km. per giornale per quantitativi non superiori alle 30000 copie. Per quantitativi superiori alle 30000 copie la tariffa sulla stessa tratta scende a 1 euro/giornale.
Formulare il modello di Programmazione Lineare per pianificare le spedizioni a costo totale minimo.

Soluzione La difficoltà in questa variante del problema sta nel fatto che il costo di trasporto è una funzione lineare a tratti del numero di giornali trasportato:

$$< \text{costo-totale} >_{ij}= \begin{cases} 3d_{ij}x_{ij} & \text{se } x_{ij} \leq 30000, \\ d_{ij}x_{ij}, & \text{se } x_{ij} \geq 30001. \end{cases}$$

Un possibile modo per ovviare a questo inconveniente è duplicare le variabili di trasporto. Si definiscono, per $i = 1, \ldots, 3$, $j = 1, \ldots, 4$:

- x_{ij} = numero di copie spedite da T_i a S_j con tariffa da 3 euro/Km ;
- y_{ij} = numero di copie spedite da T_i a S_j con tariffa da 1 euro/Km.

Si introducono inoltre le variabili binarie t_{ij} che assumono valore 1 se e solo se si vuole utilizzare la tariffa da 3 euro sulla tratta (i, j).
Il modello si formula come segue.

$$\min \sum_{i=1}^{3} \sum_{j=1}^{4} 3d_{ij}x_{ij} + \sum_{i=1}^{3} \sum_{j=1}^{4} d_{ij}y_{ij} \tag{1.39}$$

$$\sum_{j=1}^{4} (x_{ij} + y_{ij}) = a_i \qquad i = 1, \ldots, 3 \tag{1.40}$$

$$\sum_{i=1}^{3} (x_{ij} + y_{ij}) = b_j \qquad j = 1, \ldots, 4 \tag{1.41}$$

$$x_{ij} \leq 30000 t_{ij} \qquad i = 1, \ldots, 3,\ j = 1, \ldots, 4 \tag{1.42}$$

$$y_{ij} \leq M(1 - t_{ij}) \qquad i = 1, \ldots, 3,\ j = 1, \ldots, 4 \tag{1.43}$$

$$y_{ij} \geq 30001(1 - t_{ij}) \qquad i = 1, \ldots, 3,\ j = 1, \ldots, 4 \tag{1.44}$$

$$x_{ij}, y_{ij} \in \mathbb{Z}_+, t_{ij} \in \{0, 1\} \quad i = 1, \ldots, 3,\ j = 1, \ldots, 4 \tag{1.45}$$

La costante M può essere posta ad esempio al valore $a_1 + a_2 + a_3$. I vincoli (1.40) e (1.41) sono i normali vincoli di trasporto, dove però le quantità spedite su ogni tratta (i, j) sono le somme $(x_{ij} + y_{ij})$. La funzione obiettivo (1.39) pesa con costi differenti i quantitativi x_{ij} e y_{ij}. I vincoli (1.42)–(1.44) assicurano che vengano rispettate le condizioni per usufruire delle tariffe da 3 e 1 euro rispettivamente:

- se $t_{ij} = 1$, i vincoli (1.42) e (1.43) assicurano che sulla tratta (i, j) si abbia $y_{ij} = 0$ e $x_{ij} \leq 30000$;
- se $t_{ij} = 0$, i vincoli (1.42) e (1.44) assicurano che si abbia $x_{ij} = 0$ e $y_{ij} \geq 30000 + 1$.

1.4 Esercizi proposti

1. Scrivere il modello in programmazione lineare del seguente problema. Una società di investimenti finanziari deve gestire un budget di 1000000 euro per conto di un cliente. Le alternative prese in considerazione sono fondi di investimento nazionali (FIN), fondi di investimento internazionali (FII), buoni ordinari del tesoro (BOT), buoni ordinari comunali (BOC) ed azioni immobiliari (AI). I rendimenti annui e le valutazioni di rischio sono presentati nella tabella di seguito.

titoli	FIN	FII	BOT	BOC	AI
rendimenti	9.5	10.5	12	12.5	6
rischi	5	6	7	8	1

   Il cliente non vuole superare come rischio medio 6.5 e vuole garantirsi un rendimento annuo del 11.5%. L'investimento deve durare 5 anni. I BOT ed i BOC sono tassati i primi due anni del 12.5%, mentre i fondi di investimento sono tassati all'ultimo anno del 25% (le azioni immobiliari sono esenti da tasse). Scrivere il PL per la formulazione di un piano di investimenti che massimizzi il rendimento globale alla fine dei cinque anni (la decisione presa all'inizio non può essere modificata durante i cinque anni).

2. Una società finanziaria deve esaminare una serie di domande di finanziamento da parte di 10 aziende di cui 3 sono aziende tessili, 5 sono aziende agricole alimentari e 2 sono aziende meccaniche. Per quanto riguarda le aziende tessili si sa che se viene finanziata la prima non devono essere finanziate le altre due. Per ciò che concerne le aziende meccaniche, invece, o vengono finanziate entrambe oppure nessuna delle due. Infine, quanto alle aziende agricole, si vuole che almeno un'azienda venga finanziata. Le richieste di finanziamento sono di 30, 20 e 15 milioni di euro rispettivamente per la prima, la seconda e la terza delle aziende tessili, di 20 milioni di euro per ciascuna delle aziende agricole e di 40 e 30 milioni di euro rispettivamente per le aziende meccaniche. Il capitale disponibile è di 150 milioni. Trattandosi di progetti di sviluppo, ad ogni possibile finanziamento è associato un valore di ritorno che è di 15 milioni di euro, 10 milioni di euro e 10 milioni di euro rispettivamente per la prima, la seconda e la terza delle aziende tessili, mentre è di 20 e 8 milioni di euro rispettivamente per le due aziende meccaniche ed è di 12 milioni di euro per ciascuna delle aziende agricole. Si scriva il problema di PL che massimizzi il valore di ritorno totale.

3. Un'azienda produce giocattoli. Una ricerca di mercato prevede per l'anno prossimo una domanda di almeno 20.000 unità del suo giocattolo. L'azienda, per motivi strategici, ritiene indispensabile soddisfare questa domanda. Il giocattolo è costituito da 3 componenti: G1, G2, G3. Ogni componente è prodotta attraverso tre stadi di lavorazione: M1, M2, M3. I tempi di lavorazione necessari

(espressi in ore) e le disponibilità delle macchine (espresse in ore) sono riportati nella tabella seguente.

	M1	M2	M3
G1	0.05	0.06	0.03
G2	0.09	0.03	0.06
G3	0.07	0.08	0.05
Disponibilità	1000	1000	1000

L'azienda ha anche preso in esame la possibilità di acquistare fuori le tre componenti necessarie. Per motivi logistici è stato però deciso che la componente G1 debba essere o completamento prodotta in stabilimento oppure completamente acquistata all'esterno. Inoltre non è possibile acquistare anche parzialmente tutte le componenti all'esterno contemporaneamente. I costi unitari nelle due possibilità (acquisto o produzione) sono riportati nella tabella seguente.

	G1	G2	G3
Acquisto	2.20	1.40	1.90
Produzione	1.00	2.00	1.60

Si richiede di minimizzare i costi dell'azienda garantendo il soddisfacimento della domanda.

4. Secondo i dati ISTAT, la provincia di Torino può essere suddivisa in 7 centri di domanda 1,2,3,4,5,6 e 7. Un'azienda ha individuato 5 punti A,B,C,D ed E, nei quali potrebbero essere costruiti dei nuovi ipermercati per soddisfare la domanda dei centri di domanda. Tale impresa è interessata a soddisfare la domanda sopramenzionata in modo tale che i clienti non percorrano più di 30 minuti di auto per raggiungere almeno uno dei centri vendita. Nella tabella seguente viene indicato il tempo auto necessario per raggiungere un punto di offerta da un punto di domanda.

	A	B	C	D	E
1	41	33	24	29	58
2	25	12	22	58	41
3	21	43	34	54	18
4	21	42	39	26	18
5	11	23	24	29	53
6	47	23	19	16	31
7	37	47	51	26	19

L'impresa ha inoltre fatto sapere che accetterà soluzioni che prevedano l'attivazione del centro vendita B se è già attivo uno dei centri C o D.
L'apertura dei centri vendita costa rispettivamente (in milioni di euro):

A = 310, B = 250, C = 260, D = 330, E = 280.
L'obiettivo dell'impresa è di minimizzare i costi di apertura dei centri vendita garantendo il fatto che che tutti i punti di domanda vengano serviti.

5. Un fiorista deve addobbare una sala per un ricevimento. Ha a disposizione quattro tipi di fiori: rose, gerbere, lilium e calle. Rose, gerbere e lilium sono disponibili in tre colori diversi, rosa, bianco e giallo, mentre le calle sono soltanto bianche. Inoltre ha a disposizione solo 350 rose, indipendentemente dal colore. Le rose costano 4 Euro l'una, le gerbere 2.5 Euro, i lilium 2 Euro l'uno e le calle 3 Euro l'una. Per addobbare interamente la sala sono necessari almeno 1000 fiori. La sala può contenere al massimo 1600 fiori. Gli organizzatori del ricevimento desiderano che i tre colori siano presenti in percentuale uguale. Scrivere un modello in programmazione lineare che minimizzi il costo complessivo dell'addobbo.

6. Il capocuoco di un ristorante ha a disposizione 750 Euro, per approvvigionare la cucina di pesce. Sul mercato sono disponibili 5 tipi diversi di pesci: branzini, orate, totani, aragoste e gamberi. Branzini e orate vengono cucinati alla griglia, aragoste e gamberi vengono cucinati a vapore mentre i totani vengono serviti ripieni. I branzini costano 5 Euro, l'uno, le orate 6 Euro, i totani 4 Euro l'uno, i gamberi 7 Euro, l'uno mentre le aragoste costano 10 Euro, l'una. Una volta cucinati, i branzini e le orate vengono venduti dal ristorante a 9 Euro l'uno, i totani a 10 Euro, i gamberi a 13 Euro, e le aragoste a 20 Euro. Il ristorante vuole poter offrire ogni giorno almeno 50 piatti a base di pesce alla griglia e almeno 20 piatti a base di pesce a vapore. Scrivere il modello in programmazione lineare del problema, in modo da massimizzare il profitto ottenuto, supponendo di vendere tutti i piatti di pesce preparati.

7. Una coppia di sposi deve ordinare le bomboniere e i confetti per il matrimonio. Il negozio a cui si rivolgono offre tre diversi formati di bomboniere, grandi, medie e piccole: i primi due formati sono disponibili sia in ceramica sia in stoffa, mentre le bomboniere piccole sono solo in stoffa. Una bomboniera grande in stoffa costa 4 Euro e contiene 8 confetti, una grande in ceramica costa 5 Euro e contiene 6 confetti, una bomboniera media in stoffa costa 2.5 Euro e contiene 6 confetti, una media in ceramica costa 4 Euro e contiene 5 confetti, mentre una bomboniera piccola costa 1.5 Euro e contiene 3 confetti. I confetti possono essere alla mandorla o al cioccolato e, in entrambi i casi, sono venduti in confezioni da 100 confetti l'una. Una confezione di confetti al cioccolato costa 10 Euro, mentre una confezione di confetti alla mandorla costa 8 Euro. Ciascuna bomboniera deve essere riempita di un solo tipo di confetto: gli sposi desiderano che al massimo $\frac{1}{3}$ delle bomboniere sia al cioccolato. Sono necessarie almeno 150 bomboniere e di queste almeno $\frac{1}{4}$ devono essere medie o grandi.

   Scrivere il modello in programmazione lineare del problema, in modo da minimizzare il costo complessivo delle bomboniere e dei confetti.

8. Un'azienda produttrice di PC deve pianificare le forniture per il mese di novembre. I componenti necessari sono dei moduli RAM, dei processori e degli Hard Disk. I moduli RAM sono di due tipi: moduli da 1 GB e moduli da 2 GB. I processori sono di due tipi: a 3000 Mhz e a 2500 Mhz. Gli HD sono di 3 tipi: da 40, 60 o 100 GB. Il mercato ha una richiesta di

   - 4000 PC con 1 processore a 3000 Mhz, HD 60 GB e 4 moduli da 1 GB di RAM;
   - 1000 PC con 1 processore a 2500 Mhz, HD da 40 GB e due moduli da 1 GB di RAM;
   - 2000 PC con 2 processori a 3000 Mhz, 1 HD da 100 GB, 1 HD da 40 GB e 2 moduli da 2 GB di RAM;
   - 3000 PC con 1 processore a 2500 Mhz, 2 HD da 40 GB e 1 modulo da 2 GB di RAM.

   I fornitori vendono dei kit così composti:

   - un processore a 3000 Mhz, 1 HD da 100 GB e 4 moduli da 2 GB al costo di 1300 Euro;
   - 10 processori a 2500 Mhz, 10 HD da 60 GB, 4 HD da 40 GB e 24 moduli da 1 GB, al costo di 4400 Euro;
   - 8 processori a 3000 Mhz e 12 moduli da 1 GB, al costo di 3100 euro;
   - 10 moduli da 2 GB, 5 HD da 40 GB e 4 HD da 60 GB, al costo di 600 Euro.

   Determinare gli ordini dell'azienda che minimizzano il costo complessivo.

9. Una società deve pianificare la sua produzione per gli ultimi cinque mesi dell'anno. Non ci sono giacenze di magazzino all'inizio di agosto e non devono essercene alla fine dell'anno. Sono noti per ogni mese i dati riportati nella seguente tabella. Si richiede di formulare un piano di produzione a costo minimo totale.

Mese	Vendite	Capacità produttiva	Costo unitario di produzione	Costo unitario di magazzino
Agosto	25	40	36	3
Settembre	25	50	38	4
Ottobre	55	30	33	3
Novembre	35	50	40	4
Dicembre	30	20	30	5

10. Lo stato di Islandia ha quattro industrie esportatrici: acciaio, motori, elettronica e plastica. Il ministro dell'economia di questo stato vuole massimizzare il

saldo esportazioni-importazioni. La moneta di Islandia è il klunz. I prezzi in klunz sul mercato mondiale per unità di acciaio, motori, elettronica e plastica sono rispettivamente 500, 1500, 300 e 1200. La produzione di una unità di acciaio richiede 0.02 unità di motori, 0.01 unità di plastica, 250 klunz di materie prime acquistate sul mercato mondiale e mezzo anno-uomo di manodopera. La produzione di una unità di motori richiede 0.8 unità di acciaio, 0.15 unità di elettronica, 0.11 unità di plastica, 300 klunz di materie prime acquistate sul mercato mondiale e un anno-uomo di manodopera. La produzione di una unità di prodotti elettronici richiede 0.01 unità di acciaio, 0.01 unità di motori, 0.05 unità di plastica, 50 klunz di materie prime acquistate sul mercato mondiale e mezzo anno-uomo di manodopera. La produzione di una unità di plastica richiede 0.03 unità di motori, 0.2 unità di acciaio, 0.05 unità di elettronica, 300 klunz di materie prime acquistate sul mercato mondiale e due anni-uomo di manodopera. La produzione di motori è limitata a 650000 unità, quella di plastica a 60000 unità. La manodopera totale disponibile in Islandia è di 830000 uomini per anno. Acciaio, motori, elettronica e plastica non possono essere importati, ma devono essere prodotti all'interno.

11. Una ditta di abbigliamento deve pianificare la produzione di maglioni per la collezione Autunno-Inverno. La ditta produce quattro modelli diversi, utilizzando lana nera, crema, nocciola e rossa. La quantità di lana di diverso colore richiesta da ciascun maglione è riportata nella seguente tabella:

	lana nera	lana nocciola	lana crema	lana rossa
modello I	-	1 kg	1 kg	1 hg
modello II	2 kg	-	-	1 hg
modello III	-	-	2 kg	1 hg
modello IV	1.5 kg	-	5 hg	5 hg

I maglioni possono essere prodotti in tre diversi stabilimenti: il primo stabilimento ha manodopera sufficiente a produrre 500 maglioni del primo modello, il secondo stabilimento ha manodopera sufficiente a produrre 900 maglioni del primo modello e il terzo stabilimento può produrre 1500 maglioni del primo modello. Un maglione del secondo modello richiede metà del tempo richiesto dal primo modello, mentre il terzo e quarto modello ne richiedono il doppio. La direzione della ditta desidera che la quantità di maglioni prodotti nel terzo stabilimento non sia inferiore alla somma delle quantità prodotte negli altri due.

Sono richiesti dal mercato almeno 400 maglioni con lana nocciola. Per ragioni di mercato, si vuole che i maglioni neri siano almeno il 50% dei maglioni prodotti e che i maglioni color crema siano meno del 20%. Formulare un modello in programmazione lineare per minimizzare i costi, sapendo che la lana nera costa 7 euro al chilogrammo, la lana nocciola costa 5.5 euro al chilogrammo, la lana crema costa 6 euro al chilogrammo e la lana rossa 10 euro al chilogrammo.

2

Metodo del Simplesso

2.1 Simplesso primale

2.1.1 Richiami teorici

Sia dato un problema di P.L. in forma standard:

$$\min \overline{c}^T \overline{x}$$

$$A\bar{x} = \overline{b}$$

$$\bar{x} \geq \overline{0}$$

E' sempre possibile convertire un generico problema di P.L. in forma standard utilizzando le seguenti regole:

- Una F.O. di massimo $\max \overline{c}^T \overline{x}$ va cambiata di segno: $\min -\overline{c}^T \overline{x}$.
- Se l' i-esimo vincolo è nella forma $\overline{ax} \leq b_i$ si introduce una *variabile di slack* $y_i \geq 0$ e si riscrive il vincolo come $\overline{ax} + y_i = b_i$.
- Se l' i-esimo vincolo è nella forma $\overline{ax} \geq b_i$ si introduce una *variabile di surplus* $y_i \geq 0$ e si riscrive il vincolo come $\overline{ax} - y_i = b_i$.
- Se x_i è una variabile libera va sostituita con la differenza tra due variabili non negative, ovvero $x_i = u_i - v_i$ dove $u_i \geq 0, v_i \geq 0$.
- Se x_i è una variabile negativa va sostituita con una variabile non negativa cambiata di segno, ovvero $x_i = -z_i$ dove $z_i \geq 0$.

Si supponga di disporre di una soluzione di base ammissibile (SAB) del problema. Si formi il tableau del simplesso relativo a tale soluzione. L'algoritmo del simplesso è riassumibile nel seguente pseudocodice:

Function Simplesso
ottimo = *illimitato* = **false**;
while ((*ottimo* = **false**) **and** (*illimitato* = **false**)) **do**
  if $r_j \geq 0, \forall j$ **then**
    ottimo = **true**
  else
    Scegliere una colonna q (corrispondente alla variabile x_q) avente $r_q < 0$
    if $y_{iq} \leq 0, \forall i$ **then**
      illimitato = **true**
    else
      $\epsilon = \min_{i:y_{iq}>0} \frac{y_{i0}}{y_{y_{iq}}} = \frac{y_{p0}}{y_{pq}}$
      La $p-ima$ variabile della base esce dalla base e viene sostituita dalla variabile x_q
      Eseguire il pivoting su y_{pq}
    end if
  end if
end while

Nel caso in cui una SAB non sia disponibile, è necessario applicare la 'fase I' del simplesso. Si introduca un nuovo problema, chiamato problema artificiale:

$$\min \sum_{i=1}^{m} y_i$$

$$A\overline{x} + \overline{y} = \overline{b}$$

$$\overline{x} \geq \overline{0}$$

$$\overline{y} \geq \overline{0}$$

dove $\overline{y}^T = (y_1, ..., y_m)$ è il vettore delle variabili artificiali.
Tale problema presenta una soluzione ammissibile di base da cui partire (formata dalle variabili artificiali) ed è dunque risolvibile con il metodo del simplesso. Se esiste una soluzione ammissibile del problema di partenza, allora il problema artificiale ha come soluzione ottima $\overline{y} = \overline{0}$, con funzione obiettivo ottima uguale a zero. In quest'ultimo caso il tableau che rappresenta tale soluzione sul problema originale si ottiene eliminando dal tableau del problema artificiale le colonne relative alle variabili y_i, sostituendo l'ultima riga con la riga dei costi del problema originario ed infine effettuando delle operazioni di pivot sugli 1 delle colonne di base.

2.1.2 Esercizi risolti

ESEMPIO 2.1. Sia dato il seguente problema di programmazione lineare.

$$\max\ 5x_1 + 4x_2$$

s.t.

$$x_1 + 2x_2 \leq 4$$

$$x_1 + \frac{1}{3}x_2 \leq 2$$

$$x_1, x_2 \geq 0.$$

(i) Risolvere graficamente il problema.

(ii) Trasformare il problema in forma standard e risolverlo applicando l'algoritmo del simplesso.

(iii) Identificare e evidenziare i punti nel piano (x_1, x_2) corrispondenti alle soluzioni di base visitate dall'algoritmo del simplesso.

Soluzione (i) Nel piano (x_1, x_2) si può disegnare il poligono $OABC$, che contiene l'insieme delle soluzioni ammissibili del problema. Le rette isocosto (tratteggiate nel grafico) sono perpendicolari al vettore $\overline{c} = (5, 4)$ — che dà la direzione di crescita della funzione obiettivo. Poiché il problema richiede di *massimizzare,* occorre spostare la retta isocosto il più possibile in direzione $(5, 4)$, mantenendo un'intersezione non vuota con $OABC$.

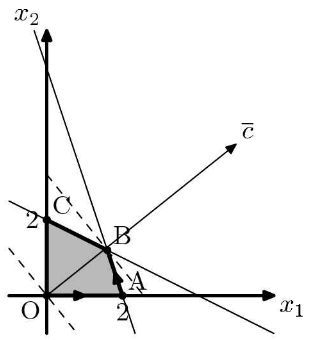
$$\max\ 5x_1 + 4x_2$$

s.t.

$$x_1 + 2x_2 \leq 4$$

$$x_1 + \frac{1}{3}x_2 \leq 2$$

$$x_1, x_2 \geq 0.$$



L'ottimo del problema si trova nel punto $B(x_1 = \frac{8}{5}, x_2 = \frac{6}{5})$, con valore $z = \frac{64}{5}$.

(ii) In forma standard, il problema si riscrive come

$$\min\ -5x_1 - 4x_2$$

$$\text{s.t.}$$

$$x_1 + 2x_2 + x_3 = 4$$

$$x_1 + \frac{1}{3}x_2 + x_4 = 2$$

$$x_1, \ldots, x_4 \geq 0.$$

Eseguendo l'algoritmo del simplesso partendo dalla base ammissibile (x_3, x_4) si ottiene quanto segue.

	x_1	x_2	x_3	x_4	
x_3	1	2	1	0	4
x_4	$\boxed{1}$	$\frac{1}{3}$	0	1	2
	-5	-4	0	0	0

La variabile da portare in base è la x_4, con il costo ridotto più negativo; la variabile uscente è la x_4.

	x_1	x_2	x_3	x_4	
x_3	0	$\boxed{\frac{5}{3}}$	1	-1	2
x_1	1	$\frac{1}{3}$	0	1	2
	0	$-\frac{7}{3}$	0	5	-10

Entra in base la x_2, esce la x_3.

	x_1	x_2	x_3	x_4	
x_2	0	1	$\frac{3}{5}$	$-\frac{3}{5}$	$\frac{6}{5}$
x_1	1	0	$-\frac{1}{5}$	$\frac{6}{5}$	$\frac{8}{5}$
	0	0	$\frac{7}{5}$	$\frac{18}{5}$	$-\frac{64}{5}$

La soluzione è ottima, avendo $r_3, r_4 \geq 0$.
(iii) I punti visitati dall'algoritmo sono, nell'ordine: $O(x_1 = 0, x_2 = 0)$ — quando x_1, x_2 sono entrambe fuori base, $A(x_1 = 2, x_2 = 0)$, e infine $B(x_1 = \frac{8}{5}, x_2 = \frac{6}{5})$.

ESEMPIO 2.2. Risolvere graficamente il seguente problema di programmazione lineare.

$$\min \; -4x_1 - 3x_2$$

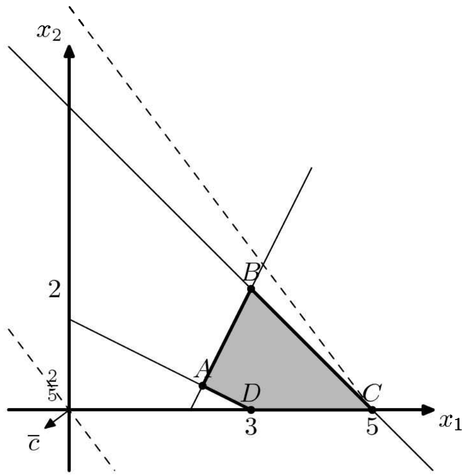
$$x_1 + 2x_2 \geq 3$$

$$2x_1 - x_2 \geq 4$$

$$x_1 + x_2 \leq 5$$

$$x_1, x_2 \geq 0.$$

Soluzione Disegnando l'intersezione dei semipiani $x_1 + 2x_2 \geq 3$, $2x_1 - x_2 \geq 4$, $x_1 + x_2 \leq 5$ nel primo quadrante del piano (x_1, x_2) si ottiene il poligono $ABCD$, che contiene le soluzioni ammissibili del problema.



Le rette isocosto sono ortogonali al vettore $\overline{c} = (-4, -3)$. Poiché il problema richiede di minimizzare la funzione obiettivo, si deve spostare una retta isocosto che interseca $ABCD$ il più possibile in direzione opposta a $\overline{c}$. Questa retta estrema interseca il poligono nel punto $C(x_1 = 5, x_2 = 0)$, che corrisponde alla soluzione ottima di costo $z = -20$.

ESEMPIO 2.3. Risolvere graficamente il seguente programma lineare.

$$\max\ -x_1 + 3x_2$$

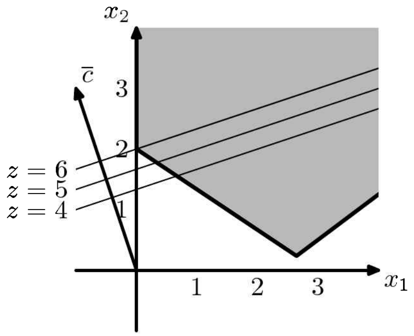
$$\text{s.t.}$$

$$2x_1 + 3x_2 \geq 6$$

$$3x_1 - 4x_2 \leq 7$$

$$x_1, x_2 \geq 0.$$

Soluzione Disegnando l'intersezione dei semipiani $2x_1 + 3x_2 \geq 6$ e $3x_1 - 4x_2 \leq 7$ con il primo quadrante del piano x_1, x_2 si ottiene il poligono illimitato riportato in figura.



Le rette isocosto sono ortogonali al vettore $\overline{c} = (-1, 3)$. Dovendo massimizzare, la retta isocosto va spinta il più possibile in direzione $\overline{c}$. Si vede dal disegno che è possibile spingerla arbitrariamente lontano senza che si svuoti la sua intersezione con l'insieme delle soluzoni ammissibili, quindi il problema ha funzione obiettivo illimitata.

ESEMPIO 2.4. Si risolva il seguente problema mediante l'algoritmo del simplesso:

$$\min 7x_1 + 2x_2 - 5x_3 - x_4$$

$$s.t.$$

$$4x_1 + 3x_2 + 2x_4 \leq 2$$

$$-5x_1 - 3x_2 + x_3 \leq 1$$

$$x_i \geq 0$$

Soluzione
Portiamo il problema in Forma Standard

$$\min 7x_1 + 2x_2 - 5x_3 - x_4$$

$$s.t.$$

$$4x_1 + 3x_2 + 2x_4 + x_5 = 2$$

$$-5x_1 - 3x_2 + x_3 + x_6 = 1$$

$$x_i \geq 0$$

Il tableau del simplesso è il seguente:

	x_1	x_2	x_3	x_4	x_5	x_6	
x_5	4	3	0	2	1	0	2
x_6	-5	-3	$\boxed{1}$	0	0	1	1
	7	2	-5	-1	0	0	0

Esiste una soluzione di base ammissibile, formata dalle variabili di slack x_5 e x_6.
La soluzione non è ottima poichè esistono $r_j < 0$ nelle colonne 3 e 4. Viene fatta entrare in base una variabile con costo ridotto negativo, in questo caso x_3. La variabile che esce viene indicata dal più piccolo dei rapporti $\frac{y_{i0}}{y_{y_{iq}}}$ avente denominatore non negativo o nullo. La colonna 3 presenta un solo elemento $y_{i3} > 0$ per cui la variabile che esce dalla base è nel nostro caso x_6.
Applicando le operazioni di pivot attorno all'elemento y_{23} si ottiene una nuova base (x_5, x_3) con il corrispondente tableau:

	x_1	x_2	x_3	x_4	x_5	x_6	
x_5	$\boxed{4}$	3	0	2	1	0	2
x_3	-5	-3	1	0	0	1	1
	-18	-13	0	-1	0	5	5

Abbiamo ancora dei costi ridotti negativi (colonne 1, 2 e 4). Scegliamo di far entrare in base x_1. La colonna 1 presenta un solo elemento $y_{i1} > 0$, per cui la variabile che esce dalla base è x_5 ed il pivot è $y_{11} = 4$.
Aggiornando il tableau mediante la regola del pivot, si ottiene:

	x_1	x_2	x_3	x_4	x_5	x_6	
x_1	1	$\frac{3}{4}$	0	$\frac{1}{2}$	$\frac{1}{4}$	0	$\frac{1}{2}$
x_3	0	$\frac{3}{4}$	1	$\frac{5}{2}$	$\frac{5}{4}$	1	$\frac{7}{2}$
	0	$\frac{1}{2}$	0	8	$\frac{9}{2}$	5	14

In questo caso il test di ottimalità è superato $(r_j \geq 0, \forall j)$ e la soluzione ottima è:

$$x_1 = \frac{1}{2}, x_3 = \frac{7}{2}, x_2 = x_4 = x_5 = x_6 = 0,$$

con funzione obiettivo $z = -14$.

ESEMPIO 2.5. Sia dato il seguente problema di PL:

$$\min 5x_1 + x_2 - \frac{1}{2}x_3$$

$$s.t.$$

$$x_1 + 5x_2 - 2x_3 = 15$$

$$x_1 - x_2 + x_3 = 3$$

$$x_i \geq 0$$

Soluzione
Non è identificabile una SAB (Soluzione Ammissibile di Base) di partenza. Applichiamo la fase I del simplesso:

$$\min y_1 + y_2$$

$$s.t.$$

$$x_1 + 5x_2 - 2x_3 + y_1 = 15$$

$$x_1 - x_2 + x_3 + y_2 = 3$$

$$x_i \geq 0, y_j \geq 0$$

Applichiamo il simplesso primale al tableau relativo al problema artificiale

	x_1	x_2	x_3	y_1	y_2	
y_1	1	5	-2	1	0	15
y_2	1	-1	1	0	1	3
	0	0	0	1	1	-18

Azzerando i costi ridotti delle variabili in base tramite operazioni di pivot, si ottiene il tableau seguente.

	x_1	x_2	x_3	y_1	y_2	
y_1	1	5	-2	1	0	15
y_2	$\boxed{1}$	-1	1	0	1	3
	-2	-4	1	0	0	-18

Le colonne 1 e 2 presentano costo ridotto negativo. Decidiamo di far entrare in base la variabile x_1. Calcoliamo la variabile che esce dalla base:

$$\varepsilon = \min_{y_{i1}>0} \left\{ \frac{15}{1}, \frac{3}{1} \right\} = 3.$$

La variabile che esce dalla base è y_2 ed il pivot è l'elemento $y_{21} = 1$.

	x_1	x_2	x_3	y_1	y_2	
y_1	0	$\boxed{6}$	-3	1	-1	12
x_1	1	-1	1	0	1	3
	0	-6	3	0	2	-12

La colonna 2 presenta un costo ridotto negativo, per cui facciamo entrare in base x_2, mentre la variabile che esce dalla base è y_1, essendo l'unico elemento non negativo della colonna 2 $y_{12} = 6$, che è anche il pivot. Aggiornando il tableau si ottiene:

	x_1	x_2	x_3	y_1	y_2	
x_2	0	1	$-\frac{1}{2}$	$\frac{1}{6}$	$-\frac{1}{6}$	2
x_1	1	0	$\frac{1}{2}$	$\frac{1}{6}$	$\frac{5}{6}$	5
	0	0	0	1	1	0

La soluzione ottima della Fase I è composta da sole variabili del problema originale, per cui tale soluzione è anche SAB del nostro problema. Possiamo a questo punto tornare al problema di partenza, considerando come base $x_2 = 2, x_1 = 5$.

Dal tableau finale della Fase I, otteniamo

	x_1	x_2	x_3	
x_2	0	$\boxed{1}$	$-\frac{1}{2}$	2
x_1	$\boxed{1}$	0	$\frac{1}{2}$	5
	5	1	$-\frac{1}{2}$	0

Facendo pivot su y_{12} e y_{21} per azzerare i costi ridotti delle variabili in base, otteniamo il tableau iniziale

	x_1	x_2	x_3	
x_2	0	1	$-\frac{1}{2}$	2
x_1	1	0	$\boxed{\frac{1}{2}}$	5
	0	0	$-\frac{5}{2}$	-27

Essendo il costo ridotto della colonna 3 negativo, facciamo entrare in base la variabile x_3. Calcoliamo la variabile che esce dalla base:

$$\varepsilon = \min_{y_{i3}>0} \left\{ \frac{5}{\frac{1}{2}} \right\} = 10.$$

La variabile che esce dalla base è x_1 ed il pivot è l'elemento $y_{23} = \frac{1}{2}$.

	x_1	x_2	x_3	
x_2	1	1	0	7
$x3$	2	0	1	10
	5	0	0	-2

Il test di ottimalità è superato $(r_j \geq 0, \forall j)$ e la soluzione ottima è:

$$x_2 = 7, x_3 = 10, x_1 = 0,$$

con funzione obiettivo $z = 2$.

ESEMPIO 2.6. Convertire il seguente modello di P.L. in forma standard:

$$\max 4x_1 + 3x_2 - x_3 - 7x_4$$

$$s.t.$$

$$x_1 - 3x_2 + x_3 - x_4 \geq 9$$

$$2x_1 + 2x_2 + \qquad x_4 \leq 5$$

$$x_1, x_4 \geq 0$$

$$x_3 \leq 0$$

Soluzione
Cambio di segno alla Funzione Obiettivo e pongo $x_2 = u - v$ e $x_3 = -w$. Ottengo quindi:

$$\min -4x_1 - 3(u-v) - z + 7x_4$$

$$s.t.$$

$$x_1 - 3(u-v) - w - x_4 - y_1 = 9$$

$$2x_1 + 2(u-v) \qquad + \ x_4 + y_2 = 5$$

$$x_1, x_4, y_1, y_2 \geq 0$$

$$u, v, w \geq 0$$

ESEMPIO 2.7. Risolvere il seguente P.L. con l'algoritmo del simplesso:

$$\max 3x_1 + 2x_2 - 5x_3$$

$$s.t.$$

$$4x_1 - 2x_2 + 2x_3 \leq 4$$

$$2x_1 - x_2 + x_3 \leq 1$$

$$x_i \geq 0$$

Soluzione
Ponendo in Forma Standard, si ottiene:

$$\min -3x_1 - 2x_2 + 5x_3$$

$$s.t.$$

$$4x_1 - 2x_2 + 2x_3 + x_4 = 4$$

$$2x_1 - x_2 + x_3 + x_5 = 1$$

$$x_i \geq 0$$

Una base iniziale è fornita dalle variabili di slack ed il tableau finale è il seguente:

	x_1	x_2	x_3	x_4	x_5	
x_4	4	-2	2	1	0	4
x_5	$\boxed{2}$	-1	1	0	1	1
	-3	-2	5	0	0	0

Essendo presenti costi ridotti negativi, non siamo ancora giunti all'ottimo. Decidiamo di far entrare in base la variabile x_1. Calcoliamo la variabile che esce dalla base:

$$\varepsilon = \min_{y_{i1}>0} \left\{ \frac{4}{1}, \frac{1}{2} \right\} = \frac{1}{2}.$$

La variabile che esce dalla base è x_5 ed il pivot è l'elemento $y_{21} = 2$.

	x_1	x_2	x_3	x_4	x_5	
x_4	0	0	0	1	-2	2
x_1	1	-$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$
	0	-$\frac{7}{2}$	$\frac{13}{2}$	0	$\frac{3}{2}$	$\frac{3}{2}$

La colonna 2 ha costo ridotto negativo e tutta la colonna è composta da elementi $y_{ij} \leq 0$, per cui il problema è illimitato.

ESEMPIO 2.8. Risolvere il seguente problema di Programmazione Lineare con l'algoritmo del simplesso:

$$\begin{array}{rl} \min & 5x_1 - 2x_2 - 3x_3 \\ \text{s.t.} & -x_1 + 4x_2 - 2x_3 \leq 8 \\ & x_1 + x_2 + 2x_3 \leq 12 \\ & x_1, x_2, x_3 \geq 0. \end{array}$$

Soluzione
Introduciamo le variabili di slack:

$$\begin{array}{rl} \min & 5x_1 - 2x_2 - 3x_3 \\ \text{s.t.} & -x_1 + 4x_2 - 2x_3 + x_4 = 8 \\ & x_1 + x_2 + 2x_3 + x_5 = 12 \\ & x_1, x_2, x_3 \geq 0. \end{array}$$

Il tableau iniziale è:

	x_1	x_2	x_3	x_4	x_5	
x_4	-1	4	-2	1	0	8
x_5	1	1	$\boxed{2}$	0	1	12
	5	-2	-3	0	0	0

Entra in base la variabile x_3 con costo ridotto -3. Esce la variabile x_5: il pivot è l'elemento $y_{33} = 2$. Il nuovo tableau è:

	x_1	x_2	x_3	x_4	x_5	
x_4	0	$\boxed{5}$	0	1	1	20
x_3	$\frac{1}{2}$	$\frac{1}{2}$	1	0	$\frac{1}{2}$	6
	$\frac{13}{2}$	$-\frac{1}{2}$	0	0	$\frac{3}{2}$	18

Entra in base la variabile x_2, esce la variabile x_4 ed il pivot è $y_{12} = 5$. Il nuovo tableau è:

	x_1	x_2	x_3	x_4	x_5	
x_2	0	1	0	$\frac{1}{5}$	$\frac{1}{5}$	4
x_3	$\frac{1}{2}$	0	1	$-\frac{1}{10}$	$\frac{2}{5}$	4
	$\frac{13}{2}$	0	0	$\frac{1}{10}$	$\frac{8}{5}$	20

La soluzione ottima è:

$$x_2 = 4, x_3 = 4, z = -20.$$

ESEMPIO 2.9. Risolvere il seguente problema di Programmazione Lineare con l'algoritmo del simplesso:

$$\begin{array}{ll} \min & -x_1 + 2x_2 - 3x_3 \\ \text{s.t.} & x_1 - 2x_2 + x_3 \leq 2 \\ & 3x_1 - x_2 + 2x_3 \leq 6 \\ & x_1, x_2, x_3 \geq 0. \end{array}$$

Soluzione
Introduciamo le variabili di slack:

$$\begin{array}{ll} \min & -x_1 + 2x_2 - 3x_3 \\ \text{s.t.} & x_1 - 2x_2 + x_3 + x_4 = 2 \\ & 3x_1 - x_2 + 2x_3 + x_5 = 6 \\ & x_1, x_2, x_3, x_4, x_5 \geq 0. \end{array}$$

Il tableau iniziale è:

	x_1	x_2	x_3	x_4	x_5	
x_4	1	-2	$\boxed{1}$	1	0	2
x_5	3	-1	2	0	1	6
	-1	2	-3	0	0	0

Entra in base la variabile x_3 con costo ridotto -3.

$$\varepsilon = \min_{y_{i3}>0} \{2, 3\} = 2.$$

Esce la variabile x_4 ed il pivot è l'elemento $y_{23} = 1$. Il nuovo tableau è:

	x_1	x_2	x_3	x_4	x_5	
x_3	1	-2	1	1	0	2
x_5	1	$\boxed{3}$	0	-2	1	2
	2	-4	0	3	0	6

Entra in base la variabile x_2 con costo ridotto -4, esce la variabile x_5 ed il pivot è l'elemento $y_{22} = 3$. Il nuovo tableau è:

	x_1	x_2	x_3	x_4	x_5	
x_3	$\frac{5}{3}$	0	1	$-\frac{1}{3}$	$\frac{2}{3}$	$\frac{10}{3}$
x_2	$\frac{1}{3}$	1	0	$-\frac{2}{3}$	$\frac{1}{3}$	$\frac{2}{3}$
	$\frac{10}{3}$	0	0	$\frac{1}{3}$	$\frac{4}{3}$	$\frac{26}{3}$

La soluzione ottima è:

$$x_2 = \frac{2}{3}, x_3 = \frac{10}{3}, z = -\frac{26}{3}.$$

ESEMPIO 2.10. Risolvere il seguente problema di programmazione lineare con l'algoritmo del simplesso:

$$\begin{array}{rl} \min & -2x_1 - x_2 - 3x_3 \\ \text{s.t.} & x_1 + x_2 + x_3 \leq 2 \\ & 2x_1 + 3x_2 + 8x_3 \leq 12 \\ & x_1, x_2, x_3 \geq 0. \end{array}$$

Soluzione
Introduciamo le variabili di slack:

$$\begin{array}{rl} \min & -2x_1 - x_2 - 3x_3 \\ \text{s.t.} & x_1 + x_2 + x_3 + x_4 = 2 \\ & 2x_1 + 3x_2 + 8x_3 + x_5 = 12 \\ & x_1, x_2, x_3 \geq 0. \end{array}$$

Il tableau iniziale è:

	x_1	x_2	x_3	x_4	x_5	
x_4	1	1	1	1	0	2
x_5	2	3	$\boxed{8}$	0	1	12
	-2	-1	-3	0	0	0

Entra in base la variabile x_3 con costo ridotto -3.

$$\varepsilon = \min_{y_{i3}>0} \left\{2, \frac{3}{2}\right\} = \frac{3}{2}.$$

Esce la variabile x_5 ed il pivot è l'elemento $y_{23} = 8$. Il nuovo tableau è:

	x_1	x_2	x_3	x_4	x_5	
x_4	$\boxed{\frac{3}{4}}$	$\frac{5}{8}$	0	1	$-\frac{1}{8}$	$\frac{1}{2}$
x_1	$\frac{1}{4}$	$\frac{3}{8}$	1	0	$\frac{1}{8}$	$\frac{3}{2}$
	$-\frac{5}{4}$	$\frac{1}{8}$	0	0	$\frac{3}{8}$	$\frac{9}{2}$

Entra in base la variabile x_1 con costo ridotto $-\frac{5}{4}$, esce la variabile x_4 ed il pivot è l'elemento $y_{11} = \frac{3}{4}$. Il nuovo tableau è:

	x_1	x_2	x_3	x_4	x_5	
x_4	1	$\frac{5}{6}$	0	$\frac{4}{3}$	$-\frac{1}{6}$	$\frac{2}{3}$
x_1	0	$\frac{1}{6}$	1	$-\frac{1}{3}$	$\frac{1}{6}$	$\frac{4}{3}$
	0	$\frac{7}{6}$	0	$\frac{5}{3}$	$\frac{1}{6}$	$\frac{16}{3}$

La soluzione ottima è:

$$x_1 = \frac{2}{3}, x_3 = \frac{4}{3}, z = -\frac{16}{3}.$$

2.2 Simplesso revisionato

2.2.1 Richiami teorici

Dato il seguente P.L.

$$\min \overline{c}^T \overline{x}$$

$$A\bar{x} = \bar{b}$$

$$\bar{x} \geq \overline{0}$$

Sia B una base ammissibile, B^{-1} la sua inversa e $\overline{x}_B = B^{-1}\overline{b}$ la corrispondente soluzione di base. Siano inoltre $\overline{c}_B^T$ i coefficienti di costo delle variabili in base, $\overline{c}_D^T$ quelli delle variabili fuori base, $\overline{r}_D^T$ i costi ridotti delle variabili fuori base. L'algoritmo del simplesso revisionato opera come segue:

Function Revisionato
ottimo = *illimitato* = **false**;
while ((*ottimo* = **false**) **and** (*illimitato* = **false**)) **do**
  Calcolare i moltiplicatori del simplesso:

$$\overline{\lambda}^T = \overline{c}_B^T B^{-1}$$

  ed i costi ridotti delle variabili fuori base

$$\overline{r}_D^T = \overline{c}_D^T - \overline{\lambda}^T D.$$

  if $\overline{r}_D \geq \overline{0}$ **then**
    $\overline{x} = (\overline{x}_B, \overline{0})$ è soluzione ottima del problema.
    ottimo = **true**
  else
    Scegliere $r_q < 0$. Il vettore $\overline{a}_q$ entra in base. Calcolare $\overline{y}_q = B^{-1}\overline{a}_q$ (è la colonna q.ma del tableau e rappresenta il vettore $\overline{a}_q$ espresso nei termini della base corrente).
    if $y_{iq} \leq 0, \forall i$ **then**
      Il problema è illimitato.
      illimitato = **true**
    else
      individuare il vettore che lascia la base calcolando i rapporti $\dfrac{y_{i0}}{y_{iq}}$, con $y_{iq} > 0$ e scegliendo il minimo tra tali valori.
      Sostituire in B la colonna relativa alla variabile che esce dalla base con quella relativa alla variabile che entra in base.
      calcolare la nuova matrice inversa B^{-1} e la corrispondente nuova soluzione di base $\overline{x}_B = B^{-1}\overline{b}$
    end if
  end if
end while

L'aggiornamento della matrice B^{-1} dopo la sostituzione di una colonna nella matrice B può essere effettuato velocemente utilizzando il "Metodo di Gauss": si costruisce un tableau di lavoro costituito dal vettore $\overline{y}_q$ che entra in base affiancato dalla matrice B^{-1} da aggiornare. Si effettua a questo punto una operazione di pivot sull'elemento y_{pq}. A seguito di tale operazione la nuova matrice B^{-1} si troverà nella stessa posizione della precedente B^{-1} nel tableau di lavoro.

2.2.2 Esercizi risolti

ESEMPIO 2.11. Si risolva il seguente problema mediante l'algoritmo del Simplesso Revisionato:

$$\begin{aligned}
&\min 3x_1 + x_2 - 2x_3 + x_4 \\
&s.t. \\
&2x_1 + x_2 - x_3 + 3x_4 \leq 8 \\
&- x_1 + 2x_2 - 2x_3 + 2x_4 = 4 \\
&x_i \geq 0
\end{aligned}$$

Soluzione
Non avendo una soluzione iniziale, costruiamo il problema artificiale

$$\begin{aligned}
&\min y_1 \\
&s.t. \\
&2x_1 + x_2 - x_3 + 3x_4 + x_5 = 8 \\
&- x_1 + 2x_2 - 2x_3 + 2x_4 + y_1 = 4 \\
&x_i \geq 0, y_j \geq 0
\end{aligned}$$

Risolviamo ora il problema artificiale con il simplesso revisionato:

$$A = \begin{bmatrix} 2 & 1 & -1 & 3 & 1 & 0 \\ -1 & 2 & -2 & 2 & 0 & 1 \end{bmatrix}$$

$$c^T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Base=$[x_5, y_1]$, $c_B = [0, 1]$, $c_D = [0, 0, 0, 0]$.

$$B^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} x_5 \\ y_1 \end{bmatrix} = \begin{bmatrix} 8 \\ 4 \end{bmatrix}$$

$$D = \begin{bmatrix} 2 & 1 & -1 & 3 \\ -1 & 2 & -2 & 2 \end{bmatrix}$$

Moltiplicatori e costi ridotti:

$$\lambda^T = c_B^T B^{-1} = [0,1] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = [0,1]$$

$$r_D^T = c_D^T - \lambda^T D = [0,0,0,0] - [0,1] \begin{bmatrix} 2 & 1 & -1 & 3 \\ -1 & 2 & -2 & 2 \end{bmatrix} = [1,-2,2,-2]\,.$$

Entra x_2.

$$y_0 = B^{-1}b = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 8 \\ 4 \end{bmatrix} = \begin{bmatrix} 8 \\ 4 \end{bmatrix}$$

$$y_2 = B^{-1}a_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$\varepsilon = \min\left\{\frac{8}{1}, \frac{4}{2}\right\} = 2.$$

Esce la variabile y_1.
La nuova base è $x_B = \{x_5, x_2\}$. Aggiorno la matrice B^{-1} utilizzando il metodo di Gauss.

$$\left[\begin{array}{cc|c} 1 & 0 & 1 \\ 0 & 1 & \boxed{2} \end{array}\right]$$

$$\left[\begin{array}{cc|c} 1 & -\frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 1 \end{array}\right]$$

La nuova B^{-1} è

$$B^{-1} = \begin{bmatrix} 1 & -\frac{1}{2} \\ 0 & \frac{1}{2} \end{bmatrix}.$$

Le altre grandezze valgono

$$c_B = [0,0]\,, c_D = [0,1,0,0]$$

$$D = \begin{bmatrix} 2 & 0 & -1 & 3 \\ -1 & 1 & -2 & 2 \end{bmatrix}.$$

Calcoliamo quindi i costi ridotti.

$$\lambda^T = c_B^T B^{-1} = [0,0] \begin{bmatrix} 1 & -\frac{1}{2} \\ 0 & \frac{1}{2} \end{bmatrix} = [0,0]$$

$$r_D^T = c_D^T - \lambda^T D = [0,1,0,0] - [0,0] \begin{bmatrix} 2 & 0 & -1 & 3 \\ -1 & 1 & -2 & 2 \end{bmatrix} = [0,1,0,0]$$

La soluzione è ottima per il problema artificiale e z=0. La base x_B è dunque ammissibile per il problema di partenza.

$$A = \begin{bmatrix} 2 & 1 & -1 & 3 & 1 \\ -1 & 2 & -2 & 2 & 0 \end{bmatrix}$$

$$c^T = \begin{bmatrix} 3 & 1 & -2 & 1 & 0 \end{bmatrix}$$

Base=$[x_5, x_2]$

$$B^{-1} = \begin{bmatrix} 1 & -\frac{1}{2} \\ 0 & \frac{1}{2} \end{bmatrix}$$

$$c_B = [0,1]\,, c_D = [3,-2,1]$$

$$D = \begin{bmatrix} 2 & -1 & 3 \\ -1 & -2 & 2 \end{bmatrix}.$$

Moltiplicatori e costi ridotti valgono quindi:

$$\lambda^T = c_B^T B^{-1} = [0, 1] \begin{bmatrix} 1 & -\frac{1}{2} \\ 0 & \frac{1}{2} \end{bmatrix} = \left[0, \frac{1}{2}\right]$$

$$r_D^T = c_D^T - \lambda^T D = [3, -2, 1] - \left[0, \frac{1}{2}\right] \begin{bmatrix} 2 & -1 & 3 \\ -1 & -2 & 2 \end{bmatrix} = \left[\frac{7}{2}, -1, 0\right]$$

Avendo costo ridotto negativo, entra in base x_3.

$$y_0 = B^{-1} b = \begin{bmatrix} 1 & -\frac{1}{2} \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 8 \\ 4 \end{bmatrix} = \begin{bmatrix} 6 \\ 2 \end{bmatrix}$$

$$y_3 = B^{-1} a_3 = \begin{bmatrix} 1 & -\frac{1}{2} \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} -1 \\ -2 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

Poichè $y_{iq} \leq 0, \forall i$, il problema è *illimitato.*

ESEMPIO 2.12. Si risolva il seguente problema mediante l'algoritmo del Simplesso Revisionato:

$$\min 5x_1 + x_2 - 3x_3$$

$$s.t.$$

$$x_1 - 2x_2 + 7x_3 = 18$$

$$x_1 - x_2 + 3x_3 = 7$$

$$x_i \geq 0$$

Soluzione

Non essendo immediatamente identificabile una soluzione di base iniziale, costruiamo il problema artificiale.

$$\min y_1 + y_2$$

$$s.t.$$

$$x_1 - 2x_2 + 7x_3 + y_1 = 18$$

$$x_1 - x_2 + 3x_3 + y_2 = 7$$

$$x_i \geq 0, y_j \geq 0.$$

$$A = \begin{bmatrix} 1 & -2 & 7 & 1 & 0 \\ 1 & -1 & 3 & 0 & 1 \end{bmatrix}$$

$$c^T = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Base=$[y_1, y_2]$

$$B = B^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$c_B = [1,1]\,, c_D = [0,0,0]\,.$$

$$D = \begin{bmatrix} 1 & -2 & 7 \\ 1 & -1 & 3 \end{bmatrix}$$

Moltiplicatori e costi ridotti:

$$\lambda^T = c_B^T B^{-1} = [1,1] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = [1,1]$$

$$r_D^T = c_D^T - \lambda^T D = [0,0,0] - [1,1] \begin{bmatrix} 1 & -2 & 7 \\ 1 & -1 & 3 \end{bmatrix} = [-2, 3, -10]$$

La variabile x_3 entra in base.

$$y_0 = B^{-1}b = \begin{bmatrix} 18 \\ 7 \end{bmatrix}$$

$$y_3 = B^{-1}a_3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 7 \\ 3 \end{bmatrix} = \begin{bmatrix} 7 \\ 3 \end{bmatrix}$$

$$\varepsilon = \min\left\{\frac{18}{7}, \frac{7}{3}\right\} = \frac{7}{3}.$$

La variabile y_2 esce dalla base.
Base=$[y_1, x_3]$.
Aggiorno la B^{-1} col metodo di Gauss.

$$\left[\begin{array}{cc|c} 1 & 0 & 7 \\ 0 & 1 & \boxed{3} \end{array}\right]$$

$$\left[\begin{array}{cc|c} 1 & -\frac{7}{3} & 0 \\ 0 & \frac{1}{3} & 1 \end{array}\right]$$

$$B^{-1} = \begin{bmatrix} 1 & -\frac{7}{3} \\ 0 & \frac{1}{3} \end{bmatrix}$$

$$c_B = [1, 0]\,, c_D = [0, 0, 1]$$

$$D = \begin{bmatrix} 1 & -2 & 0 \\ 1 & -1 & 1 \end{bmatrix}$$

Moltiplicatori e costi ridotti valgono:

$$\lambda^T = c_B^T B^{-1} = [1, 0] \begin{bmatrix} 1 & -\frac{7}{3} \\ 0 & \frac{1}{3} \end{bmatrix} = \left[1, -\frac{7}{3}\right]$$

$$r_D^T = c_D^T - \lambda^T D = [0, 0, 1] - \left[1, -\frac{7}{3}\right] \begin{bmatrix} 1 & -2 & 0 \\ 1 & -1 & 1 \end{bmatrix} = \left[\frac{4}{3}, -\frac{1}{3}, \frac{10}{3}\right]$$

Esiste ancora un costo ridotto negativo, per cui la variabile x_2 entra in base.

$$y_0 = B^{-1}b = \begin{bmatrix} \frac{5}{3} \\ \frac{7}{3} \end{bmatrix}$$

$$y_2 = B^{-1}a_2 = \begin{bmatrix} 1 & -\frac{7}{3} \\ 0 & \frac{1}{3} \end{bmatrix} \begin{bmatrix} -2 \\ -1 \end{bmatrix} = \begin{bmatrix} \frac{1}{3} \\ -\frac{1}{3} \end{bmatrix}.$$

Avendo la colonna y_2 solo l'elemento $y_{12} > 0$, la variabile ad uscire dalla base è y_1, per cui la nuova base è $x_B = [x_2, x_3]$.
Passo 2
Base $x_B = [x_2, x_3]$.
Aggiorno la B^{-1} col metodo di Gauss.

$$\left[\begin{array}{cc|c} 1 & -\frac{7}{3} & \frac{1}{3} \\ 0 & \frac{1}{3} & -\frac{1}{3} \end{array}\right]$$

$$\left[\begin{array}{cc|c} 3 & -7 & 1 \\ 1 & -2 & 0 \end{array}\right]$$

$$B^{-1} = \begin{bmatrix} 3 & -7 \\ 1 & -2 \end{bmatrix}$$

$$c_B = [0, 0], c_D = [0, 1, 1]$$

$$D = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Moltiplicatori e costi ridotti valgono:

$$\lambda^T = c_B^T B^{-1} = [0, 0] \begin{bmatrix} 3 & -7 \\ 1 & -2 \end{bmatrix} = [0, 0]$$

$$r_D^T = c_D^T - \lambda^T D = [0, 1, 1] - [0, 0] \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = [0, 1, 1].$$

Essendo tutti i costi ridotti non-negativi, il problema di fase I è giunto all'ottimo. La base ottima trovata è $x_B = [x_2, x_3]$, che, non contenendo le variabili artificiali, è anche soluzione ammissibile del problema originario.
Riconsideriamo il problema di partenza. Esso ha la seguente matrice dei coefficienti:

$$A = \begin{bmatrix} 1 & -2 & 7 \\ 1 & -1 & 3 \end{bmatrix}.$$

Essendo la base iniziale $x_B = [x_2, x_3]$, otteniamo

$$B^{-1} = \begin{bmatrix} 3 & -7 \\ 1 & -2 \end{bmatrix}$$

$$c_B = [1, -3]\,, c_D = [5]$$

$$D = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Moltiplicatori e costi ridottidiventano quindi:

$$\lambda^T = c_B^T B^{-1} = [1, -3] \begin{bmatrix} 3 & -7 \\ 1 & -2 \end{bmatrix} = [0, -1]$$

$$r_D^T = c_D^T - \lambda^T D = [5] - [0, -1] \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 6 \geq 0$$

Essendo tutti i costi ridotti non-negativi, il problema è stato risolto all'ottimo.

$$x_B = B^{-1} b = \begin{bmatrix} 3 & -7 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} 18 \\ 7 \end{bmatrix} = \begin{bmatrix} 5 \\ 4 \end{bmatrix}$$

$$z = c_B x_B = -7.$$

La soluzione ottima è $x_1 = 0, x_2 = 5, x_3 = 4$ con funzione obiettivo $z = -7$.

ESEMPIO 2.13. Risolvere il seguente problema di P.L. con l'algoritmo del simplesso revisionato.

$$\min -x_1 - 3x_2 + x_3$$

$$s.t.$$

$$2x_1 + x_2 \leq 3$$

$$x_1 + x_2 + 3x_3 \leq 6$$

$$2x_1 + x_2 + 3x_3 \leq 8$$

$$x_i \geq 0$$

Soluzione
Portando il problema in forma standard si ottiene:

$$\min -x_1 - 3x_2 + x_3$$

$$s.t.$$

$$2x_1 + x_2 - x_4 = 3$$

$$x_1 + x_2 + 3x_3 + x_5 = 6$$

$$2x_1 + x_2 + 3x_3 + x_6 = 8$$

$$x_i \geq 0$$

Una SAB può essere ottenuta grazie alle variabili di slack x_4, x_5 ed x_6.
I dati del problema sono:

$$A = \begin{bmatrix} 2 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 3 & 0 & 1 & 0 \\ 2 & 1 & 3 & 0 & 0 & 1 \end{bmatrix}$$

$$B^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$c_B = [0, 0, 0]\,, c_D = [-1, -3, 1]$$

$$D = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & 3 \\ 2 & 1 & 3 \end{bmatrix}.$$

Ricaviamo i valori dei moltiplicatori e dei costi ridotti delle variabili non in base:

$$\lambda^T = c_B^T B^{-1} = [0, 0, 0] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = [0, 0, 0]$$

$$r_D^T = c_D^T - \lambda^T D = [-1, -3, 1] - [0, 0, 0] \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & 3 \\ 2 & 1 & 3 \end{bmatrix} = [-1, -3, 1]\,.$$

I costi ridotti relativi alle variabili x_1 ed x_2 sono negativi. Decidiamo di far entrare in base x_2 e calcoliamo la variabile che esce dalla base.

$$y_0 = B^{-1} b = \begin{bmatrix} 3 \\ 6 \\ 8 \end{bmatrix}$$

$$y_2 = B^{-1} a_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\varepsilon = \min \left\{ \frac{3}{1}, \frac{6}{1}, \frac{8}{1} \right\} = 3.$$

La variabile che esce dalla base è x_4, per cui la nuova base è $x_B = \{x_2, x_5, x_6\}$.
Base $[x_2, x_5, x_6]$.
Aggiorno la B^{-1} col metodo di Gauss.

$$\left[\begin{array}{ccc|c} 1 & 0 & 0 & \boxed{1} \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{array}\right]$$

$$\left[\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \end{array}\right]$$

Le nuove matrici da utilizzare sono quindi

$$B^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

$$c_B = [-3, 0, 0]\,, c_D = [-1, 0, 1]$$

$$D = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & 3 \\ 2 & 0 & 3 \end{bmatrix}.$$

I valori dei moltiplicatori e dei costi ridotti delle variabili non in base sono:

$$y_0 = B^{-1}b = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 6 \\ 8 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \\ 5 \end{bmatrix}$$

$$\lambda^T = c_B^T B^{-1} = [-3, 0, 0] \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} = [-3, 0, 0]$$

$$r_D^T = c_D^T - \lambda^T D = [-1, 0, 1] - [-3, 0, 0] \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & 3 \\ 2 & 0 & 3 \end{bmatrix} = [5, 3, 1]\,.$$

Essendo tutti i costi ridotti non negativi, abbiamo trovato una SAB ottima, che vale $x_1 = 0$, $x_2 = 3$, $x_3 = 0$, $x_4 = 0$, $x_5 = 3$ e $x_6 = 5$, con funzione obiettivo $z = c_B x_B = -9$.

ESEMPIO 2.14. Calcolare con l'algoritmo del simplesso revisionato la soluzione ottima del seguente problema di programmazione lineare:

$$\begin{array}{rl} \max & x_1 + 3x_2 + 5x_3 + 2x_4 \\ \text{s.t.} & x_1 + 2x_2 + 3x_3 + x_4 \leq 3 \\ & 2x_1 + x_2 + x_3 + 2x_4 \leq 4 \\ & x_1, x_2, x_3, x_4 \in \mathbb{R}^+. \end{array}$$

Soluzione
Portando in forma standard, otteniamo il seguente nuovo problema.

$$\begin{array}{rl} \min & -x_1 - 3x_2 - 5x_3 - 2x_4 \\ \text{s.t.} & x_1 + 2x_2 + 3x_3 + x_4 + x_5 = 3 \\ & 2x_1 + x_2 + x_3 + 2x_4 + x_6 = 4 \\ & x_1, x_2, x_3, x_4 \in \mathbb{R}^+. \end{array}$$

Base $[x_5, x_6]$

$$B = I = B^{-1}$$

$$c_B^T = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

$$c_D^T = \begin{bmatrix} -1 & -3 & -5 & -2 \end{bmatrix}$$

$$y_0^T = \begin{bmatrix} 3 & 4 \end{bmatrix}$$

$$D = \begin{bmatrix} 1 & 2 & 3 & 1 \\ 2 & 1 & 1 & 2 \end{bmatrix}$$

$$r_D^T = c_D^T = \begin{bmatrix} -1 & -3 & -5 & -2 \end{bmatrix}$$

Scegliamo di far entrare in base x_3.

$$y_3 = B^{-1}a_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

$$\varepsilon = \min\left\{\frac{3}{3}, \frac{4}{1}\right\} = 1.$$

Esce dalla base la variabile che corrisponde alla prima colonna della matrice B (variabile di slack x_5). Aggiorniamo tutti i valori.
Base $[x_3, x_6]$

$$B = \begin{bmatrix} 3 & 0 \\ 1 & 1 \end{bmatrix}$$

$$B^{-1} = \begin{bmatrix} \frac{1}{3} & 0 \\ -\frac{1}{3} & 1 \end{bmatrix}$$

$$y_0 = x_B = B^{-1} \cdot b = [\ 1 \quad 3\ ]$$
$$c_B^T = [\ -5 \quad 0\ ]$$
$$c_D^T = [\ -1 \quad -3 \quad 0 \quad -2\ ]$$
$$\lambda^T = c_B^T \cdot B^{-1} = [\ -\tfrac{5}{3} \quad 0\ ]$$
$$D = \begin{bmatrix} 1 & 2 & 1 & 1 \\ 2 & 1 & 0 & 2 \end{bmatrix}$$
$$r_D^T = c_D^T - \lambda^T \cdot D = [\ \tfrac{2}{3} \quad \tfrac{1}{3} \quad \tfrac{5}{3} \quad -\tfrac{1}{3}\ ]$$

Entra x_4: la colonna che entra espressa in funzione della base corrente è $y_4 = B^{-1} \cdot a_4$ dove

$$a_4 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

quindi

$$y_4 = \begin{bmatrix} \frac{1}{3} \\ \frac{5}{3} \end{bmatrix}$$

Esce dalla base la variabile x_6, che corrisponde alla seconda colonna di B.
Base $[x_3, x_4]$

$$B = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}$$
$$B^{-1} = \begin{bmatrix} \frac{2}{5} & -\frac{1}{5} \\ -\frac{1}{5} & \frac{3}{5} \end{bmatrix}$$
$$y_0 = x_B = B^{-1} \cdot b = [\ \tfrac{2}{5} \quad \tfrac{9}{5}\ ]$$
$$c_B^T = [\ -5 \quad -2\ ]$$
$$c_D^T = [\ -1 \quad -3 \quad 0 \quad 0\ ]$$
$$\lambda^T = c_B^T \cdot B^{-1} = [\ -\tfrac{8}{5} \quad -\tfrac{1}{5}\ ]$$
$$D = \begin{bmatrix} 1 & 2 & 1 & 0 \\ 2 & 1 & 0 & 1 \end{bmatrix}$$
$$r_D^T = c_D^T - \lambda^T \cdot D = [\ 1 \quad \tfrac{2}{5} \quad \tfrac{8}{5} \quad \tfrac{1}{5}\ ]$$

I costi ridotti sono tutti positivi: la soluzione ottima è

$$x_B = [\ \tfrac{2}{5} \quad \tfrac{9}{5}\ ]$$

e il valore della soluzione ottima è $\frac{28}{5}$.

2.3 Simplesso duale

2.3.1 Richiami teorici

Il simplesso duale parte da una soluzione ammissibile per il problema duale ($r_j \geq 0, j = 0, .., n$) ma non ammissibile per il problema primale (esiste nell'ultima colonna del tableau almeno un termine $x_p < 0$). Tale situazione si può verificare in pratica per una serie di motivi, tra cui:

- il cambio di un termine noto che rende l'ultima colonna parzialmente negativa;
- al posto della fase 1 in un problema di minimo con tutti i costi $c_i \geq 0$ (altrimenti perdo l'ottimalità). In tal caso la SAB trovata è anche ottima;
- aggiunta di vincoli (Branch & Bound).

L'idea dell'algoritmo è quella di effettuare le solite operazioni di pivot, mantenendo l'ottimalità (costi ridotti tutti maggiori o uguali a zero) ma muovendosi verso l'ammissibilità. Il procedimento, visto dal punto di vista duale, è equivalente a mantenere l'ammissibilità duale muovendosi verso l'ottimalità duale. L'algoritmo è riassunto nel seguente pseudocodice:

Function Simplesso Duale
Si supponga di avere un tableau con base ammissibile per il duale ($r_j \geq 0, j = 1, ..., n$), ma non per il primale ($\exists p > 0 : y_{p0<0}$)

ottimo = *illimitato* = **false**;
while ottimo = **false and** illimitato = **false do**
  if $y_{pj} \geq 0, \forall j = 1, ..., n$ **then**
    illimitato = **true** (duale illimitato, primale non ammissibile)
  else
    $\epsilon = \min_{y_{pj}<0} \left\{ \frac{r_j}{|y_{pj}|} \right\} = \frac{y_{0q}}{y_{pq}}$
    Nella base la variabile x_q entra al posto della variabile x_p: eseguire l'operazione di pivot su y_{pq}.
  end if
end while

2.3.2 Esercizi risolti

ESEMPIO 2.15. Calcolare, applicando l'algoritmo del simplesso duale, la soluzione ottima del seguente problema di Programmazione Lineare:

$$\begin{array}{rl} \min & 5x_1 + 3x_2 + 8x_3 + 3x_4 \\ \text{s.t.} & 3x_1 + 2x_2 + 4x_3 + 2x_4 \geq 6 \\ & -x_1 + 3x_2 + 5x_3 + x_4 \geq 5 \\ & x_1, x_2, x_3, x_4 \in \mathbb{R}^+. \end{array}$$

Soluzione
Il tableau iniziale, dopo aver aggiunto le variabili di surplus e aver cambiato i segni, risulta:

	x_1	x_2	x_3	x_4	x_5	x_6	
x_5	-3	$\boxed{-2}$	-4	-2	1	0	-6
x_6	1	-3	-5	-1	0	1	-5
	5	3	8	3	0	0	0

La soluzione data da $x_5 = -6$, $x_6 = -5$ è ottima, ma non ammissibile (le variabili in base sono negative). Applichiamo il simplesso duale. Selezioniamo una riga con y_{p0} negativo ($p = 1$). Selezioniamo la colonna q tale che

$$q = \arg\min_j \left\{ \frac{r_j}{|y_{pj}|}, y_{pj} < 0 \right\}.$$

Le colonne che presentano il minimo sono la colonna 2 e la colonna 4. Esce dalla base x_5, entra x_2. Dopo l'operazione di pivot, il tableau diventa:

	x_1	x_2	x_3	x_4	x_5	x_6	
x_2	$\frac{3}{2}$	1	2	1	$-\frac{1}{2}$	0	3
x_6	$\frac{11}{2}$	0	1	2	$-\frac{3}{2}$	1	4
	$\frac{1}{2}$	0	2	0	$\frac{3}{2}$	0	-9

La soluzione trovata è sia ottima, che ammissibile (le variabili assumono valori tutti positivi): $x_2 = 3$, $x_6 = 4$, $z = 9$.

ESEMPIO 2.16. Si trovi la soluzione ottima del seguente PL:

$$\min 2x_1 + 3x_2 + x_3$$

$$s.t.$$

$$x_1 - 2x_2 \geq 5$$

$$3x_1 - x_2 + 5x_3 \leq 2$$

$$x_i \geq 0$$

Soluzione
Portiamo il problema in forma standard aggiungendo una variabile di surplus ed una di slack

$$\min 2x_1 + 3x_2 + x_3$$

$$s.t.$$

$$x_1 - 2x_2 - x_4 = 5$$

$$3x_1 - x_2 + 5x_3 + x_5 = 2$$

$$x_i \geq 0$$

Essendo un problema di minimo con coefficienti tutti non-negativi, possiamo applicare il simplesso duale.
Il tableau è il seguente:

	x_1	x_2	x_3	x_4	x_5	
x_4	$\boxed{-1}$	2	0	1	0	-5
x_5	3	-1	5	0	1	2
	2	3	1	0	0	0

Abbiamo una riga con termine noto negativo (riga 1), per cui facciamo uscire dalla base x_4. Per decidere chi entrerà in base, calcoliamo

$$\varepsilon = \min_{y_{1j}<0} \left\{ \frac{r_j}{|y_{1j}|} \right\} = \min \left\{ \frac{1}{2} \right\} = \frac{1}{2}.$$

Esce dalla base x_4, entra x_1 ed il pivot è l'elemento $y_{11} = -1$.

	x_1	x_2	x_3	x_4	x_5	
x_1	1	-2	0	-1	0	5
x_5	0	5	5	3	1	-13
	0	7	1	2	0	-10

Avendo ancora la riga 2 con termine noto negativo, non siamo ancora arrivati all'ammissibilità della soluzione. Facciamo uscire la variabile x_5.
Tutti i termini y_{2j} della riga sono positivi, per cui il problema duale è illimitato ed il primale non ammette soluzioni.

ESEMPIO 2.17. Trovare la soluzione ottima del seguente problema mediante l'algoritmo del simplesso duale

$$\min x_1 + 3x_2 + 4x_3$$

$$s.t.$$

$$x_1 + 2x_2 + x_3 \leq 3$$

$$2x_1 + x_2 + 2x_3 \leq 2$$

$$2x_1 + 2x_2 \geq 5$$

$$x_i \geq 0$$

Soluzione
Portando in forma standard il problema, otteniamo:

$$\min x_1 + 3x_2 + 4x_3$$

$$s.t.$$

$$x_1 + 2x_2 + x_3 + x_4 = 3$$

$$2x_1 + x_2 + 2x_3 + x_5 = 2$$

$$2x_1 + 2x_2 - x_6 = 5$$

$$x_i \geq 0$$

Essendo un problema di minimo con coefficienti tutti non-negativi, possiamo applicare il simplesso duale.
Costruiamo il tableau

	x_1	x_2	x_3	x_4	x_5	x_6	
x_4	1	2	1	1	0	0	3
x_5	2	1	2	0	1	0	2
x_6	$\boxed{-2}$	-2	0	0	0	1	-5
	1	3	4	0	0	0	0

Abbiamo una riga con termine noto negativo (riga 3), per cui facciamo uscire dalla base x_6. Per decidere chi entrerà in base, calcoliamo

$$\varepsilon = \min_{y_{3j}<0} \left\{ \frac{r_j}{|y_{1j}|} \right\} = \min \left\{ \frac{1}{2}, \frac{3}{2} \right\} = \frac{1}{2}.$$

Entra in base la variabile x_1 ed il pivot è l'elemento $y_{31} = -2$.

	x_1	x_2	x_3	x_4	x_5	x_6	
x_4	0	1	1	1	0	$\frac{1}{2}$	$\frac{1}{2}$
x_5	0	$\boxed{-1}$	2	0	1	1	-3
x_1	1	1	0	0	0	$-\frac{1}{2}$	$\frac{5}{2}$
	0	2	4	0	0	$\frac{1}{2}$	$-\frac{5}{2}$

Abbiamo una riga con termine noto negativo (riga 2), per cui facciamo uscire dalla base x_5. Per decidere chi entrerà in base, calcoliamo

$$\varepsilon = \min_{y_{2j}<0} \left\{ \frac{r_j}{|y_{1j}|} \right\} = \min \left\{ \frac{2}{1} \right\} = \frac{1}{2}.$$

Esce dalla base x_5, entra x_2 ed il pivot è l'elemento $y_{22} = -1$.

	x_1	x_2	x_3	x_4	x_5	x_6	
x_4	0	0	3	1	1	$\frac{3}{2}$	$-\frac{5}{2}$
x_2	0	1	-2	0	-1	-1	3
x_1	1	0	2	0	1	$\frac{1}{2}$	$-\frac{1}{2}$
	0	0	8	0	2	$\frac{5}{2}$	$\frac{17}{2}$

Avendo ancora la riga 1 con termine noto negativo, non siamo ancora arrivati all'ammissibilità della soluzione. Tutti i termini y_{1j} della riga sono positivi, per cui il problema duale è illimitato ed il primale non ammette soluzioni.

2.4 Simplesso Primale-Duale

2.4.1 Richiami teorici

Si supponga di voler risolvere il seguente problema di P.L. in forma standard (PP):

$$
\begin{aligned}
(\text{PP}) \qquad & \min \overline{c}^T \overline{x} \\
& A\overline{x} = \overline{b} \\
& \overline{x} \geq \overline{0},
\end{aligned}
$$

a cui corrisponde il duale (PD)

$$
\begin{aligned}
(\text{PD}) \qquad & \max \overline{\lambda}^T \overline{b} \\
& \overline{\lambda}^T A \leq \overline{c}^T.
\end{aligned}
$$

Si definisca "Problema primale ristretto" (PPR) il seguente problema:

$$
\begin{aligned}
(\text{PPR}) \qquad & \min \overline{1}^T \overline{y} \\
& A\overline{x} + \overline{y} = \overline{b} \\
& x_j = 0, \forall j \notin J \\
& \overline{x} \geq \overline{0}, \overline{y} \geq \overline{0}.
\end{aligned}
$$

dove J è l'insieme dei vincoli soddisfatti all'uguaglianza da una soluzione ammissibile duale λ:

$$J = \left\{ j : \overline{\lambda}^T \overline{a}_j = c_j \right\} = \left\{ j : r_j = c_j - \overline{\lambda}^T \overline{a}_j = 0 \right\}.$$

L'algoritmo del simplesso primale-duale parte da una soluzione ammissibile per il problema duale PD, ricava il relativo problema primale ristretto PPR e lo risolve. Se la soluzione ottima di (PPR) è ammissibile anche per (PP), si è trovato l'ottimo di (PP), altrimenti la soluzione ammissibile per (PD) viene migliorata e viene generato un nuovo (PPR).

Function Simplesso Primale-Duale
Si supponga di avere una soluzione ammissibile per il duale λ

$ottimo = infeasible =$ **false**;
while ottimo = **false and** infeasible = TRUE **do**
  Costruire il Problema Primale Ristretto.
  Risolvere il Problema Primale Ristretto mediante il Simplesso: sia μ la soluzione del problema duale del PPR.
  if $z_{PPR} = 0$ **then**
    $ottimo =$ **true**
  else
    if $\mu^T a_j \leq 0, \forall j \notin J$ **then**
      $infeasible =$ **true**
    else
      $\epsilon = min_{j \notin J : \mu^T a_j > 0} \left\{ \frac{c_j - \lambda^T a_j}{\mu^T a_j} \right\}$
      $\lambda = \lambda + \epsilon\mu$
    end if
  end if
end while

2.4.2 Esercizi risolti

ESEMPIO 2.18. Risolvere il seguente problema mediante l'algoritmo del simplesso primale-duale:

$$\min 6x_1 + x_2 + 3x_3$$

$$s.t.$$

$$10x_1 - 2x_2 + 5x_3 = 15$$

$$x_1 - x_2 + 3x_3 = 6$$

$$x_i \geq 0$$

Soluzione
Il Problema Duale (PD) è

$$\max 15\lambda_1 + 6\lambda_2$$

$$s.t.$$

$$10\lambda_1 + \lambda_2 \leq 6$$

$$-2\lambda_1 - \lambda_2 \leq 1$$

$$5\lambda_1 + 3\lambda_2 \leq 3$$

$$\lambda_i \quad libere.$$

Le matrici del problema sono

$$A = \begin{bmatrix} 10 & -2 & 5 \\ 1 & -1 & 3 \end{bmatrix}$$

$$b = \begin{bmatrix} 15 \\ 6 \end{bmatrix}$$

$$c^T = [6, 1, 3].$$

Non avendo una soluzione ammissibile di partenza per il duale, consideriamo

$$\lambda = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

$$\begin{array}{c} 10\lambda_1 + \lambda_2 = 0 < 6 \Rightarrow x_1 = 0 \\ -2\lambda_1 - \lambda_2 = 0 < 1 \Rightarrow x_2 = 0 \\ 5\lambda_1 + 3\lambda_2 = 0 < 3 \Rightarrow x_3 = 0. \end{array}$$

Nessun vincolo viene soddisfatto all'uguaglianza da λ, per cui $J = \{\emptyset\}$.
Il PPR diventa, quindi

$$\min y_1 + y_2$$

$$s.t.$$

$$\begin{array}{cccc} y_1 & & = & 15 \\ & y_2 & = & 6 \end{array}$$

$$y_j \geq 0$$

Una soluzione iniziale al PPR è $x_B = \{y_1, y_2\}$. Tale base è anche quella ottima del problema PPR e presenta

$$B = B^{-1} = I.$$

$$\mu = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

La funzione obiettivo non è nulla, per cui non abbiamo ancora trovato la soluzione ottima del PP. Aggiorniamo la soluzione duale λ.

$$c^T - \lambda^T A = [6,1,3] - [0,0] \begin{bmatrix} 10 & -2 & 5 \\ 1 & -1 & 3 \end{bmatrix} = [6,1,3]$$

$$\mu^T A = [1,1] \begin{bmatrix} 10 & -2 & 5 \\ 1 & -1 & 3 \end{bmatrix} = [11,-3,8]$$

$$\varepsilon = \min_{\mu_j^T a_j > 0} \left\{ \frac{c_j^T - \lambda j^T a_j}{\mu_j^T a_j} \right\} = \left\{ \frac{6}{11}, \frac{3}{8} \right\} = \frac{3}{8}.$$

$$\lambda = \lambda + \varepsilon\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \frac{3}{8} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{3}{8} \\ \frac{3}{8} \end{bmatrix}.$$

La nuova soluzione duale è $\lambda^T = \left[\frac{3}{8}, \frac{3}{8}\right]$. Controlliamo il valore dei vincoli del PD.

$$\begin{array}{c} 10\lambda_1 + \lambda_2 = \frac{33}{8} < 6 \Rightarrow x_1 = 0 \\ -2\lambda_1 - \lambda_2 = -\frac{9}{8} < 1 \Rightarrow x_2 = 0 \\ 5\lambda_1 + 3\lambda_2 = 3 = 3 \Rightarrow x_3 \geq 0. \\ J = \{3\}. \end{array}$$

Il Problema Primale Ristretto (PPR) diventa, quindi

$$\min y_1 + y_2$$

$$s.t.$$

$$\begin{array}{cccc} 5x_3 & +y_1 & & = & 15 \\ 3x_3 & & +y_2 & = & 6 \end{array}$$

$$y_j \geq 0$$

Risolviamo il nuovo PPR. Si ricordi che, nel risolvere il PPR ad un generico passo k del simplesso primale-duale, la soluzione ottima del PPR al passo $k-1$ è soluzione ammissibile del PPR al passo k.

$$x_B = \{y_1, y_2\}$$

$$B^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$D = \begin{bmatrix} 5 \\ 3 \end{bmatrix}$$

$$c_B^T = [1, 1]$$

$$c_D^T = [0] \, .$$

Calcoliamo i moltiplicatori ed i costi ridotti:

$$\mu^T = c_B^T B^{-1} = [1, 1] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = [1, 1]$$

$$r_D^T = c_D^T - \mu^T D = [0] - [1, 1] \begin{bmatrix} 5 \\ 3 \end{bmatrix} = -8$$

Il costo ridotto della variabile x_3 è negativo, per cui la facciamo entrare in base. Calcoliamo la variabile che deve uscire dalla base.

$$B^{-1} b = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 15 \\ 6 \end{bmatrix} = \begin{bmatrix} 15 \\ 6 \end{bmatrix}$$

$$B^{-1} a_{x_3} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 3 \end{bmatrix} = \begin{bmatrix} 5 \\ 3 \end{bmatrix}$$

$$\varepsilon = \min \left\{ \frac{15}{5}, \frac{6}{3} \right\} = 2.$$

Esce dalla base la variabile y_2.
La nuova base è, quindi, $x_B = \{y_1, x_3\}$. Aggiorniamo la marice B^{-1} con il metodo di Gauss.

$$\left[\begin{array}{cc|c} 1 & 0 & 5 \\ 0 & 1 & \boxed{3} \end{array} \right]$$

$$\left[\begin{array}{cc|c} 1 & -\frac{5}{3} & 0 \\ 0 & \frac{1}{3} & 1 \end{array} \right]$$

$$x_B = \{y_1, x_3\}$$

$$B^{-1} = \begin{bmatrix} 1 & -\frac{5}{3} \\ 0 & \frac{1}{3} \end{bmatrix}$$

$$D = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$c_B^T = [1, 0]$$

$$c_D^T = [1]\,.$$

Calcoliamo i moltiplicatori ed i costi ridotti:

$$\mu^T = c_B^T B^{-1} = [1, 0] \begin{bmatrix} 1 & -\frac{5}{3} \\ 0 & \frac{1}{3} \end{bmatrix} = \left[1, -\frac{5}{3}\right]$$

$$r_D^T = c_D^T - \mu^T D = [1] - \left[1, -\frac{5}{3}\right] \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{8}{3}$$

$$x_B = B^{-1} b = \begin{bmatrix} 1 & -\frac{5}{3} \\ 0 & \frac{1}{3} \end{bmatrix} \begin{bmatrix} 15 \\ 6 \end{bmatrix} = \begin{bmatrix} 5 \\ 2 \end{bmatrix}.$$

Siamo giunti alla soluzione ottima del PPR, ma la funzione obiettivo è chiaramento non nulla ($y_1 = 5$), per cui non abbiamo ancora trovato la soluzione ottima del PP. Aggiorniamo la soluzione duale λ.

$$c^T - \lambda^T A = [6, 1, 3] - \left[\frac{3}{8}, \frac{3}{8}\right] \begin{bmatrix} 10 & -2 & 5 \\ 1 & -1 & 3 \end{bmatrix} = \left[\frac{15}{8}, \frac{17}{8}, 0\right]$$

$$\mu^T A = \left[1, -\frac{5}{3}\right] \begin{bmatrix} 10 & -2 & 5 \\ 1 & -1 & 3 \end{bmatrix} = \left[\frac{25}{3}, -\frac{1}{3}, 0\right]$$

$$\varepsilon = \min_{\mu^T a_j > 0} \left\{ \frac{c_j^T - \lambda^T a_j}{\mu^T a_j} \right\} = \left\{ \frac{\frac{15}{8}}{\frac{25}{3}} \right\} = \frac{9}{40}.$$

$$\lambda = \lambda + \varepsilon\mu = \begin{bmatrix} \frac{3}{8} \\ \frac{3}{8} \end{bmatrix} + \frac{9}{40} \begin{bmatrix} 1 \\ -\frac{5}{3} \end{bmatrix} = \begin{bmatrix} \frac{3}{5} \\ 0 \end{bmatrix}.$$

La nuova soluzione duale è $\lambda^T = \left[\frac{3}{5}, 0\right]$. Controlliamo il valore dei vincoli del PD.

$$10\lambda_1 + \lambda_2 = 6 = 6 \Rightarrow x_1 \geq 0$$

$$-2\lambda_1 - \lambda_2 = -\tfrac{6}{5} < 1 \Rightarrow x_2 = 0$$
$$5\lambda_1 + 3\lambda_2 = 3 = 3 \Rightarrow x_3 \geq 0.$$
$$J = \{1, 3\}.$$

Il Problema Primale Ristretto (PPR) diventa, quindi

$$\min y_1 + y_2$$

$$s.t.$$

$$\begin{array}{cccccc} 10x_1 & +5x_3 & +y_1 & & = & 15 \\ x_1 & +3x_3 & & +y_2 & = & 6 \end{array}$$

$$x_i \geq 0, y_j \geq 0$$

La base iniziale è la base ottima del PPR al passo precedente, $x_B = \{y_1, x_3\}$.

$$x_B = \{y_1, x_3\}$$

$$B^{-1} = \begin{bmatrix} 1 & -\frac{5}{3} \\ 0 & \frac{1}{3} \end{bmatrix}$$

$$D = \begin{bmatrix} 10 & 0 \\ 1 & 1 \end{bmatrix}$$

$$c_B^T = [1, 0]$$

$$c_D^T = [0, 1] .$$

Calcoliamo i moltiplicatori ed i costi ridotti:

$$\mu^T = c_B^T B^{-1} = [1, 0] \begin{bmatrix} 1 & -\frac{5}{3} \\ 0 & \frac{1}{3} \end{bmatrix} = \left[1, -\frac{5}{3}\right]$$

$$r_D^T = c_D^T - \mu^T D = [0, 1] - \left[1, -\frac{5}{3}\right] \begin{bmatrix} 10 & 0 \\ 1 & 1 \end{bmatrix} = \left[-\frac{25}{3}, \frac{8}{3}\right]$$

$$x_B = B^{-1} b = \begin{bmatrix} 1 & -\frac{5}{3} \\ 0 & \frac{1}{3} \end{bmatrix} \begin{bmatrix} 15 \\ 6 \end{bmatrix} = \begin{bmatrix} 5 \\ 2 \end{bmatrix} .$$

Il costo ridotto della variabile x_1 è negativo, per cui la facciamo entrare in base. Calcoliamo la variabile che deve uscire dalla base.

$$B^{-1}b = \begin{bmatrix} 1 & -\frac{5}{3} \\ 0 & \frac{1}{3} \end{bmatrix} \begin{bmatrix} 15 \\ 6 \end{bmatrix} = \begin{bmatrix} 5 \\ 2 \end{bmatrix}$$

$$B^{-1}a_{x_1} = \begin{bmatrix} 1 & -\frac{5}{3} \\ 0 & \frac{1}{3} \end{bmatrix} \begin{bmatrix} 10 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{25}{3} \\ \frac{1}{3} \end{bmatrix}$$

$$\varepsilon = \min \left\{ \frac{5}{\frac{25}{3}}, \frac{2}{\frac{1}{3}} \right\} = \frac{3}{5}.$$

Esce dalla base la variabile y_1.
La nuova base è, quindi, $x_B = \{x_1, x_3\}$. Aggiorniamo la marice B^{-1} con il metodo di Gauss.

$$\left[\begin{array}{cc|c} 1 & -\frac{5}{3} & \boxed{\frac{25}{3}} \\ 0 & \frac{1}{3} & \frac{1}{3} \end{array} \right]$$

$$\left[\begin{array}{cc|c} \frac{3}{25} & -\frac{1}{5} & 1 \\ -\frac{1}{25} & \frac{2}{5} & 0 \end{array} \right]$$

$$x_B = \{x_1, x_3\}$$

$$B^{-1} = \begin{bmatrix} \frac{3}{25} & -\frac{1}{5} \\ -\frac{1}{25} & \frac{2}{5} \end{bmatrix}$$

$$D = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$c_B^T = [0, 0]$$

$$c_D^T = [1, 1].$$

Calcoliamo i moltiplicatori ed i costi ridotti:

$$\mu^T = c_B^T B^{-1} = [0, 0] \begin{bmatrix} \frac{3}{25} & -\frac{1}{5} \\ -\frac{1}{25} & \frac{2}{5} \end{bmatrix} = [0, 0]$$

$$r_D^T = c_D^T - \mu^T D = [1, 1] - [0, 0] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = [1, 1]$$

$$x_B = B^{-1}b = \begin{bmatrix} \frac{3}{25} & -\frac{1}{5} \\ -\frac{1}{25} & \frac{2}{5} \end{bmatrix} \begin{bmatrix} 15 \\ 6 \end{bmatrix} = \begin{bmatrix} \frac{3}{5} \\ \frac{9}{5} \end{bmatrix}.$$

La soluzione ottima del PPR è soluzione ammissibile del PP (tutte le var. artificiali sono uscite dalla base), per cui è anche soluzione ottima del PP.
$x_1 = 1, x_3 = \frac{9}{5} \quad \Rightarrow z = 9.$

ESEMPIO 2.19. Risolvere il seguente problema mediante l'algoritmo del simplesso primale-duale:

$$\min 7x_1 + 2x_2 - 5x_3 - x_4$$

$$s.t.$$

$$4x_1 + 3x_2 + 2x_4 = 2$$

$$-5x_1 - 3x_2 + x_3 = 1$$

$$x_i \geq 0$$

considerando come soluzione duale di partenza $\lambda = \begin{bmatrix} -\frac{9}{2} \\ -5 \end{bmatrix}$.

Soluzione
Il Problema Duale (PD) è

$$\max 2\lambda_1 + \lambda_2$$

$$s.t.$$

$$4\lambda_1 - 5\lambda_2 \leq 7$$

$$3\lambda_1 - 3\lambda_2 \leq 2$$

$$\lambda_2 \leq -5$$

$$2\lambda_1 \leq -1$$

$$\lambda_i \text{ libere.}$$

Le matrici del problema sono

$$A = \begin{bmatrix} 4 & 3 & 0 & 2 \\ -5 & -3 & 1 & 0 \end{bmatrix}$$

$$b = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$c^T = [7, 2, -5, -1] .$$

$$\lambda = \begin{bmatrix} -\frac{9}{2} \\ -5 \end{bmatrix}$$

La soluzione duale è $\lambda^T = \left[-\frac{9}{2}, -5\right]$. Controlliamo il valore dei vincoli del PD.

$$\begin{gathered} 4\lambda_1 - 5\lambda_2 = 7 = 7 \Rightarrow x_1 \geq 0 \\ 3\lambda_1 - 3\lambda_2 = \frac{3}{2} < 2 \Rightarrow x_2 = 0 \\ \lambda_2 = -5 = -5 \Rightarrow x_3 \geq 0 \\ 2\lambda_1 = -9 < -1 \Rightarrow x_4 = 0 \\ J = \{1{,}3\}. \end{gathered}$$

Il Problema Primale Ristretto (PPR) diventa, quindi

$$\min y_1 + y_2$$

$$s.t.$$

$$\begin{array}{rrrrcl} 4x_1 & & +y_1 & & = & 2 \\ -5x_1 & +x_3 & & +y_2 & = & 1 \end{array}$$

$$x_i \geq 0, y_j \geq 0$$

Risolviamo il PPR.

$$x_B = \{y_1, y_2\}$$

$$B^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$D = \begin{bmatrix} 4 & 0 \\ -5 & 1 \end{bmatrix}$$

$$c_B^T = [1, 1]$$

$$c_D^T = [0, 0] .$$

Calcoliamo i moltiplicatori ed i costi ridotti:

$$\mu^T = c_B^T B^{-1} = [1,1]\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = [1,1]$$

$$r_D^T = c_D^T - \mu^T D = [0,0] - [1,1]\begin{bmatrix} 4 & 0 \\ -5 & 1 \end{bmatrix} = [1,-1]$$

Il costo ridotto della variabile x_3 è negativo, per cui la facciamo entrare in base. Calcoliamo la variabile che deve uscire dalla base.

$$B^{-1}b = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$B^{-1}a_{x_3} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\varepsilon = \min\left\{\frac{1}{1}\right\} = 1.$$

Esce dalla base la variabile y_2.
La nuova base è, quindi, $x_B = \{y_1, x_3\}$. Aggiorniamo la marice B^{-1} con il metodo di Gauss.

$$\left[\begin{array}{cc|c} 1 & 0 & 0 \\ 0 & 1 & \boxed{1} \end{array}\right]$$

$$\left[\begin{array}{cc|c} 1 & 0 & 0 \\ 0 & 1 & 1 \end{array}\right]$$

$$x_B = \{y_1, x_3\}$$

$$B^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$D = \begin{bmatrix} 4 & 0 \\ -5 & 1 \end{bmatrix}$$

$$c_B^T = [1,0]$$

$$c_D^T = [0,1]\,.$$

Calcoliamo i moltiplicatori ed i costi ridotti:

$$\mu^T = c_B^T B^{-1} = [1,0] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = [1,0]$$

$$r_D^T = c_D^T - \mu^T D = [0,1] - [1,0] \begin{bmatrix} 4 & 0 \\ -5 & 1 \end{bmatrix} = [-4,1]$$

$$x_B = B^{-1}b = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}.$$

Il costo ridotto della variabile x_1 è negativo, per cui la facciamo entrare in base. Calcoliamo la variabile che deve uscire dalla base.

$$B^{-1}b = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$B^{-1}a_{x_3} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ -5 \end{bmatrix} = \begin{bmatrix} 4 \\ -5 \end{bmatrix}$$

$$\varepsilon = \min \left\{ \frac{2}{4} \right\} = \frac{1}{2}.$$

Esce dalla base la variabile y_1.
La nuova base è, quindi, $x_B = \{x_1, x_3\}$. Aggiorniamo la marice B^{-1} con il metodo di Gauss.

$$\left[\begin{array}{cc|c} 1 & 0 & \boxed{4} \\ 0 & 1 & -5 \end{array} \right]$$

$$\left[\begin{array}{cc|c} \frac{1}{4} & 0 & 1 \\ \frac{5}{4} & 1 & 0 \end{array} \right]$$

$$x_B = \{x_2, x_3\}$$

$$B^{-1} = \begin{bmatrix} \frac{1}{4} & 0 \\ \frac{5}{4} & 1 \end{bmatrix}$$

$$D = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$c_B^T = [0, 0]$$

$$c_D^T = [1, 1]\,.$$

Calcoliamo i moltiplicatori ed i costi ridotti:

$$\mu^T = c_B^T B^{-1} = [0, 0] \begin{bmatrix} \frac{1}{4} & 0 \\ \frac{5}{4} & 1 \end{bmatrix} = [0, 0]$$

$$r_D^T = c_D^T - \mu^T D = [1, 1] - [0, 0] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = [1, 1]$$

$$x_B = B^{-1} b = \begin{bmatrix} \frac{1}{4} & 0 \\ \frac{5}{4} & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ \frac{7}{2} \end{bmatrix}.$$

La soluzione ottima del PPR è soluzione ammissibile del PP, per cui è anche soluzione ottima del PP.
$x_2 = \frac{1}{2}$,$x_3 = \frac{7}{2} \Rightarrow z = -\frac{33}{2}$.

ESEMPIO 2.20. Risolvere il seguente problema mediante l'algoritmo del simplesso primale-duale:

$$\min 2x_1 + x_2 + 4x_3$$

$$s.t.$$

$$x_1 + x_2 + 2x_3 = 3$$

$$2x_1 + x_2 + 3x_3 = 5$$

$$x_i \geq 0$$

considerando come soluzione duale di partenza $\lambda = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}$.

Soluzione

Il Problema Duale (PD) è

$$\max 3\lambda_1 + 5\lambda_2$$

$$s.t.$$

$$\lambda_1 + 2\lambda_2 \leq 2$$

$$\lambda_1 + \lambda_2 \leq 1$$

$$2\lambda_1 + 3\lambda_2 \leq 4$$

$$\lambda_i \quad libere$$

Le matrici del problema sono

$$A = \begin{bmatrix} 1 & 1 & 2 \\ 2 & 1 & 3 \end{bmatrix}$$

$$b = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$$

$$c^T = [2, 1, 4] \,.$$

$$\lambda = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}$$

La soluzione duale è $\lambda^T = \left[\frac{1}{2}, \frac{1}{2}\right]$. Controlliamo il valore dei vincoli del PD.

$$\begin{array}{c} \lambda_1 + 2\lambda_2 = \frac{3}{2} < 2 \Rightarrow x_1 = 0 \\ \lambda_1 + \lambda_2 = 1 = 1 \Rightarrow x_2 \geq 0 \\ 2\lambda_1 + 3\lambda_2 = \frac{5}{2} < 4 \Rightarrow x_3 = 0 \\ J = \{2\}. \end{array}$$

Il Problema Primale Ristretto (PPR) diventa, quindi

$$\min y_1 + y_2$$

$$s.t.$$

$$\begin{array}{lllll} x_2 & +y_1 & & = & 3 \\ x_2 & & +y_2 & = & 5 \end{array}$$

$$x_i \geq 0, y_j \geq 0$$

Risolviamo il PPR.

$$x_B = \{y_1, y_2\}$$

$$B^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$D = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$c_B^T = [1, 1]$$

$$c_D^T = [0] \, .$$

Calcoliamo i moltiplicatori ed i costi ridotti:

$$\mu^T = c_B^T B^{-1} = [1,1] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = [1,1]$$

$$r_D^T = c_D^T - \mu^T D = [0] - [1,1] \begin{bmatrix} 1 \\ 1 \end{bmatrix} = [-2]$$

$$x_B = B^{-1} b = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 5 \end{bmatrix} = \begin{bmatrix} 3 \\ 5 \end{bmatrix} .$$

Il costo ridotto della variabile x_2 è negativo, per cui la facciamo entrare in base. Calcoliamo la variabile che deve uscire dalla base.

$$B^{-1} b = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 5 \end{bmatrix} = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$$

$$B^{-1}a_{x_2} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\varepsilon = \min\{3, 5\} = 3.$$

Esce dalla base la variabile y_1.
La nuova base è, quindi, $x_B = \{x_2, y_2\}$. Aggiorniamo la matrice B^{-1} con il metodo di Gauss.

$$\left[\begin{array}{cc|c} 1 & 0 & \boxed{1} \\ 0 & 1 & 1 \end{array}\right]$$

$$\left[\begin{array}{cc|c} 1 & 0 & 1 \\ -1 & 1 & 0 \end{array}\right]$$

$$x_B = \{x_2, y_2\}$$

$$B^{-1} = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}$$

$$D = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$c_B^T = [0, 1]$$

$$c_D^T = [1].$$

Calcoliamo i moltiplicatori ed i costi ridotti:

$$\mu^T = c_B^T B^{-1} = [0, 1] \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} = [-1, 1]$$

$$r_D^T = c_D^T - \mu^T D = [1] - [-1, 1] \begin{bmatrix} 1 \\ 0 \end{bmatrix} = [2]$$

$$x_B = B^{-1}b = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 5 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}.$$

Siamo giunti alla soluzione ottima del PPR, ma la funzione obiettivo è chiaramento non nulla ($y_2 = 2$), per cui non abbiamo ancora trovato la soluzione ottima del PP.

Aggiorniamo la soluzione duale λ.

$$c^T - \lambda^T A = [2,1,4] - \left[\frac{1}{2},\frac{1}{2}\right] \left[\begin{array}{ccc} 1 & 1 & 2 \\ 2 & 1 & 3 \end{array}\right] = \left[\frac{1}{2},0,\frac{3}{2}\right]$$

$$\mu^T A = [-1,1] \left[\begin{array}{ccc} 1 & 1 & 2 \\ 2 & 1 & 3 \end{array}\right] = [1,0,1]$$

$$\varepsilon = \min_{\mu^T a_j > 0} \left\{ \frac{c_j^T - \lambda^T a_j}{\mu^T a_j} \right\} = \left\{ \frac{1}{2}, \frac{3}{2} \right\} = \frac{1}{2}.$$

$$\lambda = \lambda + \varepsilon\mu = \left[\begin{array}{c} \frac{1}{2} \\ \frac{1}{2} \end{array}\right] + \frac{1}{2} \left[\begin{array}{c} -1 \\ 1 \end{array}\right] = \left[\begin{array}{c} 0 \\ 1 \end{array}\right].$$

La soluzione duale è $\lambda^T = [0,1]$. Controlliamo il valore dei vincoli del PD.

$$\begin{array}{c} \lambda_1 + 2\lambda_2 = 2 = 2 \Rightarrow x_1 \geq 0 \\ \lambda_1 + \lambda_2 = 1 = 1 \Rightarrow x_2 \geq 0 \\ 2\lambda_1 + 3\lambda_2 = 4 < 4 \Rightarrow x_3 = 0 \\ J = \{1,2\}. \end{array}$$

Il Problema Primale Ristretto (PPR) diventa, quindi

$$\min y_1 + y_2$$

$$s.t.$$

$$\begin{array}{cccccc} x_1 & +x_2 & +y_1 & & = & 3 \\ 2x_1 & +x_2 & & +y_2 & = & 5 \end{array}$$

$$x_i \geq 0, y_j \geq 0$$

Risolviamo il PPR, prendendo come SAB la soluzione ottima del PPR al passo 1.

$$x_B = \{x_2, y_2\}$$

$$B^{-1} = \left[\begin{array}{cc} 1 & 0 \\ -1 & 1 \end{array}\right]$$

$$D = \left[\begin{array}{cc} 1 & 1 \\ 2 & 0 \end{array}\right]$$

$$c_B^T = [0,1]$$

$$c_D^T = [0,1].$$

Calcoliamo i moltiplicatori ed i costi ridotti:

$$\mu^T = c_B^T B^{-1} = [0,1] \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} = [-1,1]$$

$$r_D^T = c_D^T - \mu^T D = [0,1] - [-1,1] \begin{bmatrix} 1 & 1 \\ 2 & 0 \end{bmatrix} = [-1,2]$$

$$x_B = B^{-1} b = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 5 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}.$$

Il costo ridotto della variabile x_1 è negativo, per cui la facciamo entrare in base. Calcoliamo la variabile che deve uscire dalla base.

$$B^{-1} b = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 5 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

$$B^{-1} a_{x_1} = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\varepsilon = \min\{3,2\} = 2.$$

Esce dalla base la variabile y_2.
La nuova base è, quindi, $x_B = \{x_2, x_1\}$. Aggiorniamo la marice B^{-1} con il metodo di Gauss.

$$\left[\begin{array}{cc|c} 1 & 0 & 1 \\ -1 & 1 & \boxed{1} \end{array}\right]$$

$$\left[\begin{array}{cc|c} 2 & -1 & 0 \\ -1 & 1 & 1 \end{array}\right]$$

$$x_B = \{x_2, x_1\}$$

$$B^{-1} = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$c_B^T = [0,0]$$

$$c_D^T = [1,1].$$

Calcoliamo i moltiplicatori ed i costi ridotti:

$$\mu^T = c_B^T B^{-1} = [0,0]\begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix} = [0,0]$$

$$r_D^T = c_D^T - \mu^T D = [1,1] - [0,0]\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = [1,1]$$

$$x_B = B^{-1}b = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}\begin{bmatrix} 3 \\ 5 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

La soluzione ottima del PPR è soluzione ammissibile del PP, per cui è anche soluzione ottima del PP.
$x_2 = 2$,$x_1 = 1 \Rightarrow z_P = 4$.

2.5 Analisi di sensibilità

2.5.1 Richiami teorici

L'analisi di sensibilità è lo studio della stabilità della soluzione ottima di un problema di P.L. al variare dei dati del problema stesso. Dato il seguente modello di P.L.:

$$\begin{aligned} &\min \overline{c}^T\overline{x} \\ &A\overline{x} = \overline{b} \\ &\overline{x} \geq 0 \end{aligned}$$

le condizioni di ammissibilità di una generica soluzione di base x_B sono:

$$\overline{x}_B = B^{-1}\overline{b} \geq \overline{0}$$

le condizioni di ottimalità invece sono:

$$r_j = c_j - \underbrace{\overline{c}_B^T B^{-1}}_{\lambda^T}\overline{a}_j \geq 0, \;\; \forall j \notin B.$$

Variazioni di b (sensibilità sui termini noti)

Nel tableau relativo alla soluzione ottima del problema cambia solo il vettore relativo alla soluzione corrente x_B. Bisogna dunque solo verificare che la perturbazione non renda la soluzione ottima del problema non ammissibile. (le condizioni di ottimalità sono invece certamente rispettate).

- In caso di perturbazione fissa $\overline{\theta}$: $\overline{b}^* = \overline{b} + \overline{\theta}$, deve essere rispettata la seguente condizione:
$$B^{-1}\overline{b}^* = B^{-1}\overline{b} + B^{-1}\overline{\theta} = \overline{x}_B + B^{-1}\overline{\theta} \geq \overline{0}.$$

- In caso di perturbazione parametrica $\overline{b}^* = \overline{b} + k\overline{\theta}$:

$$B^{-1}\overline{b}^* = B^{-1}\overline{b} + kB^{-1}\overline{\theta} \geq \overline{0}.$$

- Per calcolare l'intervallo di stabilità di b_i si utilizza la condizione precedente (perturbazione parametrica) con:

$$k\overline{\theta} = \begin{bmatrix} 0 \\ \vdots \\ k \\ \vdots \\ 0 \end{bmatrix}$$

con k in i-ma posizione.

Variazioni di c (sensibilità sui coefficienti di costo)

In questo caso è necessario verificare se la base ottima del problema non perturbato rimane ottima in quello perturbato (le condizioni di ammissibilità sono invece certamente rispettate).

- perturbazione fissa: $\overline{c}^* = \overline{c} + \overline{\theta}$. Deve essere ancora verificata la condizione:

$$\overline{c}_D^{*T} - \overline{c}_B^{*T} B^{-1}D = (\overline{c}_D + \overline{\theta}_D)^T - (\overline{c}_B + \overline{\theta}_B)^T B^{-1}D = \overline{r}_D^T + \overline{\theta}_D^T - \overline{\theta}_B^T B^{-1}D \geq 0.$$

- perturbazione parametrica: $\overline{c}^* = \overline{c} + k\overline{\theta}$. Deve essere verificata la seguente condizione:

$$\overline{r}_D^{*T} = (\overline{c}_D + k\overline{\theta}_D)^T - (\overline{c}_B + k\overline{\theta}_B)^T B^{-1}D \geq \overline{0}$$

ossia

$$\overline{r}_D^T + k\overline{\theta}_D^T - k\overline{\theta}_B^T B^{-1}D \geq \overline{0}.$$

- intervallo di stabilità di un coefficiente di costo di una variabile in base:

$$k\theta = \begin{bmatrix} 0 \\ \vdots \\ k \\ \vdots \\ 0 \end{bmatrix}$$

con k in posizione i-esima per c_i e x_i in base. In questo caso si ha $\overline{\theta}_D^T = \overline{0}$ e quindi la condizione precedente diventa

$$\overline{r}_D^T - k\overline{\theta}_B^T B^{-1}D \geq \overline{0}.$$

- intervallo di stabilità di un coefficiente di costo di una variabile fuori base:

$$k\theta = \begin{bmatrix} 0 \\ \vdots \\ k \\ \vdots \\ 0 \end{bmatrix}$$

con k in posizione i-esima per c_i e x_i fuori base. Si ha $\overline{\theta}_B^T = \overline{0}$ e quindi la condizione di ottimalità diventa

$$\overline{r}_D^T + k\overline{\theta}_D^T \geq \overline{0},$$

ovvero $k \geq -r_i$. Il costo ridotto $r_i \geq 0$ di una variabile fuori base può, quindi, essere interpretato come il massimo decremento del costo c_i per cui la base B rimane ottima.

2.5.2 Esercizi risolti

ESEMPIO 2.21. Si consideri il seguente problema

$$\begin{aligned}
&\min -x_1 - x_2 - x_3 \\
&x_1 - 2x_2 + x_3 + x_4 = 5 \\
&x_2 + x_3 \leq 3 \\
&3x_2 - 2x_4 \leq 6 \\
&x_i \geq 0
\end{aligned}$$

1. Si controlli l'ottimalità della soluzione x_1, x_2, x_4.
2. Data la seguente variazione dei termini noti,

$$\theta = \begin{bmatrix} -\frac{1}{2} \\ -1 \\ -1 \end{bmatrix}$$

si calcoli l'intervallo di stabilità della soluzione per i seguenti tipi di variazione:

- fissa
- parametrica
- intervallo di stabilità di b_3

3. Data la seguente variazione dei costi:

$$\theta = \begin{bmatrix} 0 \\ 1 \\ -1 \\ -1 \\ 0 \\ -1 \end{bmatrix}$$

si calcoli l'intervallo di stabilità della soluzione per i seguenti tipi di variazione:

- fissa
- parametrica
- intervallo di stabilità di c_1
- intervallo di stabilità di c_3

Soluzione

1. Portando il problema in Forma Standard, si ottiene

$$\begin{array}{l} \min -x_1 - x_2 - x_3 \\ x_1 - 2x_2 + x_3 + x_4 = 5 \\ x_2 + x_3 + x_5 = 3 \\ 3x_2 - 2x_4 - x_6 = 6 \\ x_i \geq 0 \end{array}$$

$$\begin{array}{cccccc} & & x_1 & x_2 & x_4 & \\ c_B = & [& -1 & -1 & 0 &] \end{array}$$

$$\begin{array}{cccccc} & & x_3 & x_5 & x_6 & \\ c_D = & [& -1 & 0 & 0 &] \end{array}$$

$$B = \begin{bmatrix} 1 & -2 & 1 \\ 0 & 1 & 0 \\ 0 & 3 & -2 \end{bmatrix}$$

$$D = \begin{array}{ccc} x_3 & x_5 & x_6 \end{array} \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Calcoliamo ora la matrice B^{-1}:

$$B^{-1} = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} \\ 0 & 1 & 0 \\ 0 & \frac{3}{2} & -\frac{1}{2} \end{bmatrix}$$

$$b = \begin{bmatrix} 5 \\ 3 \\ 6 \end{bmatrix}$$

$$x_B = B^{-1}b = \begin{bmatrix} 5 + \frac{3}{2} + 3 \\ 3 \\ \frac{9}{2} - \frac{6}{2} \end{bmatrix} = \begin{bmatrix} \frac{19}{2} \\ 3 \\ \frac{3}{2} \end{bmatrix}$$

Termini noti:

- Parte fissa:

$$\theta = \begin{bmatrix} -\frac{1}{2} \\ -1 \\ -1 \end{bmatrix}$$

$$x_B = B^{-1}(b + \theta) = x_B + B^{-1}\theta$$

$$B^{-1}\theta = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} \\ 0 & 1 & 0 \\ 0 & \frac{3}{2} & -\frac{1}{2} \end{bmatrix} = \begin{bmatrix} -\frac{1}{2} \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} -\frac{3}{2} \\ -1 \\ -1 \end{bmatrix}$$

$$x_B = \begin{bmatrix} \frac{19}{2} \\ 3 \\ \frac{3}{2} \end{bmatrix} + \begin{bmatrix} -\frac{3}{2} \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} 8 \\ 2 \\ \frac{1}{2} \end{bmatrix} \geq 0$$

  quindi ammissibile

- parte parametrica (stessa θ)

$$x_B = B^{-1}(b + k\theta) = x_B + kB^{-1}\theta = \begin{bmatrix} \frac{19}{2} \\ 3 \\ \frac{3}{2} \end{bmatrix} + \begin{bmatrix} -\frac{3}{2}k \\ -k \\ -k \end{bmatrix} =$$

$$= \begin{bmatrix} \frac{19}{2} + \frac{3}{2}k \\ 3 - k \\ \frac{3}{2} - k \end{bmatrix} \geq 0 \Rightarrow \begin{array}{l} k \geq -\frac{19}{3} \\ k \leq 3 \\ k \leq \frac{3}{2} \end{array}$$

$$k \in \left[-\frac{19}{3}, \frac{3}{2}\right]$$

- intervallo di stabilità di b3

$$x_B + B^{-1} \begin{bmatrix} 0 \\ 0 \\ k \end{bmatrix} \geq 0$$

$$x_B + \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} \\ 0 & 1 & 0 \\ 0 & \frac{3}{2} & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ k \end{bmatrix} \geq 0$$

$$\begin{bmatrix} \frac{19}{2}+\frac{1}{2}k \\ 3+0 \\ \frac{3}{2}-\frac{1}{2}k \end{bmatrix} \geq 0 \Rightarrow \begin{matrix} k \geq -19 \\ \\ k \leq 3 \end{matrix}$$

$$k \in [-19,\ 3]$$

$$b \in [-13,\ 9]$$

Costo:

- parte fissa:

$$\theta = \begin{array}{c} \\ [\end{array} \begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ 0 & 1 & -1 & -1 & 0 & -1 \end{array} \begin{array}{c} \\ ] \end{array}$$

$$c_B = \begin{bmatrix} -1 & -1 & 0 \end{bmatrix}$$

$$c_D = \begin{bmatrix} -1 & 0 & 0 \end{bmatrix}$$

$$\theta_B = \begin{bmatrix} 0 & 1 & -1 \end{bmatrix}$$

$$\theta_D = \begin{bmatrix} -1 & 0 & -1 \end{bmatrix}$$

$$r_3^* = c_3^* - \lambda^T b_3$$

$$\lambda^T = c_B^{*T} B^{-1} = \left(c_B^T + \theta^T\right) B^{-1} =$$

$$\begin{bmatrix} -1 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} \\ 0 & 1 & 0 \\ 0 & \frac{3}{2} & -\frac{1}{2} \end{bmatrix} = \begin{bmatrix} -1 & -2 & 0 \end{bmatrix}$$

$$r_3^* = (C_3 - 1) - \begin{bmatrix} -1 & -2 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = -2 - [-3] = 1 \geq 0$$

$$r_5 = (0 - 0) - \begin{bmatrix} -1 & -2 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = 0 - [-2] = 2 \geq 0$$

$$r_6 = (0 - 1) - \begin{bmatrix} -1 & -2 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = -1 - 0 = -1 \leq 0 \Rightarrow \textbf{non ottima}$$

- parte parametrica:

$$r_D^* = c_D^* - c_B B^{-1} D = (c_D + k\theta_D)^T - \underbrace{(c_B + k\theta_B)^T B^{-1}}_{\lambda^T} D$$

$$\lambda^{*T} = \begin{bmatrix} -1 & -1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & k & -k \end{bmatrix} \left[B^{-1}\right] =$$

$$\begin{bmatrix} -1 & k-1 & -k \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} \\ 0 & 1 & 0 \\ 0 & \frac{3}{2} & -\frac{1}{2} \end{bmatrix} = \begin{bmatrix} -1 & -\frac{1}{2} + k - 1 - \frac{3}{2}k & -\frac{1}{2} + \frac{k}{2} \end{bmatrix} =$$

$$\begin{bmatrix} -1 & -\frac{1}{2}k-\frac{3}{2} & \frac{k}{2}-\frac{1}{2} \end{bmatrix}$$

$$r_3 = (-k) - \begin{bmatrix} -1 & -\frac{1}{2}k-\frac{3}{2} & \frac{k}{2}-\frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = -2 - \begin{bmatrix} -1 & -\frac{1}{2}k & -\frac{3}{2} \end{bmatrix} =$$

$$-2+1+\frac{3}{2}+\frac{1}{2}k = \frac{1}{2}+\frac{1}{2}k+1-k \geq 0 \rightarrow k \geq -3$$

$$r_5 = 0 - \begin{bmatrix} -1 & -\frac{1}{2}k-\frac{3}{2} & \frac{1}{2}k-\frac{1}{2} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \frac{1}{2}k+\frac{3}{2} \geq 0 \rightarrow k \geq -3$$

$$r_6 = -k - \begin{bmatrix} -1 & -\frac{1}{2}k-\frac{3}{2} & \frac{1}{2}k-\frac{1}{2} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = -1-\frac{k}{2}+\frac{1}{2} \geq 0$$

$$-\frac{k}{2}-\frac{1}{2}-k \geq 0 \rightarrow k \leq 13 \Rightarrow k \in \left[-1, \frac{1}{2}\right]$$

- intervallo di stabilità c_1:

$$\lambda^* = C_B^T B^{-1} = \begin{bmatrix} -1+k & -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} \\ 0 & 1 & 0 \\ 0 & \frac{3}{2} & -\frac{1}{2} \end{bmatrix} =$$

$$= \begin{bmatrix} k-1 & \frac{k}{2}-\frac{1}{2}-1 & \frac{k}{2}-\frac{1}{2} \end{bmatrix} = \begin{bmatrix} k-1 & \frac{k}{2}-\frac{3}{2} & \frac{k}{2}-\frac{1}{2} \end{bmatrix}$$

$$r_3 = c_3 - \lambda^* a_3 = -1 - \begin{bmatrix} k-1 & \frac{k}{2}-\frac{3}{2} & \frac{k}{2}-\frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} =$$

$$= -1-k+1-\frac{k}{2}+\frac{3}{2} = \frac{3}{2}-\frac{3}{2}k \geq 0 \rightarrow k \leq 1$$

$$r_5 = c_5 - \lambda^* a_5 = 0 - \begin{bmatrix} k-1 & \frac{k}{2}-\frac{3}{2} & \frac{k}{2}-\frac{1}{2} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} =$$

$$= 0-\frac{k}{2}+\frac{3}{2} \geq 0 \rightarrow k \leq 3$$

$$r_6 = c_6 - \lambda^* a_6 = 0 - \begin{bmatrix} k-1 & \frac{k}{2}-\frac{3}{2} & \frac{k}{2}-\frac{1}{2} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} =$$

$$= 0 - \frac{k}{2} + \frac{1}{2} \geq 0 \rightarrow k \leq 1$$

$$k \in [-\infty, +1]$$

- intervallo di stabilità c_3:

$$\lambda = c_B^* B^{-1} = \begin{bmatrix} -1 & -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} \\ 0 & 1 & 0 \\ 0 & \frac{3}{2} & -\frac{1}{2} \end{bmatrix} = \begin{bmatrix} -1 & -\frac{3}{2} & -\frac{1}{2} \end{bmatrix}$$

$$r_3^* = c_3 - \lambda a_3 = k + c_3 - \lambda a_3 = k + r_3 = k + \left(-1 - \begin{bmatrix} -1 & -\frac{3}{2} & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \right) =$$

$$k - 1 + 1 + \frac{3}{2} = k + \frac{3}{2} \geq \rightarrow k \geq -\frac{3}{2}$$

$$k \in \left[-\frac{3}{2}, +\infty \right]$$

2.6 Esercizi proposti

1. Scrivere in forma standard il seguente modello di programmazione lineare:

$$\begin{array}{rl} \max & 4x_1 - 3x_2 + 5x_3 + 2x_4 \\ \text{s.t.} & -x_1 + 2x_2 + x_3 + 3x_4 \geq 5 \\ & 2x_1 - x_2 + x_3 - 6x_4 \leq -1 \\ & x_1 \in \mathbb{R}^-; x_2 \in \mathbb{R}^+; x_3, x_4 \in \mathbb{R} \end{array}$$

2. Risolvere il seguente problema di Programmazione Lineare:

$$\begin{array}{rl} \max & x_1 + x_2 + x_3 + x_4 \\ \text{s.t.} & x_1 + 3x_2 - x_3 + x_4 = 3 \\ & 3x_1 + x_2 + 2x_3 - x_4 = 6 \\ & x_i \in \mathbb{R}^+ \ \ \forall i. \end{array}$$

3. Si risolva il seguente problema di programmazione lineare:

$$\begin{array}{rl} \max & x_1 + 3x_2 \\ \text{s.t.} & x_1 + x_2 \leq 6 \\ & -x_1 + 2x_2 \leq 4 \\ & x_i \in \mathbb{R}^+ \ \ \forall i. \end{array}$$

4. Si risolva il seguente problema di programmazione lineare:

$$\begin{array}{rl} \min & 4x_1 - 5x_2 + 3x_3 - 2x_4 \\ \text{s.t.} & x_1 + 2x_2 + x_3 + 2x_4 = 4 \\ & x_1 + x_2 - 2x_3 + 2x_4 = 6 \\ & x_i \in \mathbb{R}^+ \ \ \forall i. \end{array}$$

5. Risolvere il seguente programma lineare.

$$\begin{array}{rl} \max & 2x_1 - 3x_2 + 2x_3 \\ \text{s.t.} & 2x_1 + x_2 + x_3 \le 6 \\ & x_1 + 3x_2 + 2x_3 \le 7 \\ & x_i \in \mathbb{R}^+ \ \ \forall i. \end{array}$$

6. Risolvere il seguente problema di programmazione lineare.

$$\begin{array}{rl} \max & x_1 + 2x_2 + x_3 \\ \text{s.t.} & x_1 + x_2 + 3x_3 \le 6 \\ & 2x_1 + 3x_2 + x_3 \le 15 \\ & x_i \in \mathbb{R}^+ \ \ \forall i. \end{array}$$

7. Calcolare la soluzione ottima del seguente problema di programmazione lineare.

$$\begin{array}{rl} \min & x_1 - 2x_2 + x_3 \\ \text{s.t.} & x_1 - x_2 + 3x_3 = 5 \\ & x_1 - 4x_2 + 8x_3 = 5 \\ & 2x_1 + x_2 + x_3 = 4 \\ & x_i \in \mathbb{R}^+ \ \ \forall i. \end{array}$$

   Ripetere l'esercizio eliminando il secondo vincolo.

8. Calcolare la soluzione ottima del seguente problema di programmazione lineare.

$$\begin{array}{rl} \min & x_1 + x_2 - 2x_3 \\ \text{s.t.} & 2x_1 + x_2 + x_3 = 2 \\ & 3x_1 + x_2 + 2x_3 = 5 \\ & x_1 + 2x_2 + x_3 = 4 \\ & x_i \in \mathbb{R}^+ \ \ \forall i. \end{array}$$

9. Calcolare la soluzione ottima del seguente problema di programmazione lineare.

$$\begin{array}{rl} \min & x_1 - x_2 + x_3 \\ \text{s.t.} & x_1 + x_3 = 2 \\ & 2x_1 + x_3 = 3 \\ & x_2 \ge 3 \\ & x_i \in \mathbb{R}^+ \ \ \forall i. \end{array}$$

10. Calcolare la soluzione ottima del seguente problema di programmazione lineare.

$$\begin{array}{rl} \min & 3x_1 + 4x_2 - x_3 \\ \text{s.t.} & x_1 + x_2 - 3x_3 \leq 2 \\ & -x_1 - 2x_2 + 2x_3 \leq 5 \\ & x_2 - 4x_3 \leq 2 \\ & x_i \in \mathbb{R}^+ \quad \forall i. \end{array}$$

11. Risolvere il seguente problema di Programmazione Lineare:

$$\begin{array}{rl} \min & 2x_1 + x_2 + 3x_3 + 2x_4 + 10x_5 \\ \text{s.t.} & x_1 + x_3 - x_4 + 2x_5 = 5 \\ & x_2 + 2x_3 + 2x_4 + x_5 = 9 \\ & x_i \in \mathbb{R}^+ \quad \forall i. \end{array}$$

12. Risolvere il seguente problema di programmazione lineare con l'algoritmo del simplesso:

$$\begin{array}{rl} \min & -3x_1 + 4x_2 - 2x_3 + 8x_4 \\ \text{s.t.} & +3x_1 - x_2 + x_3 - 3x_4 \leq 6 \\ & x_1 - 2x_2 + 2x_3 - 3x_4 \leq 6 \\ & x_i \in \mathbb{R}^+ \quad \forall i. \end{array}$$

13. Risolvere il seguente problema di programmazione lineare:

$$\begin{array}{rl} \max & -2x_1 + x_2 + 3x_3 \\ \text{s.t.} & x_1 + 2x_2 + x_3 \leq 2 \\ & 2x_1 + x_2 + x_3 \leq 3 \\ & x_i \in \mathbb{R}^+ \quad \forall i. \end{array}$$

14. Si risolva il seguente problema di programmazione lineare:

$$\begin{array}{rl} \min & 9x_1 - x_2 \\ \text{s.t.} & 3x_1 + x_2 \geq 9 \\ & -3x_1 + 2x_2 \leq 6 \\ & x_i \in \mathbb{R}^+ \quad \forall i. \end{array}$$

15. Si risolva il seguente problema di programmazione lineare:

$$\begin{array}{rl} \max & 2x_1 + 3x_2 + 5x_3 + 4x_4 \\ \text{s.t.} & x_1 + 2x_2 + 3x_3 + x_4 = 2 \\ & 2x_1 + x_2 + 2x_3 + 3x_4 = 3 \\ & x_i \in \mathbb{R}^+ \quad \forall i. \end{array}$$

Discutere ottimalità e stabilità della soluzione al variare del coefficiente c_1.

16. Si risolva partendo dalla base $[x_1, x_4]$ il seguente problema di programmazione lineare:

$$\begin{array}{rl} \max & 3x_1 + 2x_2 + x_3 + x_4 \\ \text{s.t.} & x_1 + 2x_2 + x_3 + 3x_4 = 2 \\ & 2x_1 + x_2 + 3x_3 + 2x_4 = 3 \\ & x_i \in \mathbb{R}^+ \ \ \forall i. \end{array}$$

17. Si risolva partendo dalla base $[x_1, x_2]$ il seguente problema di programmazione lineare:

$$\begin{array}{rl} \min & -2x_1 + 4x_2 + 2x_3 - 4x_4 \\ \text{s.t.} & x_1 + 4x_2 - x_3 - 2x_4 = 8 \\ & -2x_1 + 3x_2 + 2x_3 + x_4 = 6 \\ & x_i \in \mathbb{R}^+ \ \ \forall i. \end{array}$$

18. Sia dato il seguente problema di programmazione lineare:

$$\begin{array}{rl} \max & 3x_1 + 2x_2 + x_3 + x_4 + 2x_5 \\ \text{s.t.} & x_1 + 2x_2 + x_3 + 3x_4 + 2x_5 = 10 \\ & 2x_1 + x_2 + 3x_3 + 2x_4 + 3x_5 = 15 \\ & x_i \in \mathbb{R}^+ \ \ \forall i. \end{array}$$

Sapendo che all'ottimo la base è costituita dalle variabili $[x_2, x_1]$,

- verificare l'ottimalità della soluzione e calcolare il corrispondente valore della funzione obiettivo;
- calcolare l'intervallo di stabilità della base al variare del coefficiente di costo c_1.

19. Si risolva il seguente problema di programmazione lineare:

$$\begin{array}{rl} \min & 8x_1 + 8x_2 + 4x_3 + 6x_4 \\ \text{s.t.} & 3x_1 + 2x_2 + \frac{3}{2}x_3 + x_4 \geq 2 \\ & 8x_1 + 12x_2 + 8x_3 + 6x_4 \geq 8 \\ & x_i \in \mathbb{R}^+ \ \ \forall i. \end{array}$$

Quale è l'intervallo di stabilità della soluzione ottima del problema al variare del termine noto b_2?

20. Risolvere con il simplesso duale il seguente problema di Programmazione Lineare:

$$\begin{array}{rl} \min & 4x_1 + 3x_2 + x_3 + 2x_4 \\ \text{s.t.} & -x_1 + 2x_2 + x_3 - 3x_4 \geq 2 \\ & 2x_1 - x_2 + x_3 - 5x_4 \leq -3 \\ & x_i \in \mathbb{R}^+ \ \ \forall i. \end{array}$$

21. Si risolva utilizzando il simplesso duale il seguente problema di programmazione lineare:

$$\begin{array}{rl} \min & 4x_1 + 4x_2 + 2x_3 + 3x_4 \\ \text{s.t.} & 6x_1 + 4x_2 + 3x_3 + 2x_4 \geq 4 \\ & 4x_1 + 6x_2 + 4x_3 + 3x_4 \geq 4 \\ & x_i \in \mathbb{R}^+ \;\; \forall i. \end{array}$$

Quale è l'intervallo di stabilità della soluzione ottima del problema al punto 2 al variare del termine noto b_1?

22. Risolvere il seguente problema di Programmazione Lineare con l'algoritmo del simplesso:

$$\begin{array}{rl} \min & 3x_1 - 5x_2 - 4x_3 \\ \text{s.t.} & -x_1 + 4x_2 - 2x_3 \leq 4 \\ & x_1 + 3x_2 + 4x_3 \leq 6 \\ & x_i \in \mathbb{R}^+ \;\; \forall i. \end{array}$$

23. E' dato il seguente problema di Programmazione Lineare:

$$\begin{array}{rl} \min & 2x_1 + x_2 + 3x_3 + 2x_4 + 10x_5 \\ \text{s.t.} & x_1 + x_3 - x_4 + 2x_5 = 5 \\ & x_2 + 2x_3 + 2x_4 + x_5 = 9 \\ & x_i \in \mathbb{R}^+ \;\; \forall i. \end{array}$$

Sapendo che la base ottima è x_1, x_3, calcolare l'intervallo di stabilità della base ottima del problema al variare di

(a) b_2
(b) c_1
(c) c_4

24. Risolvere con l'algoritmo del simplesso primale-duale il seguente problema di programmazione lineare:

$$\begin{array}{rl} \min & 2x_1 + 3x_2 + 2x_3 + 5x_4 \\ \text{s.t.} & x_1 + 2x_2 + x_3 + 2x_4 = 3 \\ & x_1 + x_2 + 2x_3 + 4x_4 = 5 \\ & x_i \in \mathbb{R}^+ \;\; \forall i. \end{array}$$

25. Risolvere con l'algoritmo del simplesso primale-duale il seguente problema di programmazione lineare:

$$\begin{array}{rl} \min & 4x_1 + x_2 + x_3 \\ \text{s.t.} & 2x_1 + x_2 + 2x_3 = 4 \\ & 3x_1 + 3x_2 + x_3 = 3 \\ & x_i \in \mathbb{R}^+ \;\; \forall i. \end{array}$$

26. Risolvere con l'algoritmo del simplesso primale-duale il seguente problema di programmazione lineare:

$$\begin{array}{rl} \min & 2x_1 - x_2 \\ \text{s.t.} & 2x_1 - x_2 - x_3 = 3 \\ & x_1 - x_2 + x_3 = 2 \\ & x_i \in \mathbb{R}^+ \ \ \forall i. \end{array}$$

3

Algoritmi su grafi

3.1 Notazione

Nell'esposizione successiva si utilizzerà la seguente notazione:
$G(N, A)$: grafo definito dall'insieme dei nodi N e da quello degli archi A;
$n = |N|$: numero di nodi del grafo;
l_{ij}: lunghezza dell'arco (i, j);
$P(i)$: predecessore del nodo i nel cammino minimo da 1 a i;
Γ_i: insieme dei successori di i nel grafo;
Γ_i^{-1}: insieme dei predecessori di i nel grafo;
Γ_S: insieme dei successori dei nodi appartenenti all'insieme S.

3.2 Camino minimo (shortest path)

3.2.1 Richiami teorici

Algoritmo di Dijkstra

L'algoritmo di Dijkstra si applica ai grafi i cui archi abbiano lunghezza non negativa, in quanto non è in grado di riconoscere la presenza di circuiti negativi. Si suppone di voler calcolare lo shortest path tra il nodo 1 e gli altri nodi del grafo.

Function Dijkstra

$\Pi(1) = 0$,
$\Pi(i) = l_{ij}$, $\forall i \neq 1, i \in \Gamma_1$
$\Pi(i) = \infty$, $\forall i \neq 1, i \notin \Gamma_1$
$\overline{S} = \{2, \ldots, n\}$
while $\overline{S} \neq \{\emptyset\}$ **do**
  $j = argmin_{i \in \overline{S}}\{\Pi(i)\}$
  $\overline{S} = \overline{S} \setminus \{j\}$

for $i \in \Gamma_j$ **do**
$\Pi(i) = \min(\Pi(i), \Pi(j) + l_{ij})$
if $(\Pi(i) = \Pi(j) + l_{ij})$ **then**
$P(i) = j$
end if
end for
$\overline{S} = \overline{S} \setminus \{j\}$
end while

Algoritmo di Bellman-Ford

L'algoritmo di Bellman-Ford si applica ai grafi i cui archi abbiano lunghezza arbitraria ed è in grado di riconoscere la presenza di circuiti negativi. Si suppone di voler calcolare lo shortest path tra il nodo 1 e gli altri nodi del grafo e che l'algoritmo ritorni **true** se è riuscito a convergere, **false** altrimenti.

Function Bellman-Ford

optimal = **false**
$\Pi^0(1) = 0,\ \ \Pi^0(i) = \infty,\ \forall i \neq 1$
$P(i) = 1$
while $k \leq (N)$ **and** *optimal* = **false do**
$\Pi^k(i) = \min\left[\Pi^{k-1}(i), \min_{j \in \Gamma_i^{-1}}\left(\Pi^{k-1}(j) + l_{ji}\right)\right]$
if $(\Pi(i) = \Pi(j) + l_{ji})$ **then**
$P(i) = j$
end if
if $\Pi^k(i) = \Pi^{k-1}(i),\ \forall i$ **then**
optimal = **true**
end if
end while

return *optimal*

Algoritmo di Gondran-Minoux

Si usa per ricercare il cammino minimo tra il vertice 1 e gli altri vertici di un grafo senza cicli. Si suppone che il vertice 1 sia l'unico vertice senza predecessori.

Function Gondran-Minoux
$\Pi(1) = 0,\ S = \{1\},\ \overline{S} = \{2, \ldots, N\}$

while $\overline{S} \neq \{\emptyset\}$ **do**
  Trovare $j \in \overline{S} \,|\, \Gamma_j^{-1} \subseteq S$
  $\Pi(j) = min_{i \in \Gamma_j^{-1}} [\Pi(i) + l_{ij}]$
  $S = S \cup \{j\}$
  $\overline{S} = \overline{S} \setminus \{j\}$
end while

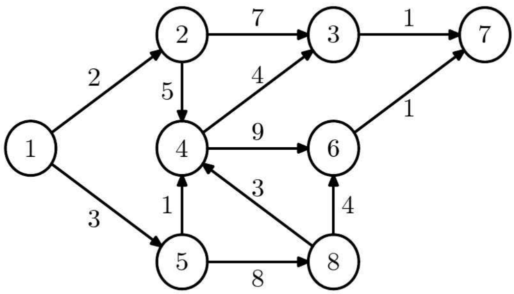
La seguente versione dell'algoritmo di Gondran-Minoux risolve il problema di dover cercare il nodo j. Per far ciò i nodi vengono ordinati attraverso la funzione di rank del grafo.

Function Gondran-Minoux con Grafo di Rank
r_i : rank del nodo i;
S_k : insieme dei nodi che hanno rank k;

$d_i = \left|\Gamma_i^{-1}\right|, \forall i \in N$
$k = 0$, $S_k = \{1\}, S = N \setminus \{1\}$
while $(S \neq \{\emptyset\})$ **do**
  $S_{k+1} = \{\emptyset\}$
  for all $(i \in S_k)$ **do**
    $r_i = k$
    for all $(j \in \Gamma_i)$ **do**
      $d_j = d_j - 1$
      if $(d_j = 0)$ **then**
        $S_{k+1} = S_{k+1} \cup \{j\}$
        $S = N \setminus \{j\}$
      end if
    end for
  end for
  $k = k + 1$
end while
$r = 0$
$S = N$
while $(S \neq \{\emptyset\})$ **do**
  for all $(j \in S \,|\, r_j = r)$ **do**
    $S = S \setminus \{j\}$
    $\Pi(j) = min_{i \in \Gamma_j^{-1}} [\Pi(i) + l_{ij}]$
  end for
  r=r+1
end while

3.2.2 Esercizi risolti

ESEMPIO 3.1. Risolvere tramite l'algoritmo di Dijkstra il seguente problema di cammino di costo minimo considerando come radice il nodo 1.



1. $\overline{S} = \{2, 3, 4, 5, 6, 7, 8\}$
   $\Pi(1) = 0$
   $\Pi(2) = 2, \Pi(5) = 3$
   $\Pi(3) = \Pi(4) = \Pi(6) = \Pi(7) = \Pi(8) = \infty$
   $P(2) = P(5) = 1.$

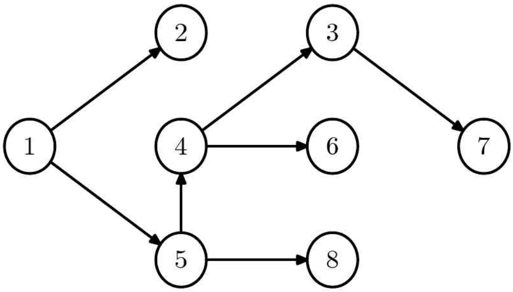
2. $j = 2$, $\overline{S} = \{3, 4, 5, 6, 7, 8\}$
   $\Gamma_2 \cap \overline{S} = \{3, 4\}$
   $\Pi(3) = \min(\infty, 9) = 9$
   $\Pi(4) = \min(\infty, 7) = 7$
   $P(3) = P(4) = 2.$

3. $j = 5$, $\overline{S} = \{3, 4, 6, 7, 8\}$
   $\Gamma_5 \cap \overline{S} = \{4, 8\}$
   $\Pi(4) = \min(7, 4) = 4$
   $\Pi(8) = \min(\infty, 11) = 11$
   $P(4) = P(8) = 5.$

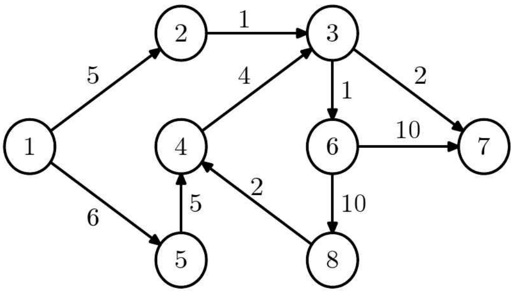
4. $j = 4$, $\overline{S} = \{3, 6, 7, 8\}$
   $\Gamma_4 \cap \overline{S} = \{3, 6\}$
   $\Pi(3) = \min(9, 8) = 8$
   $\Pi(6) = \min(\infty, 13) = 13$
   $P(3) = P(6) = 4.$

5. $j = 3$, $\overline{S} = \{6, 7, 8\}$
   $\Gamma_3 \cap \overline{S} = \{7\}$
   $\Pi(7) = \min(\infty, 9) = 9$
   $P(7) = 3$.
6. $j = 7$, $\overline{S} = \{6, 8\}$
   $\Gamma_7 \cap \overline{S} = \{\emptyset\}$
7. $j = 8$, $\overline{S} = \{6\}$
   $\Gamma_8 \cap \overline{S} = \{6\}$
   $\Pi(7) = \min(9, 14) = 13$.

I valori dei cammini minimi dei nodi sono:
$\Pi(2) = 2$, $\Pi(3) = 8$, $\Pi(4) = 4$, $\Pi(5) = 3$, $\Pi(6) = 13$, $\Pi(7) = 9$, $\Pi(8) = 11$,
mentre l'albero dei cammini minimi è il seguente:

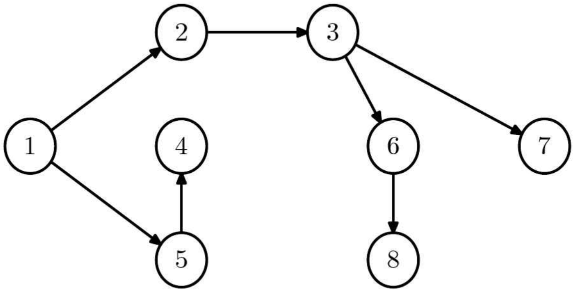


ESEMPIO 3.2. Risolvere tramite l'algoritmo di Dijkstra il seguente problema di cammino di costo minimo considerando come radice il nodo 1.

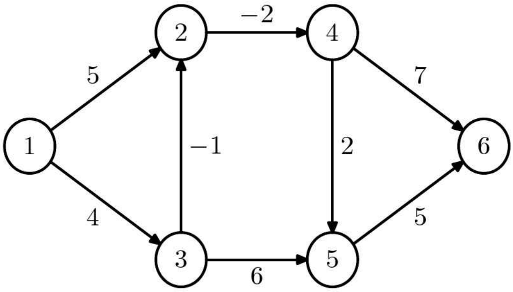


1. $\overline{S} = \{2, 3, 4, 5, 6, 7, 8\}$
   $\Pi(1) = 0$
   $\Pi(2) = 5, \Pi(5) = 6$
   $\Pi(4) = \Pi(5) = \Pi(6) = \Pi(7) = \Pi(8) = \infty$
   $P(2) = P(5) = 1.$
2. $j = 2$, $\overline{S} = \{3, 4, 5, 6, 7, 8\}$
   $\Gamma_2 \cap \overline{S} = \{3\}$
   $\Pi(3) = \min(\infty, 6) = 6$
   $P(3) = 2.$
3. $j = 3$, $\overline{S} = \{4, 5, 6, 7, 8\}$
   $\Gamma_3 \cap \overline{S} = \{6, 7\}$
   $\Pi(6) = \min(\infty, 7) = 7$
   $\Pi(7) = \min(\infty, 8) = 8$
   $P(4) = P(5) = 3.$
4. $j = 5$, $\overline{S} = \{4, 6, 7, 8\}$
   $\Gamma_5 \cap \overline{S} = \{4\}$
   $\Pi(4) = \min(\infty, 11) = 11$
   $P(4) = 5.$
5. $j = 6$, $\overline{S} = \{4, 7, 8\}$
   $\Gamma_6 \cap \overline{S} = \{7, 8\}$
   $\Pi(7) = \min(8, 17) = 8$
   $\Pi(8) = \min(\infty, 17) = 17$
   $P(8) = 6.$
6. $j = 7$, $\overline{S} = \{4, 8\}$
   $\Gamma_7 \cap \overline{S} = \{\emptyset\}.$
7. $j = 4$, $\overline{S} = \{8\}$
   $\Gamma_4 \cap \overline{S} = \{\emptyset\}.$
8. $j = 8$, $\overline{S} = \{\emptyset\}$
   $\Gamma_8 \cap \overline{S} = \{\emptyset\}$.

I valori dei cammini minimi dei nodi sono:
$\Pi(2) = 5$, $\Pi(3) = 6$, $\Pi(4) = 11$, $\Pi(5) = 6$, $\Pi(6) = 7$, $\Pi(7) = 8$, $\Pi(8) = 17$,
mentre l'albero dei cammini minimi è il seguente:



ESEMPIO 3.3. Calcolare il cammino minimo tra il nodo 1 e tutti gli altri nodi del grafo dato:



Soluzione
Dal momento che sono presenti archi di costo negativo, bisogna applicare l'algoritmo di Bellman.

	Π^0	Π^1	Π^2	Π^3	Π^4	Π^5
1	0	∞	∞	∞	∞	∞
2	0	5	4	∞	∞	∞
3	0	3	4	3	10	∞
4	0	3	4	1	5	10
5	0	3	4	1	3	8
6	0	3	4	1	3	8

Nelle ultime due iterazioni i cammini non sono cambiati: abbiamo trovato l'ottimo.
I valori dei cammini minimi sono i seguenti:
$\Pi(1) = 0, \Pi(2) = 3, \Pi(3) = 4, \Pi(4) = 1, \Pi(5) = 3, \Pi(6) = 8.$

ESEMPIO 3.4. Trovare, con l'algoritmo di Bellman, il cammino minimo tra il nodo 1 e tutti gli altri nodi nel grafo la cui matrice delle distanze è riportata di seguito:

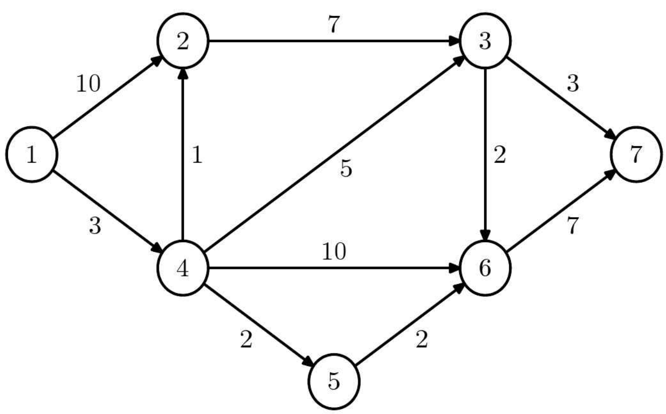
	1	2	3	4	5	6
1	0	10	3	∞	∞	∞
2	∞	0	∞	2	-3	∞
3	∞	1	0	∞	6	∞
4	∞	1	∞	0	∞	4
5	∞	∞	∞	1	0	3
6	∞	∞	∞	∞	∞	0

Soluzione

k	$\Pi(1)$	$\Pi(2)$	$\Pi(3)$	$\Pi(4)$	$\Pi(5)$	$\Pi(6)$
0	0	∞	∞	∞	∞	∞
1	0	10	3	∞	∞	∞
2	0	4	3	12	7	∞
3	0	4	3	6	1	10
4	0	4	3	2	1	4
5	0	3	3	2	1	4
6	0	3	3	2	0	4

Visto che siamo arrivati all'iterazione $k = 6$ senza che l'algoritmo abbia raggiunto la convergenza, possiamo concludere che sul grafo esiste un ciclo di lunghezza negativa.

ESEMPIO 3.5. Trovare il cammino minimo tra il nodo 1 e tutti gli altri nodi nel seguente grafo, applicando l'algoritmo di Dijkstra.



Soluzione

1. Inizializzazione: $\overline{S} = \{2,3,4,5,6,7\}$ $\Pi(1) = 0,\ \Pi(2) = 10,\ \Pi(3) = \infty,\ \Pi(4) = 3,\ \Pi(5) = \Pi(6) = \Pi(7) = \infty$

   $P(2) = 1,\ P(4) = 1$

2. $j = 4,\ \Pi(4) = 3,\ P(4) = 1$

   $\overline{S} = \{2,3,5,6,7\}$, $\Gamma(4) = \{2,3,5,6\}$, $\Gamma(4) \cap \overline{S} = \{2,3,5,6\}$

   $\Pi(2) = \min\{\Pi(2), \Pi(4) + l_{42}\} = \min\{10, 4\} = 4,\ P(2) = 4$

   $\Pi(3) = \min\{\Pi(3), \Pi(4) + l_{43}\} = \min\{\infty, 8\} = 8,\ P(3) = 4$

   $\Pi(5) = \min\{\Pi(5), \Pi(4) + l_{45}\} = \min\{\infty, 5\} = 5,\ P(5) = 4$

   $\Pi(6) = \min\{\Pi(6), \Pi(4) + l_{46}\} = \min\{\infty, 13\} = 13,\ P(6) = 4$

3. $j = 2,\ \Pi(2) = 4,\ P(2) = 4$

   $\overline{S} = \{3,5,6,7\}$, $\Gamma(2) = \{3\}$, $\Gamma(2) \cap \overline{S} = \{3\}$

   $\Pi(3) = \min\{\Pi(3), \Pi(2) + l_{23}\} = \min\{8, 11\} = 8,\ P(3) = 4$

4. $j = 5,\ \Pi(5) = 5,\ P(5) = 4$

   $\overline{S} = \{3,6,7\}$, $\Gamma(5) = \{6\}$, $\Gamma(5) \cap \overline{S} = \{6\}$

   $\Pi(6) = \min\{\Pi(6), \Pi(5) + l_{56}\} = \min\{13, 7\} = 7,\ P(6) = 5$

5. $j = 6,\ \Pi(6) = 7,\ P(6) = 5$

   $\overline{S} = \{3,7\}$, $\Gamma(6) = \{7\}$, $\Gamma(6) \cap \overline{S} = \{7\}$

   $\Pi(7) = \min\{\Pi(7), \Pi(6) + l_{67}\} = \min\{\infty, 14\} = 14,\ P(7) = 6$

6. $j = 3,\ \Pi(3) = 8,\ P(3) = 4$

   $\overline{S} = \{7\}$, $\Gamma(3) = \{6,7\}$, $\Gamma(3) \cap \overline{S} = \{7\}$

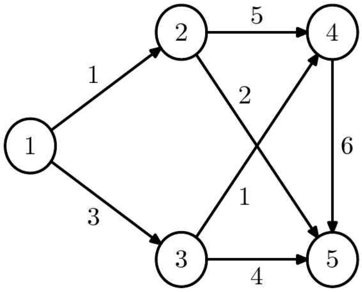
   $\Pi(7) = \min\{\Pi(7), \Pi(3) + l_{37}\} = \min\{14, 11\} = 11,\ P(7) = 3$

7. $j = 7, \Pi(7) = 11, P(7) = 3$

   $\overline{S} = \{\emptyset\}$

I valori dei cammini minimi sono i seguenti:
$\Pi(1) = 0, \Pi(2) = 4, \Pi(3) = 8, \Pi(4) = 3, \Pi(5) = 5, \Pi(6) = 7, \Pi(7) = 11.$

ESEMPIO 3.6. Trovare il cammino minimo tra il nodo 1 e tutti gli altri nodi nel seguente grafo.



Soluzione Il grafo in figura è aciclico, per cui può essere applicato l'algoritmo di Gondran-Minoux. In particolare applicheremo la versione dell'algoritmo che utilizza il valore del rank dei nodi per determinare l'ordine con cui considerarli.
La prima cosa da fare, quindi, è determinare proprio il rank dei nodi.

Iter	1	2	3	4	5	r_i
1	0	1	1	2	3	1
2	-	0	0	2	3	2,3
3		-	-	0	1	4
4				-	0	5

Avendo determinato i valori dei rank dei nodi, possiamo applicare l'algoritmo di Gondran-Minoux considerando i nodi ordinati per rank crescenti. L'ordine è $1-2-3-4-5$. $S=\{1\}$, $\overline{S}=\{2,3,4,5\}$, $\Pi(1)=1$.

1. $j=2$
   $\Pi(2)=\min\{\Pi 1+l_{12}\}=1$
   $S=\{1,2\}$, $\overline{S}=\{3,4,5\}$
2. $j=3$
   $\Pi(3)=\min\{\Pi 1+l_{13}\}=3$
   $S=\{1,2,3\}$, $\overline{S}=\{4,5\}$
3. $j=4$
   $\Pi(4)=\min\{\Pi 2+l_{24},\Pi 3+l_{34}\}=4$
   $S=\{1,2,3,4\}$, $\overline{S}=\{5\}$
4. $j=5$
   $\Pi(5)=\min\{\Pi 2+l_{25},\Pi 3+l_{35},\Pi 4+l_{45}\}=3$
   $S=\{1,2,3,4,5\}$, $\overline{S}=\{\emptyset\}$

3.3 Flusso di costo minimo (Min Cost Flow)

Si consideri una rete composta di n nodi. Per ogni nodo si definisca un coefficiente b_i definito come la differenza tra i flussi uscenti e quelli entranti nel nodo. Ciò vuol dire che $b_i > 0$ individua un'offerta di flusso da parte del nodo i, mentre $b_i < 0$ rappresenta una domanda di flusso da parte del nodo i. Si assume che la rete sia bilanciata, cioè

$$\sum_{i=1}^{n} b_i = 0.$$

Per ogni arco (i, j) si definisce una costante c_{ij} che esprime il costo per unità di flusso che percorre l'arco stesso. Il problema del flusso di costo minimo consiste nel trovare i flussi x_{ij} che percorrono gli archi della rete in modo che il flusso netto in ogni nodo sia esattamente b_i ed il costo totale sia quello minimo. La formulazione in programmazione lineare del problema è la seguente

$$\min \sum_{ij} c_{ij} x_{ij} \tag{3.1}$$

$$\sum_{j} x_{ij} - \sum_{k} x_{ki} = b_i, \quad i = 1, ..., n \tag{3.2}$$

$$x_{ij} \geq 0, \quad i, j = 1, ..., n. \tag{3.3}$$

La matrice A dei coefficienti del sistema (3.2) è la matrice di incidenza nodi-archi.

Definizione 3.1 (Albero ricoprente (Spanning Tree)) Dato un grafo $G = (V, a)$, un albero $T = (V', A')$ si dice ricoprente se è un albero e $V' = V$. Dato il numero di nodi presenti nel grafo $n = |V| = |V'|$, un albero spanning ha esattamente $N - 1$ archi.

Teorema 3.2 Ogni soluzione ammissibile di base del problema di flusso di costo minimo è un albero ricoprente del grafo che rappresenta il problema.

3.3.1 Richiami teorici

L'algoritmo per il problema del flusso di costo minimo presentato di seguito è una versione modificata dell'algoritmo del Simplesso specializzata per il problema in esame. Per tale ragione, anche l'algoritmo di seguito presentato ha bisogno, come nel caso del Simplesso, di una soluzione ammissibile di base di partenza. Inoltre, come nel caso del Simplesso, l'algoritmo cerca la soluzione ottima spostandosi da una soluzione ammissibile di base all'altra.
Nel caso non si abbia a disposizione una soluzione ammissibile di base, l'algoritmo applica una versione specializzata dell'algoritmo della Fase I del Simplesso.

Function Min_Cost_Flow

X : Soluzione Ammissibile di Base corrente

Trovare una Soluzione Ammissibile di Base
$X = Find\_SAB\_Min\_Cost\_Flow$
if $X \neq NULL$ **then**
  optimal = **false**;
  while (*optimal* = **false**) **do**
    if $r_{ij} \geq 0, \forall\ (i,j)$ **then**
      optimal = **true**
    else
      Si consideri un arco $(l,m) \,|\, r_{lm} < 0$. L'arco (l,m) entra in base.
      Si consideri il ciclo $C = \{(l,m),(m,k_p),...,(k_q,l)\}$ che si otterrebbe aggiungendo l'arco (l,m) alla base, se gli archi fossero non-orientati.
      $C^- = \{(i,j) \,|\, (i,j) \in C, (i,j) \neq (l,m), (i,j)$ verso discorde a (l,m) nel ciclo$\}$
      $C^+ = \{(i,j) \,|\, (i,j) \in C, (i,j) \neq (l,m), (i,j)$ verso concorde a (l,m) nel ciclo$\}$
      $\theta = \min_{(i,j)\in C^-} \{x_{ij}\}$
      $x_{ij} = x_{i,j} - \theta, (i,j) \in C^-$
      $x_{ij} = x_{i,j} + \theta, (i,j) \in C^+$
      $x_{lm} = \theta$
      Eliminare dalla base un arco (i,j) avente, dopo l'aggiornamento, flusso nullo.
    end if
  end while
  Return X
else
  Return Not_Exist
end if

Come accennato in precedenza, l'algoritmo del Simplesso per il problema del flusso di costo minimo ha bisogno di una soluzione ammissibile di base di partenza. Per trovare tale soluzione, viene applicata una versione modificata dell'algoritmo di Fase I del Simplesso. Nella descrizione dello pseudocodice, si suppone di disporre di una funzione *IsSpanning* che ritorni **true** se la soluzione data è un albero spanning e **false** altrimenti e di una funzione *CompleteSpanningTree* che, data una soluzione senza cicli ma che non sia spanning, la completi aggiungendo archi presi dal grafo. Questo perché, come vedremo nell'esempio 3.10, può accadere che la Fase I del Simplesso per il flusso di costo minimo contenga, all'ottimo, più di un arco appartenente all'insieme degli archi fittizi sia posto nella soluzione ottima. La condizione per controllare che la soluzione data dalla Fase I del Simplesso per il flusso di costo minimo possa essere una soluzione ammissibile di base (o possa essere completata a formare una soluzione ammissibile di base) è che non esistano, nella soluzione ottima della Fase I, archi fittizi con flusso non-nullo. Si ricorda comunque che, ad ogni modo, una soluzione ottima della Fase I del Simplesso per il flusso di costo minimo contiene almeno un arco fittizio con flusso nullo. Questo si ha a causa dell'aggiunta, nella costruzione del grafo di appoggio $\overline{G}$, del nodo fittizio X. In tal modo l'insieme dei nodi $\overline{V}$ ha cardinalità $N + 1$, per cui un albero spanning del grafo $\overline{G}$ conterrà N archi, cioè uno in più di un qualsiasi albero spanning del grafo G.

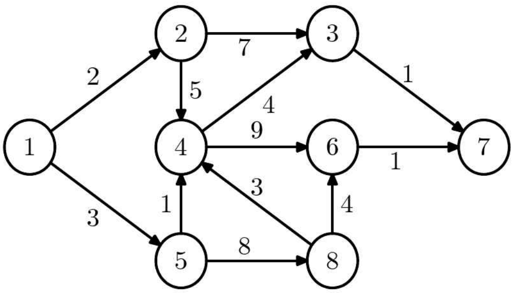
Function Find_SAB_Min_Cost_Flow
$\overline{X}$: soluzione corrente della Fase 1
X_{F1} : soluzione ammissibile di base per il grafo G

Viene creato un grafo di appoggio $\overline{G}$ a partire dal grafo originale G
$\overline{G} = (\overline{V}, \overline{A}), \overline{V} = V \cup \{X\}, \overline{A} = A \cup A_f, A_f = \{x_{iX} : b_i > 0, \} \cup \{x_{Xj} : b_i < 0, \}$
$c_{ij} \begin{cases} 0, & c_{ij} \in A \\ 1, & c_{ij} \in A_f \end{cases}$
$\overline{X} = \{x_{ix} = b_i\} \cup \{x_{xj} = b_j\}$
if not *isSpanning*$(\overline{X}, \overline{G})$ **then**
  $\overline{X} = completeSpanningTree(\overline{G})$
end if
$\overline{X} = MinCostFlow(\overline{G})$
optimal = **true**
$X_{F1} = \{\emptyset\}$
for all $x_{ij} \in \overline{X}$ **do**
  $X_{F1} = X_{F1} \cup x_{ij}$
  if $(x_{ij} > 0)$ **and** $(x_{ij} \in A_f)$ **then**
    optimal = **false**
    break
  end if
end for
if *optimal* = **true then**
  if not *isSpanning*(X_{F1}, G) **then**
    $X_{F1} = completeSpanningTree(G)$
  end if

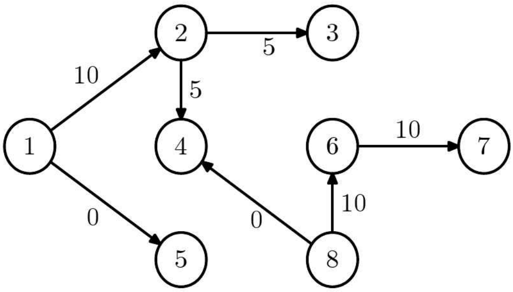
  return X_{F1}
else
  Return Not_Exist
end if

3.3.2 Esercizi risolti

ESEMPIO 3.7. Risolvere il seguente problema di flusso di costo minimo.



dati i seguenti valori per le offerte e le domande dei nodi:
$b_1 = 10$, $b_8 = 10$, $b_4 = -5$, $b_3 = -5$, $b_7 = -10$.
Si consideri, inoltre, la seguente soluzione ammissibile di base iniziale (i numeri presenti sugli archi indicano le quantità di flusso sui rispettivi archi).



Soluzione
Per tutti gli archi presenti in base risulta valida l'equazione:

$$c_{ij} = \lambda_i - \lambda_j.$$

Inoltre, una variabile duale λ_i risulta essere libera, per cui può essere impostata ad un valore arbitrario. Per il seguente esercizio si sceglie la variabile λ_8, ottenendo i seguenti valori:

$\lambda_8 = 0$	$\lambda_2 = \lambda_4 + c_{24} = 2$
$\lambda_6 = \lambda_8 - c_{86} = -4$	$\lambda_1 = \lambda_2 + c_{12} = 4$
$\lambda_7 = \lambda_6 - c_{67} = -5$	$\lambda_3 = \lambda_2 - c_{23} = -5$
$\lambda_4 = \lambda_8 - c_{84} = -3$	$\lambda_5 = \lambda_1 - c_{15} = 1.$

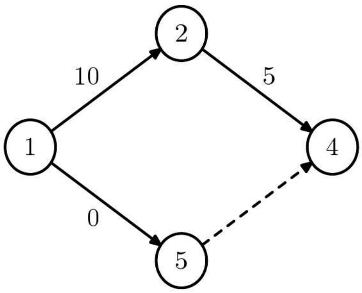
Calcolando i costi ridotti relativi agli archi fuori base mediante l'espressione

$$r_{ij} = c_{ij} - \lambda_i + \lambda_j,$$

si ottengono i seguenti valori dei corti ridotti:

$r_{37} = 1 + 5 - 5 = 1$ $\quad r_{54} = 1 - 1 - 3 = -3$
$r_{43} = 4 + 3 - 5 = 2$ $\quad r_{58} = 8 - 1 + 0 = 7$
$r_{46} = 9 + 3 - 4 = 8.$

Abbiamo ancora dei costi ridotti negativi ($r_{54} = -3$), per cui l'arco $(5, 4)$ è candidato ad entrare in base.
Il ciclo che si ottiene inserendo l'arco $(5, 4)$ è il seguente:



Si imposta quindi la quantità di flusso con cui far entrare il nuovo arco a $\theta = \min\{x_{12}, x_{24}\} = \min\{10, 5\} = 5$.

$C^+ = \{x_{15}\}$

$C^- = \{x_{12}, x_{24}\}$

$x_{54} = 5$

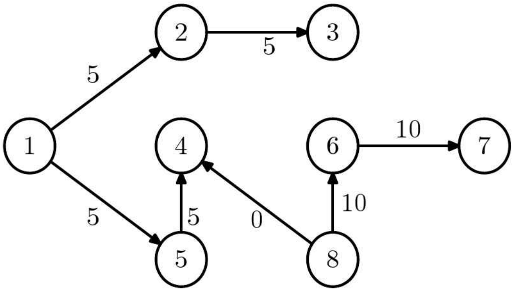
$x_{15} = 5$

$x_{12} = 5$

$x_{24} = 0$, per cui x_{24} esce dalla base.

.

Si ottiene quindi la nuova soluzione di base rappresentata in figura:



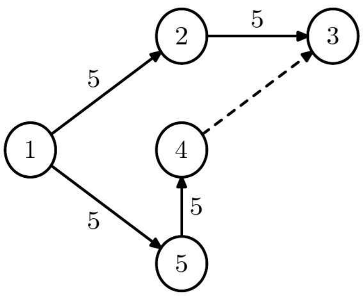
A questa nuova soluzione di base corrispondono i seguenti moltiplicatori:

$\lambda_8 = 0$	$\lambda_5 = \lambda_4 + c_{54} = -2$
$\lambda_6 = \lambda_8 - c_{86} = -4$	$\lambda_1 = \lambda_5 + c_{15} = 1$
$\lambda_7 = \lambda_6 - c_{67} = -5$	$\lambda_2 = \lambda_1 - c_{12} = -1$
$\lambda_4 = \lambda_8 - c_{84} = -3$	$\lambda_3 = \lambda_2 - c_{23} = -8.$

Agli archi fuori base corrispondono i seguenti costi ridotti:

$r_{24} = 5 + 1 - 3 = 3$	$r_{46} = 9 + 3 - 4 = 8$
$r_{37} = 1 + 8 - 5 = 4$	$r_{58} = 8 + 2 - 0 = 10$
$r_{43} = 4 + 3 - 8 = -1.$	

L'arco candidato ad entrare in base risulta essere r_{43}, inserito il quale si ottiene il seguente ciclo.



$$\theta = \min\{x_{12}, x_{23}\} = \min\{5, 5\} = 5$$

$$C^+ = \{x_{12}, x_{23}\}$$

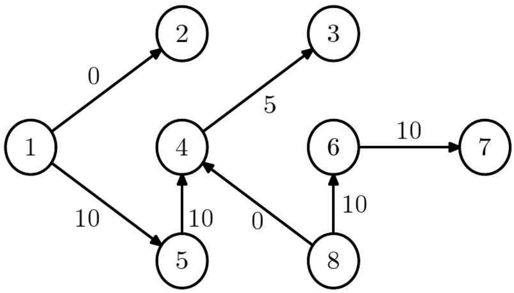
$$C^- = \{x_{15}, x_{54}\}$$

$$x_{43} = 5$$

$$x_{15} = 10,\ x_{54} = 10$$

$$x_{12} = 0,\ x_{23} = 0.$$

Essendo a flusso nullo sia x_{12} che x_{23}, entrambe le variabili sono candidate ad uscire dalla base. Scegliamo di far uscire x_{23}.
Si ottiene, quindi, la seguente nuova soluzione di base:



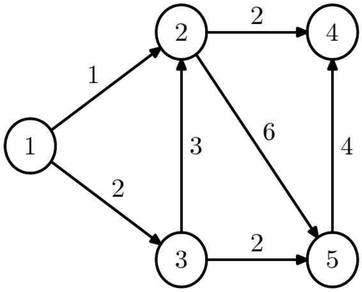
A questa nuova soluzione corrispondono i seguenti moltiplicatori:

$\lambda_8 = 0$	$\lambda_5 = \lambda_4 + c_{54} = -2$
$\lambda_6 = \lambda_8 - c_{86} = -4$	$\lambda_1 = \lambda_5 + c_{15} = 1$
$\lambda_7 = \lambda_6 - c_{67} = -5$	$\lambda_2 = \lambda_1 - c_{12} = -1$
$\lambda_4 = \lambda_8 - c_{84} = -3$	$\lambda_3 = \lambda_4 - c_{43} = -7.$

Agli archi fuori base corrispondono i seguenti costi ridotti: $r_{23} = 1$, $r_{24} = 3$, $r_{37} = 3$, $r_{46} = 8$, $r_{58} = 10$.
Avendo tutti i costi ridotti associato agli archi fuori base non-negativi, abbiamo raggiunto l'ottimo. Il costo di questa soluzione vale

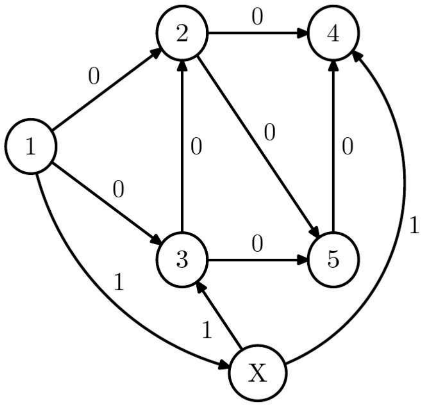
$$\overline{z} = 0 \cdot 2 + 10 \cdot 3 + 10 \cdot 1 + 5 \cdot 4 + 0 \cdot 3 + 10 \cdot 4 + 10 \cdot 1 = 110.$$

ESEMPIO 3.8. Risolvere il seguente problema di flusso di costo minimo.

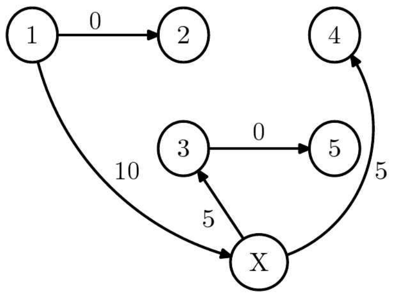


I valori per le offerte e le domande dei nodi sono i seguenti:
$b_1 = 10$, $b_3 = -5$, $b_4 = -5$.
Si consideri, inoltre, di porre $\lambda_5 = 0$ nel calcolo delle variabili duali.

Soluzione
Il problema non presenta una soluzione iniziale, per cui applichiamo la Fase 1 per il metodo del Simplesso per il problema del flusso di costo minimo.
Aggiungiamo un nodo fittizio X e lo colleghiamo ai nodi di domanda e di offerta. Per ogni nodo di offerta i, aggiungiamo un arco artificiale (i, X), mentre per ogni nodo di domanda j aggiungiamo un arco artificiale (X, j). I costi associati agli archi del problema originale sono $c_{ij} = 0$, mentre gli archi artificiali hanno costo $c_{ij} = 1$. Il grafo diventa quindi il seguente:



Una soluzione iniziale si ottiene mettendo in base tutti gli archi artificiali con flusso pari alla domanda o all'offerta del nodo al quale sono associati e completando la base del problema con archi del problema originale che colleghino gli archi di transito.



A questa soluzione corrispondono i seguenti moltiplicatori:

$$\lambda_5 = 0,\ \lambda_3 = 0,\ \lambda_2 = 2$$

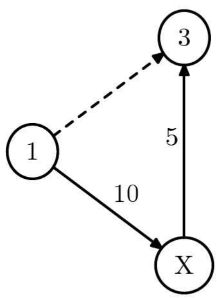
$$\lambda_X = 1,\ \lambda_4 = 0,\ \lambda_1 = 2.$$

Agli archi fuori base corrispondono i seguenti costi ridotti:

$$r_{13} = -2,\ r_{24} = -2, r_{25} = -2$$

$$r_{32} = 2,\ r_{54} = 0.$$

L'arco candidato ad entrare in base risulta essere r_{13}, inserito il quale si ottiene il seguente ciclo.



.

$$\theta = \min\{x_{1X}, x_{X3}\} = \min\{10, 5\} = 5$$

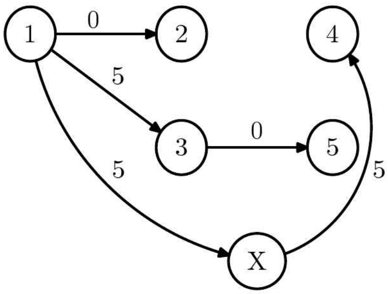
$$C^+ = \{\emptyset\}$$

$$C^- = \{x_{1X}, x_{X3}\}$$

$$x_{13} = 5$$

$$x_{1X} = 5,\ x_{X3} = 0.$$

La variabile x_{X3} esce dalla base, ottenendo la nuova soluzione in figura:



A questa soluzione corrispondono i seguenti moltiplicatori:

$$\lambda_5 = 0,\ \lambda_3 = 0,\ \lambda_1 = 0$$

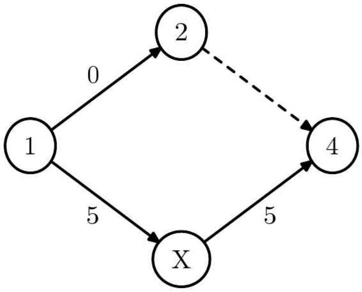
$$\lambda_2 = 0,\ \lambda_X = -1,\ \lambda_4 = -2.$$

Agli archi fuori base corrispondono i seguenti costi ridotti:

$$r_{X3} = 2,\ r_{24} = -2,\ r_{25} = 0$$

$$r_{32} = 0,\ r_{54} = -2.$$

L'arco candidato ad entrare in base risulta essere r_{24}, inserito il quale si ottiene il seguente ciclo.



$$\theta = \min\{x_{1X}, x_{X4}\} = \min\{5, 5\} = 5$$

$$C^+ = \{\ x_{12}\}$$

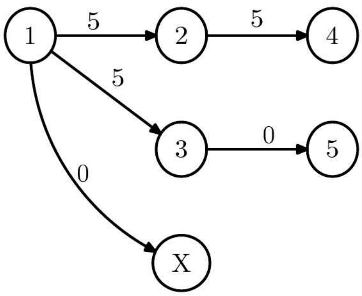
$$C^- = \{x_{1X}, x_{X4}\}$$

$$x_{24} = 5$$

$$x_{12} = 5$$

$$x_{1X} = 0,\ x_{X4} = 0.$$

Essendo a flusso nullo sia x_{1X} che x_{X4}, entrambe le variabili sono candidate ad uscire dalla base. Scegliamo di far uscire x_{X4}. La nuova soluzione diventa la seguente:



A questa soluzione corrispondono i seguenti moltiplicatori:

$$\lambda_5 = 0,\ \lambda_3 = 0,\ \lambda_1 = 0$$

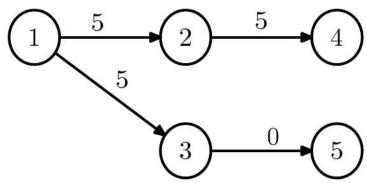
$$\lambda_2 = 0,\ \lambda_X = -1,\ \lambda_4 = 0.$$

Agli archi fuori base corrispondono i seguenti costi ridotti:

$$r_{X3} = 2,\ r_{X4} = 2, r_{25} = 0$$

$$r_{32} = 0,\ r_{54} = 0.$$

La soluzione ottenuta è ottima per il problema di Fase I. La soluzione iniziale per il problema originale si ottiene semplicemente eliminando il nodo fittizio e l'arco artificiale che incide su di esso. Si noti che, alla fine della Fase I del Simplesso per il problema del flusso di costo minimo, se il problema originale ammette soluzione, si ottiene una soluzione ottima della Fase I contenente un arco artificiale con flusso nullo. Ciò deriva dal fatto che, nel costruire il problema di Fase I, si aggiunge il nodo fittizio e, quindi, una soluzione ammissibile di base della Fase I è composta di N archi (mentre una soluzione ammissibile di base del problema originale è composta da $N-1$ archi).
La soluzione iniziale del nostro problema è la seguente:



I moltiplicatori associati alla soluzione sono:

$$\lambda_5 = 0,\ \lambda_3 = 2,\ \lambda_1 = 4$$

$$\lambda_2 = 3,\ \lambda_4 = 1.$$

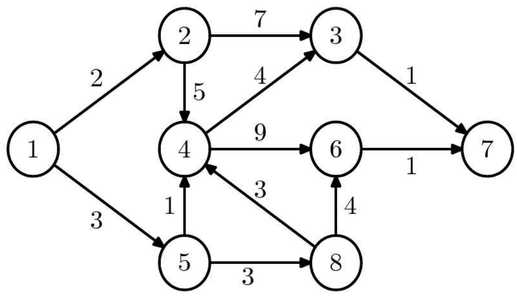
Agli archi fuori base corrispondono i seguenti costi ridotti:

$$r_{25} = 6 - 3 + 0 = 3,\ r_{32} = 3 - 2 + 3 = 4,\ r_{54} = 4 - 0 + 1 = 5.$$

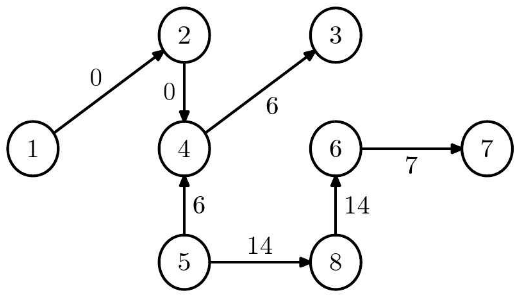
Essendo tutti i costi ridotti non-negativi, la soluzione è ottima.
La funzione obiettivo vale:

$$\overline{z} = 5 \cdot 1 + 5 \cdot 2 + 5 \cdot 2 + 0 \cdot 2 = 25.$$

ESEMPIO 3.9. Risolvere il seguente problema di flusso di costo minimo.



I valore di domanda ed offerta dei nodi siano i seguenti:
$b_5 = 20$, $b_6 = -7$, $b_7 = -7$, $b_3 = -6$.
Si consideri, inoltre, la soluzione ammissibile di base in figura (i numeri presenti sugli archi indicano le quantità di flusso sui rispettivi archi) e di porre, nel calcolo delle variabili duali, $\lambda_8 = 0$.



Soluzione
I moltiplicatori associati alla soluzione sono:

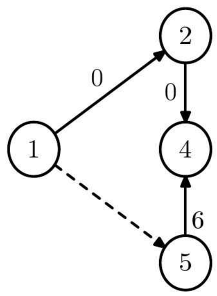
$$\lambda_8 = 0, \lambda_6 = -4, \lambda_7 = -5, \lambda_5 = 3, \lambda_4 = 2, \lambda_3 = -2, \lambda_2 = 7, \lambda_1 = 9.$$

Agli archi fuori base corrispondono i seguenti costi ridotti:

$$r_{15} = 3 - 9 + 3 = -3,\ r_{23} = 7 - 7 - 2 = -2,\ r_{37} = 1 + 2 - 5 = -2,$$

$$r_{46} = 9 - 2 - 4 = 3,\ r_{84} = 3 - 0 + 2 = 5.$$

L'arco candidato ad entrare in base risulta essere r_{15}, inserito il quale si ottiene il seguente ciclo.



$$\theta = \min\{x_{12}, x_{24}\} = \min\{0, 0\} = 0$$

$$C^+ = \{x_{54}\}$$

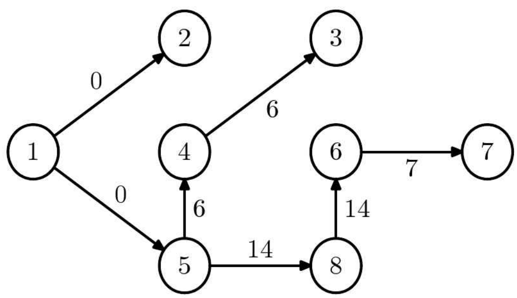
$$C^- = \{x_{12}, x_{24}\}$$

$$x_{15} = 0$$

$$x_{54} = 0$$

$$x_{12} = 0, \, x_{24} = 0.$$

Essendo a flusso nullo sia x_{12} che x_{24}, entrambe le variabili sono candidate ad uscire dalla base. Scegliamo di far uscire x_{24}. La nuova soluzione diventa la seguente:



I moltiplicatori associati alla soluzione sono:

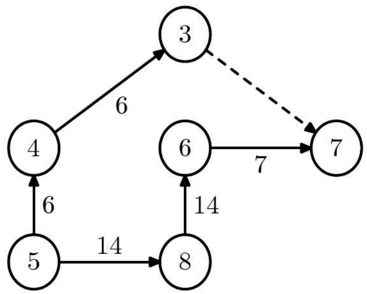
$$\lambda_8 = 0, \lambda_6 = -4, \lambda_7 = -5, \lambda_5 = 3, \lambda_1 = 6, \lambda_2 = 4, \lambda_4 = 2, \lambda_3 = -2.$$

Agli archi fuori base corrispondono i seguenti costi ridotti:

$$r_{23} = 7 - 4 - 2 = 1,\ r_{24} = 5 - 4 + 2 = 3,\ r_{37} = 1 + 2 - 5 = -2,$$

$$r_{46} = 9 - 2 - 4 = 3,\ r_{84} = 3 - 0 + 2 = 5.$$

L'arco candidato ad entrare in base risulta essere r_{37}, inserito il quale si ottiene il seguente ciclo.



$$\theta = \min\{\ x_{58}, x_{86}, x_{67}\} = \min\{14, 14, 7\} = 7$$

$$C^+ = \{x_{54}, x_{43}\}$$

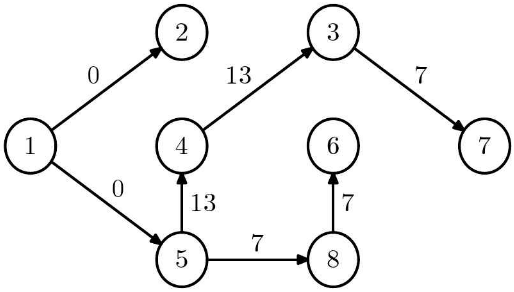
$$C^- = \{\ x_{58}, x_{86}, x_{67}\}$$

$$x_{37} = 7$$

$$x_{58} = 7,\ x_{86} = 7,\ x_{67} = 0$$

$$x_{54} = 13,\ x_{43} = 13.$$

Essendo nullo il flusso di x_{67}, tale variabile esce dalla base. La nuova soluzione diventa la seguente:



I moltiplicatori associati alla soluzione sono:

$$\lambda_8 = 0, \lambda_6 = -4, \lambda_5 = 3, \lambda_1 = 6, \lambda_2 = 4, \lambda_4 = 2, \lambda_3 = -2, \lambda_7 = -3.$$

Agli archi fuori base corrispondono i seguenti costi ridotti:

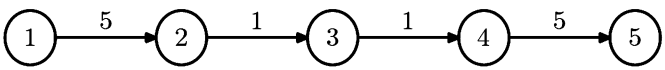
$$r_{23} = 7 - 4 - 2 = 1,\ r_{24} = 5 - 4 + 2 = 3,\ r_{46} = 9 - 2 - 4 = 3,$$

$$r_{67} = 1 + 4 - 3 = 2,\ r_{84} = 3 - 0 + 2 = 5.$$

Avendo ottenuto tutti costi ridotti non-negativi, la soluzione è ottima. La funzione obiettivo vale: $\overline{z} = 0 \cdot 2 + 0 \cdot 3 + 13 \cdot 1 + 13 \cdot 4 + 7 \cdot 1 + 7 \cdot 3 + 7 \cdot 4 = 121$.

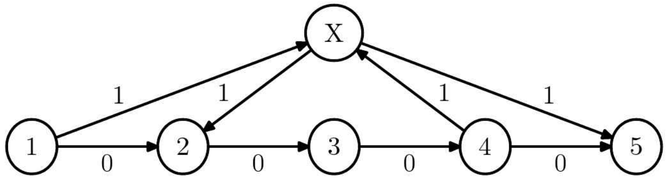
ESEMPIO 3.10.

Si consideri il seguente problema di flusso di costo minimo.

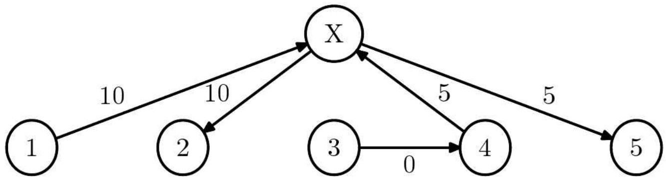


Dati i seguenti valori per le offerte e le domande dei nodi:
$b_1 = 10$, $b_2 = -10$, $b_4 = 5$, $b_5 = -5$,
si ricavi una soluzione ammissibile di base.

Soluzione Il problema non presenta una soluzione iniziale, per cui la Fase I per il Simplesso per il problema del flusso di costo minimo.
Aggiungiamo un nodo fittizio X e lo colleghiamo ai nodi di domanda e di offerta. Per ogni nodo di offerta i, aggiungiamo un arco artificiale (i, X), mentre per ogni nodo di domanda j aggiungiamo un arco artificiale (X, j). I costi associati agli archi del problema originale sono $c_{ij} = 0$, mentre gli archi artificiali hanno costo $c_{ij} = 1$. Il grafo diventa quindi il seguente:



Una soluzione iniziale si ottiene mettendo in base tutti gli archi artificiali con flusso pari alla domanda o all'offerta del nodo al quale sono associati e completando la base del problema con archi del problema originale che colleghino gli archi di transito.



A questa soluzione corrispondono i seguenti moltiplicatori:

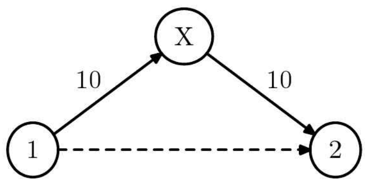
$$\lambda_5 = 0,\ \lambda_X = 1,\ \lambda_4 = 2$$

$$\lambda_3 = 2,\ \lambda_2 = 0,\ \lambda_1 = 2.$$

Agli archi fuori base corrispondono i seguenti costi ridotti:

$$r_{12} = -2,\ r_{23} = 2, r_{45} = -2.$$

L'arco candidato ad entrare in base risulta essere x_{12}, inserito il quale si ottiene il ciclo $1 - 2 - X$.



.

$$\theta = \min\{x_{1X}, x_{X2}\} = \min\{10, 10\} = 10$$

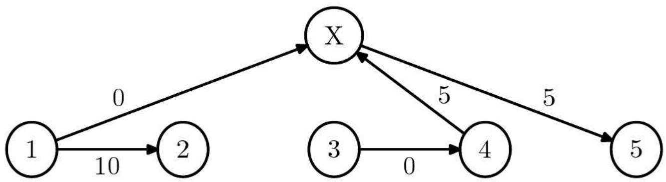
$$C^+ = \{\emptyset\}$$

$$C^- = \{x_{1X}, x_{X2}\}$$

$$x_{12} = 10$$

$$x_{1X} = 0,\ x_{X2} = 0.$$

La variabile x_{X2} esce dalla base, ottenendo la nuova soluzione seguente:



A questa soluzione corrispondono i seguenti moltiplicatori:

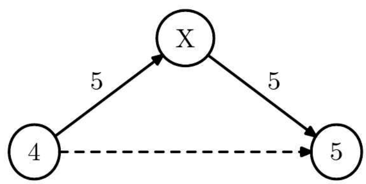
$$\lambda_5 = 0,\ \lambda_X = 1,\ \lambda_4 = 2$$

$$\lambda_3 = 2,\ \lambda_2 = 2,\ \lambda_1 = 2,$$

ai quali corrispondono i seguenti costi ridotti:

$$r_{X2} = 2,\ r_{23} = 0, r_{45} = -2.$$

Essendovi costi ridotti negativi, non siamo ancora all'ottimo. L'arco candidato ad entrare in base risulta essere x_{45}, inserito il quale si ottiene il ciclo $4 - 5 - X$.



.

$$\theta = \min\{x_{4X}, x_{X5}\} = \min\{5, 5\} = 5$$

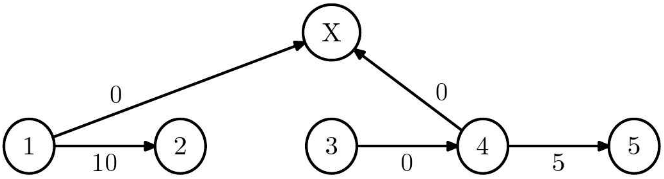
$$C^{+} = \{\emptyset\}$$

$$C^{-} = \{x_{4X}, x_{X5}\}$$

$$x_{45} = 5$$

$$x_{4X} = 0,\ x_{X5} = 0.$$

La variabile x_{X5} esce dalla base, ottenendo la nuova soluzione seguente:



A questa soluzione corrispondono i seguenti moltiplicatori:

$$\lambda_5 = 0,\ \lambda_X = -1,\ \lambda_4 = 0$$

$$\lambda_3 = 0,\ \lambda_2 = 0,\ \lambda_1 = 0,$$

ai quali corrispondono i seguenti costi ridotti:

$$r_{X2} = 1,\ r_{23} = 0, r_{X5} = 1.$$

I costi ridotti sono tutti non-negativi, per cui siamo giunti all'ottimo. Se analizziamo la soluzione trovata, ci accorgiamo che essa è degenere, ma che contiene due archi fittizi con flusso nullo, per cui, una volta rimosso l'arco fittizio, ci ritroveremmo con una soluzione che non è un albero. Per tale motivo, dobbiamo completare la soluzione con un qualsiasi arco che renda uno Spanning Tree la soluzione trovata. Nel nostro esempio, la scelta è obbligata: dobbiamo aggiungere l'arco $(2, 3)$.
Tale situazione si verifica perché, quando andiamo a risolvere il problema di flusso di costo minimo, l'algoritmo non è in grado di scegliere tra il contributo di un arco fittizio con costo unitario ma flusso nullo e quello di un arco reale con costo nullo. L'algoritmo della Fase I del flusso di costo minimo, quindi, a differenza dell'analogo per il Simplesso, può generare una soluzione ammissibile che, per essere di base, deve essere completata con variabili scelte opportunamente ed il cui valore viene fissato a 0.

3.4 Massimo flusso (Max flow)

3.4.1 Richiami teorici

Si consideri un grafo orientato $G = (N, A)$. Si supponga che esista su tale grafo un nodo s che abbia solo archi uscenti ed un nodo t avente solo archi entranti: tali nodi sono chiamati rispettivamente sorgente e pozzo. Se x_{ij} è il flusso sull'arco (i, j) e c_{ij} è la capacità massima di tale arco, quello che si vuole determinare è il valore totale del flusso da s a t.
Il modello di programmazione lineare del problema del flusso massimo è il seguente:

$$\max\ v$$

$$\sum_{i=1}^{n} x_{ij} - \sum_{k=1}^{n} x_{jk} = \begin{cases} -v, & j = s \\ 0, & j \neq s, t \\ v, & j = t \end{cases}$$

$$x_{ij} \leq c_{ij}, \quad \forall (i, j) \in A.$$

Definizione 3.3 (Taglio) Dato un grafo $G = (N, A)$ ed un sottoinsieme S di N tale che $s \in S$ e $t \notin S$, l'insieme di archi H_S di G aventi un estremo in S e l'altro estremo in $T = N \setminus S$ viene chiamato *taglio*. L'insieme H_S è partizionabile in due sottoinsiemi F_S e B_S definiti nella seguente maniera

- F_S : insieme degli *archi in avanti* del taglio, cioè degli archi diretti da S a T (archi *forward*);
- B_S : insieme degli *archi all'indietro* del taglio, cioè degli archi diretti da T ad S (archi *backward*).

Si definisce *capacità del taglio* la quantità

$$c(H_S) = \sum_{(i,j) \in F_S} c_{ij}.$$

Lemma 3.4 Per ogni taglio H_S il flusso sul grafo vale

$$v = \sum_{(i,j) \in F_S} x_{ij} - \sum_{(i,j) \in B_S} x_{ij}.$$

Lemma 3.5 Dato un qualsiasi flusso sul grafo G che soddisfi i vincoli del problema ed un qualunque taglio H_S, si ha $v \leq c(H_S)$.

Corollario 3.6 Se la disuguaglianza del lemma precedente è soddisfatta all'uguaglianza, il valore del flusso v è massimo ed il taglio H_S è un taglio di minima capacità.

Definizione 3.7 (Grafo di scarto) Dato un grafo $G = (N, A)$, il *grafo di scarto* è un grafo orientato $\overline{G} = (N, \overline{A})$ottenuto da G assegnando ad ogni arco di G al più due archi. In particolare ad un generico arco $(i, j) \in A$ attraversato da un flusso x_{ij} ed avente capacità c_{ij} corrispondono i seguenti archi di $\overline{A}$

- un arco a^+ da i a j se $x_{ij} < c_{ij}$;
- un arco a^- da j a i se $x_{ij} > 0$.

A tali archi viene assegnata una lunghezza, che vale

$$\begin{array}{ll} l_{ij} = c_{ij} - x_{ij} & \text{nel caso di } a^+, \\ l_{ji} = x_{ij} & \text{nel caso di } a^-. \end{array}$$

Di seguito si riporta lo pseudocodice dell'algoritmo di Ford-Fulkerson per trovare il massimo flusso tra il nodo sorgente s ed il nodo pozzo t.

Function Max_Flow
$\overline{G}$: grafo di scarto
$\lambda(i, j)$: variabile booleana che vale **true** se l'arco (i, j) del grafo di scarto è di tipo a^+, **false** altrimenti

```
v = 0,  x_ij = 0, ∀i, j ∈ A
optimal = false
while (optimal = false) do
  A̅ = {∅}
  for all ((i, j) ∈ A) do
    if (x_ij < c_ij) then
      A̅ = A̅ ∪ (i, j),  λ(i, j) = true, l_ij = c_ij − x_ij
    end if
    if (x_ij > 0) then
      A̅ = A̅ ∪ (j, i),  λ(j, i) = false, l_ji = x_ij
    end if
    Cercare un cammino elementare P su G̅ da s a t
    if (∄P) then
      optimal = true
    else
      ε = ∞
      for all ((i, j) ∈ P) do
        ε = min{ε, l_ij}
      end for
      v = v + ε
      for all ((i, j) ∈ P) do
        if (λ_ij = true) then
          x_ij = x_ij + ε
```

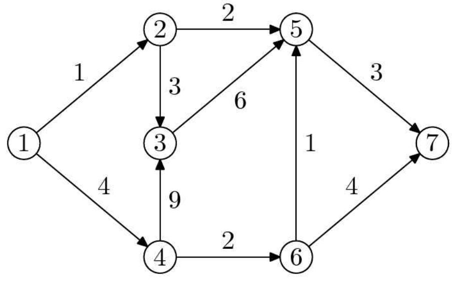
```
            else
               x_ij = x_ij − ε
            end if
         end for
      end if
   end for
end while

return  v
```

Il problema del massimo flusso ha, come problema duale, il problema del taglio di capacità minima. Ciò implica che il flusso massimo instradabile tra il nodo s ed il nodo d sia pari alla capacità del taglio con la minima capacità. Il processo per determinare, a partire dalla soluzione ottima del problema del flusso massimo, uno dei tagli di capacità minima (i tagli possono essere più di uno, è il valore della capacità minima ad essere unico) è molto semplice. Si consideri il grafo di scarto ottenuto all'ultima iterazione dell'algoritmo di Ford-Fulkerson. Visto che, essendo all'ottimo, non esiste nessun cammino tra s e d, esisteranno dei nodi raggiungibili da s, mentre altri nodi (eventualmente il solo nodo d), sono non raggiungibili a partire da s. Il taglio si ottiene ponendo in S i nodi raggiungibili da s sul grafo di scarto e in $N \setminus S$ d e gli altri nodi non raggiungibili sul grafo di scarto a partire da s. Si noti che, dati gli archi del taglio di capacità minima, gli archi forward del taglio sono saturi, mentre quelli backward sono completamente scarichi.

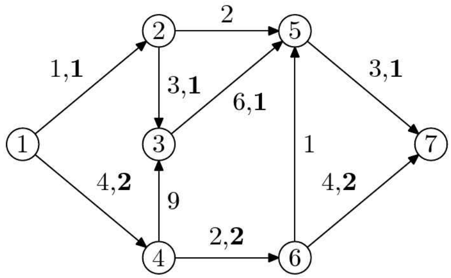
3.4.2 Esercizi risolti

ESEMPIO 3.11. Dato il grafo capacitato di figura con sorgente al nodo 1 e pozzo al nodo 7, trovare il massimo flusso partendo dalla soluzione iniziale ($x_{12} = 1$, $x_{14} = 2$, $x_{23} = 1$, $x_{35} = 1$, $x_{46} = 2$, $x_{57} = 1$, $x_{67} = 2$).

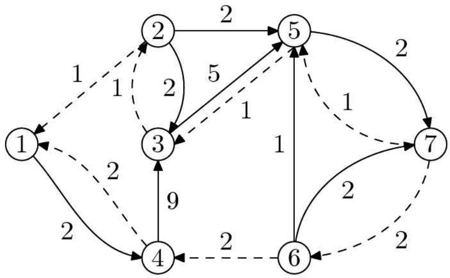


Soluzione

Dalla soluzione iniziale con flusso $v = 3$ otteniamo la situazione di figura (a) (**flussi in neretto**), alla quale corrisponde il grafo di scarto di figura (b).



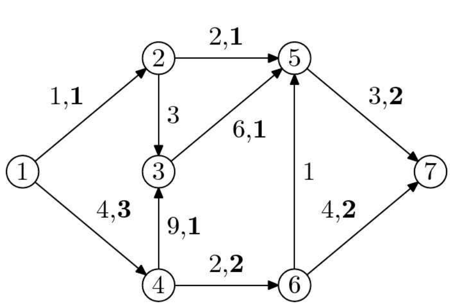
(a)



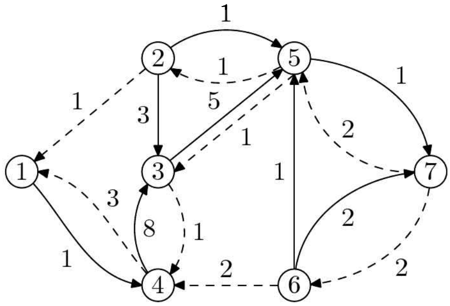
(b)

1. Un possibile cammino aumentante è $(1, 4, 3, 2, 5, 7)$, che contiene l'arco inverso $(3, 2)$. La quantità ϵ della quale possiamo aumentare il flusso lungo tale cammino è $\epsilon = \min(2, 9, 1, 2, 2) = 1$. Aumentano quindi gli archi $(1, 4)$, $(4, 3)$, $(2, 5)$, $(5, 7)$ e diminuisce l'arco $(2, 3)$.

2. La nuova situazione (flusso $v = 4$) è rappresentata in figura (a), con il relativo grafo di scarto (figura (b)).

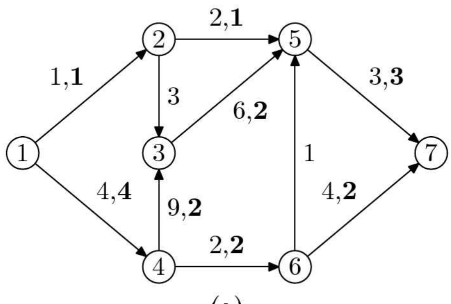
   (a)

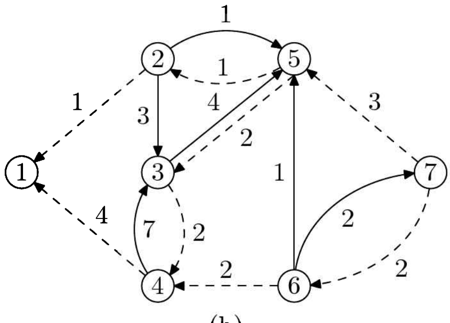
   (b)

   A questo punto un possibile cammino aumentante è dato da $(1, 4, 3, 5, 7)$, con $\epsilon = 1$.

3. Ripetendo il passo di incremento del flusso arriviamo alla situazione di figura (a), nella quale il grafo di scarto (figura (b)) certifica l'ottimalità (non ci sono più cammini orientati che collegano 1 a 7 sul grafo di scarto). Il massimo flusso è quello corrente ($v = 5$).



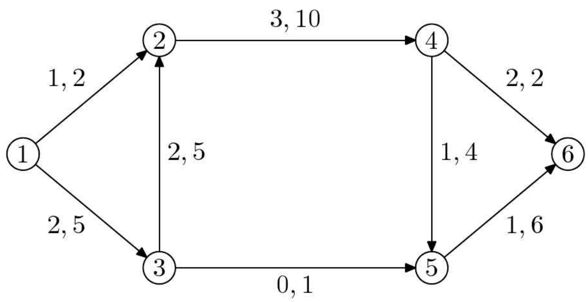
(a)



(b)

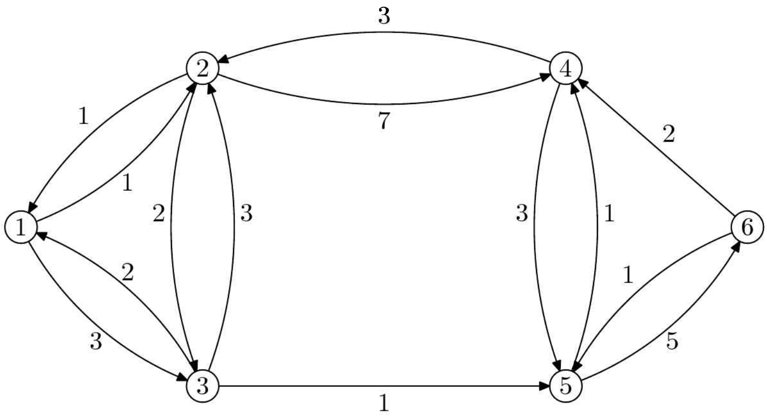
Imponendo $S = \{1\}$, si ottiene il taglio di capacità minima dato dagli archi $(2,1)$ e $(4,1)$.

ESEMPIO 3.12. Risolvere, con l'algoritmo di Ford-Fulkerson, il problema di flusso massimo in figura con sorgente al nodo 1 e pozzo al nodo 6 ed in cui, per ogni arco, sono specificati il flusso iniziale ed il massimo flusso che può attraversarlo.



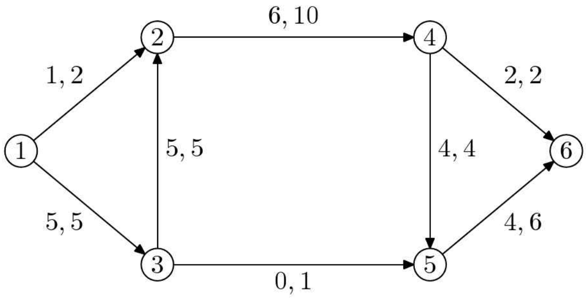
Soluzione

1. Il flusso iniziale è $v = 3$. Il grafo di scarto che si ricava dal grafo iniziale è il seguente:

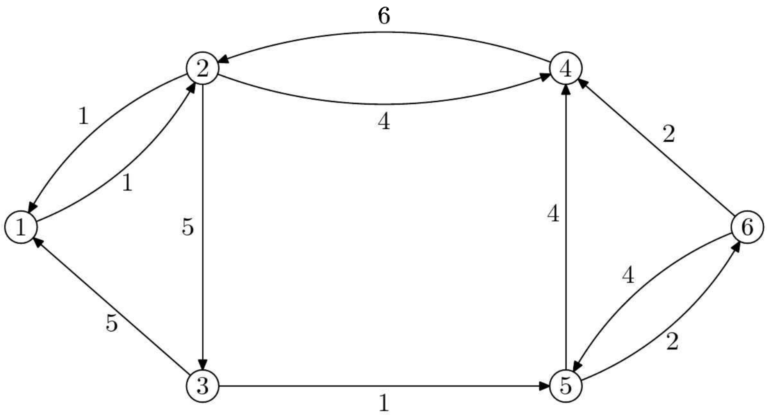


Su tale grafo di scarto si individua il cammino aumentante il cammino $(1, 3, 2, 4, 5, 6)$, con $\epsilon = \min\{3, 3, 7, 3, 5\} = 3$.

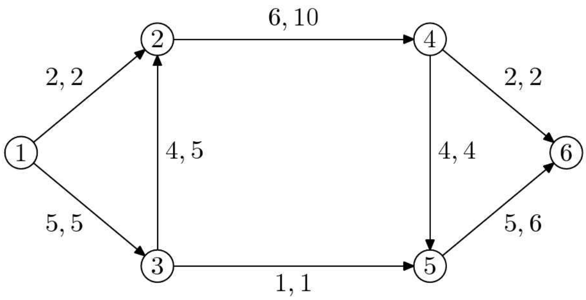
2. Il grafo aggiornato è:



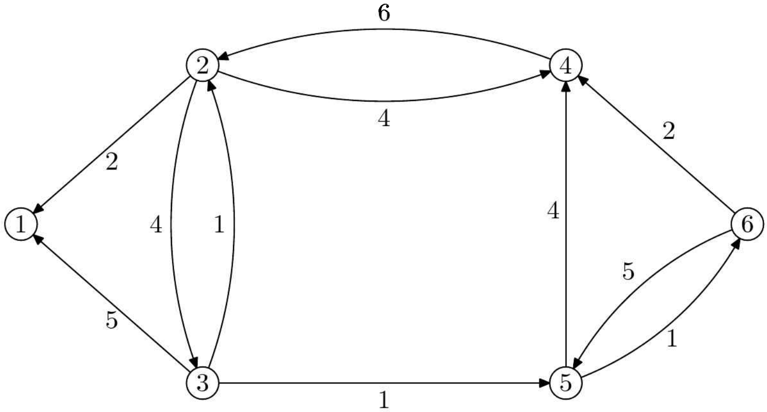
Il flusso diventa $v = 3 + \epsilon = 6$. Sul nuovo grafo di scarto si individua il cammino $(1, 2, 3, 5, 6)$, con $\epsilon = \min\{1, 5, 1, 2\} = 1$.



3. Il grafo aggiornato, relativo al flusso $v = 6 + 1 = 7$, è:

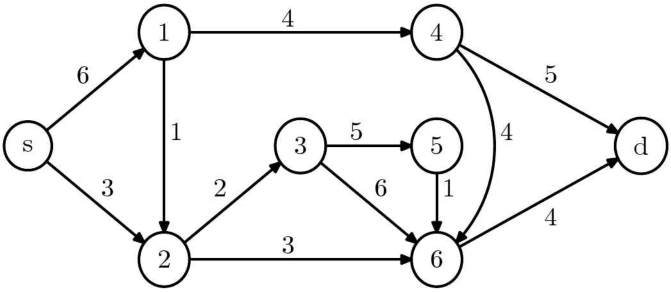


Sul nuovo grafo di scarto non è possibile individuare nuovi cammini elementari tra 1 e 6: l'algoritmo è terminato. Il flusso massimo è pari a 7.



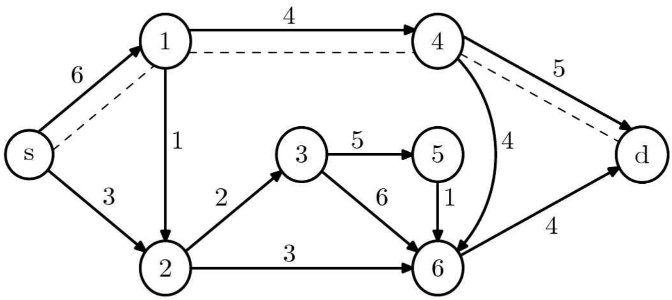
Imponendo $S = \{1\}$, si ottiene il taglio di capacità minima dato dagli archi $(1, 2)$ e $(1, 3)$.

ESEMPIO 3.13. Risolvere, con l'algoritmo di Ford-Fulkerson, il problema di flusso massimo in figura con sorgente al nodo s e pozzo al nodo d ed in cui, per ogni arco, è specificato il massimo flusso che può attraversarlo.

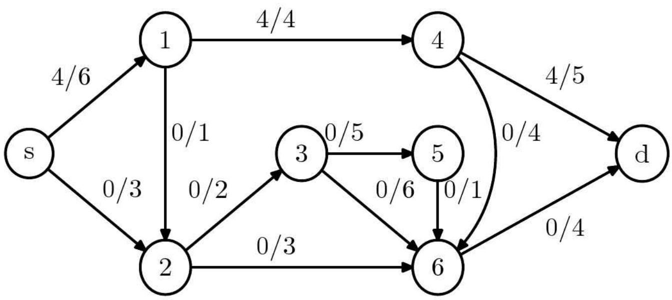


Soluzione

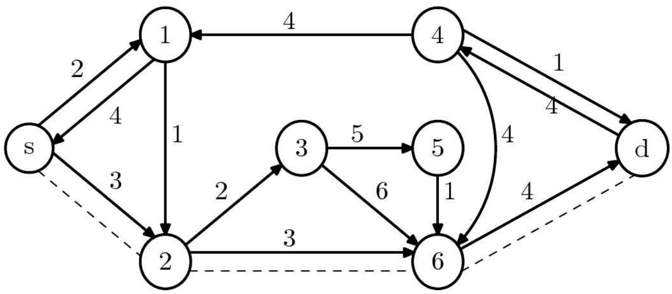
1. Visto che non sono stati dati flussi iniziali, si considera il grafo scarico, per cui si ottiene il seguente grafo di scarto, sul quale si individua il cammino elementare $(s, 1, 4, 3, d)$, evidenziato in figura tramite una linea tratteggiata.



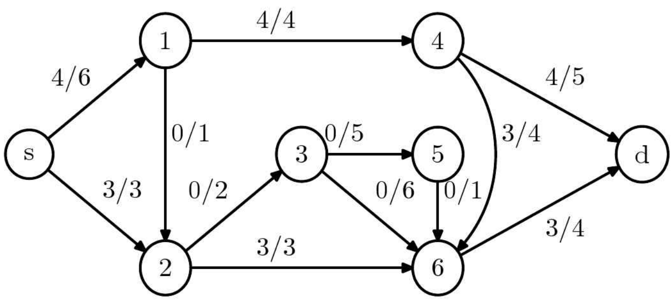
Sul grafo originale tale cammino elementare corrisponde ad un incremento del flusso pari a $\epsilon = \min\{6, 4, 5\} = 4$, da cui il flusso totale che attraversa il grafo diventa $v = 4$. Dopo l'aggiornamento, il grafo si presenta nella situazione seguente.



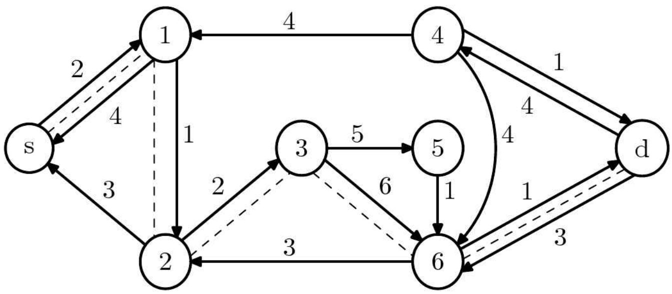
2. Aggiornato il grafo di scarto, si individua il nuovo cammino elementare $(s, 2, 6, d)$, a cui corrisponde sul grafo originale un cammino aumentante con flusso $\epsilon = \min\{3, 3, 4\} = 3$.



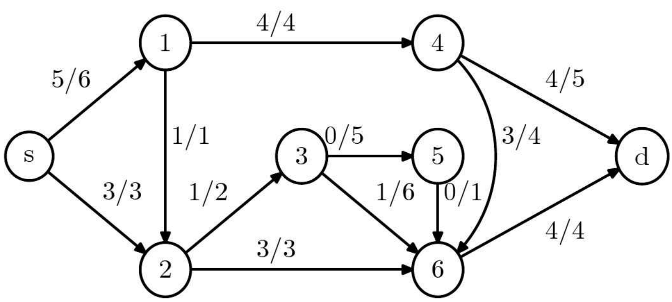
Il flusso totale sul grafo è $v = 7$.



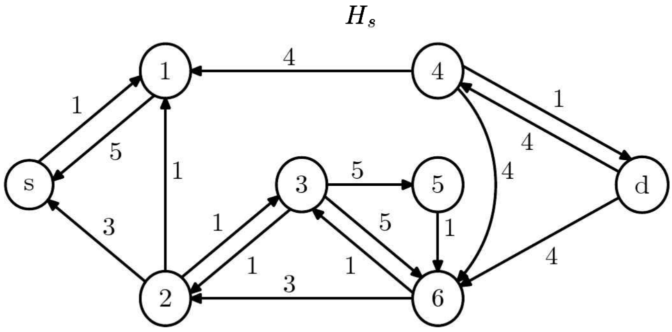
3. Aggiornato il grafo di scarto, si individua il nuovo cammino elementare $(s, 1, 2, 3, 6, d)$, a cui corrisponde sul grafo originale un cammino aumentante con flusso $\epsilon = \min\{2, 1, 2, 6, 1\} = 1$.



Il flusso totale sul grafo è $v = 8$.



4. Il nuovo grafo di scarto è il seguente:



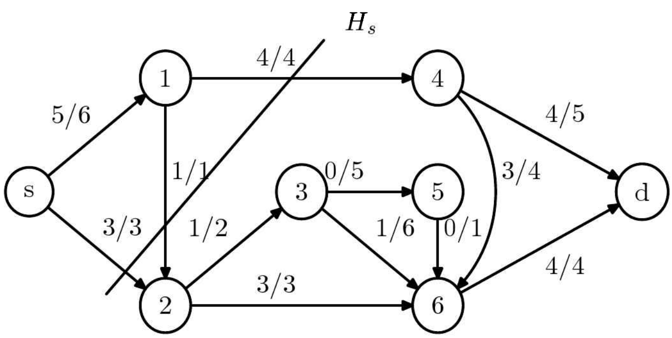
Dopo l'aggiornamento del grafo di scarto, si nota che non esiste nessun cammino elementare tra s e d, per cui l'algoritmo termina ed il flusso massimo risulta essere pari a $v = 8$.

Per trovare il taglio, si deve innanzitutto determinare l'insieme dei nodi S. Considerando il grafo di scarto relativo al flusso ottimo, S è pari al nodo sorgente piú tutti i nodi che, sull'ultimo grafo di scarto, sono raggiungibili dalla sorgente. Nei nostro caso si ottiene:

$$S = \{s, 1\}$$

$$N \setminus S = \{2, 3, 4, 5, 6, d\}.$$

Il taglio di capacitá minima H_s è indicato nella figura seguente.



3.5 Matching su grafi bipartiti

3.5.1 Richiami teorici

Definizione 3.8 (Grafo bipartito $G = (N = N_1 \cup N_2, A)$) Un grafo si dice *bipartito* se l'insieme dei nodi può essere partizionato in due sottoinsiemi N_1 e N_2, $N_1 \cap N_2 = \{\emptyset\}$ tali per cui esiste un arco tra due nodi se solo se uno dei nodi appartiene a N_1 e l'altro nodo ad N_2.

Definizione 3.9 (Matching di un grafo $G = (N, A)$) Il *matching* di un grafo è un sottoinsieme M di archi tali per cui nessuna coppia di essi condivida lo stesso nodo.

Definizione 3.10 (Matching completo su grafo bipartito) Il *matching completo su grafo bipartito* è un matching M il cui numero di archi (cardinalità) è $|M| = \min\{|N_1|, |N_2|\}$.

Definizione 3.11 (Arco accoppiato) Un *arco* è *accoppiato* se è appartenente al matching M.

Definizione 3.12 (Arco libero) Un *arco* è *libero* se non è accoppiato.

Definizione 3.13 (Vertice esposto) Un *vertice esposto* è un vertice non incidente con archi accoppiati.

Definizione 3.14 (Vertice accoppiato) Un *vertice accoppiato* è un vertice non esposto, cioè incidente con un arco accoppiato.

Definizione 3.15 (Cammino alternante) Un *cammino alternante* è un cammino $p = [u_1, u_2, ..., u_{2k}]$ tale per cui gli archi $[u_1, u_2] ... [u_{2k-1}, u_{2k}]$ siano liberi e gli archi $[u_2, u_3] ... [u_{2k-2}, u_{2k-1}]$ siano accoppiati.

Definizione 3.16 (Cammino aumentante) Un *cammino aumentante* è un cammino $p = [u_1, u_2, ..., u_{2k}]$ alternante dove u_1 e u_{2k} sono esposti.

Teorema 3.17 Un matching M in un grafo $G = (N, A)$ è massimo se e solo se non esiste alcun cammino aumentante in $G = (N, A)$ rispetto a M.

Nel seguito si suppone di etichettare un nodo con una coppia di valori L/P. L rappresenta lo stato del nodo e può assumere i seguenti valori:

- U : non etichettato;
- P : pari;
- D : dispari;

- Q : ultimo nodo di un cammino aumentante.

P, invece, rappresenta il nodo predecessore nel cammino aumentante. Il valore $-$ indica che il nodo non ha predecessori.
L'algoritmo per ricavare il massimo matching su un grafo bipartito è riassunto nello pseudocodice seguente.

Function BipartiteMatching
$G(N, A)$: grafo bipartito
M : matching sul grafo bipartito
P : cammino alternante

$M = \{\emptyset\}$
for all $p \in N$ **do**
  if p non è accoppiato **then**
    $BipartiteMatchingSearch(p, found, P)$
    if $found =$ **true then**
      $M = M \oplus P$
    else
      Si cancellano p e tutti gli archi incidenti su di lui
      $N = N \setminus \{p\}$
      $A = A \setminus \{a_{ij} : i = p\}$
    end if
  end if
end for

Function BipartiteMatchingSearch
p : nodo da cui partire nella ricerca
$label(i)$: etichetta nodo i
$LIST$: lista FIFO di nodi

$P = \{\emptyset\}$
$found =$ **false**
for all $i \in N$ **do**
  $label(i) = U/-$
end for
$label(p) = P/-,\ LIST = \{p\}$
while $LIST \neq \emptyset$ **do**
  Estrarre il primo nodo i da $LIST$
  if $label(i) = P$ **then**
    $found = BipartiteMatchingExamineEven(i, found, P)$
  else
    $found = BipartiteMatchingExamineOdd(i, found, P)$

```
    end if
    if found = true then

        return  found
    end if

    return  found
end while
```

Function BipartiteMatchingExamineEven
i : nodo etichettato come P
$label(i)$: etichetta nodo i
$LIST$: lista FIFO di nodi

for all $j \in \Gamma_i$ **do**
  if j non è accoppiato **then**
    $label(j) = Q/i$
    $found =$ **true**

    return $found$
    $P = P \cup \{(i, j)\}$
  else
    if $label(j) = U/-$ **then**
      $label(j) = D/i$
      $LIST = LIST \cup \{j\}$
      $P = P \cup \{(i, j)\}$
    end if
  end if
end for

return $found$

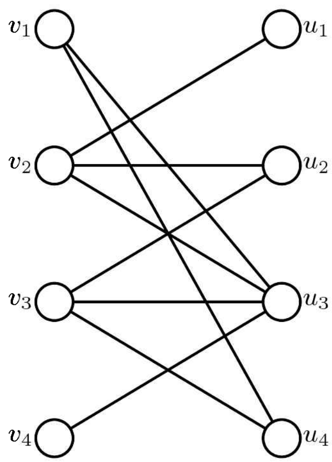
Function BipartiteMatchingExamineOdd
i : nodo etichettato come D
$label(i)$: etichetta nodo i
$LIST$: lista FIFO di nodi

j nodo accoppiato con i nel matching M
if $label(j) = U/-$ **then**
  $label(j) = P/i$
  $LIST = LIST \cup \{j\}$

$P = P \cup \{(i, j)\}$
end if

3.5.2 Esercizi risolti

ESEMPIO 3.14.
Si trovi il massimo matching per il seguente grafo bipartito.



Soluzione

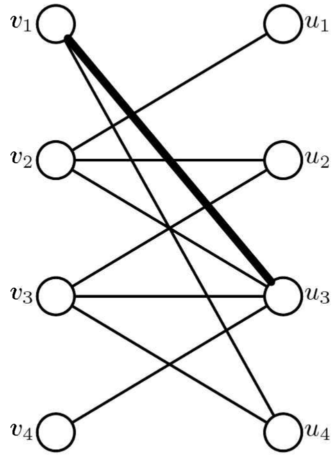
1. Effettuiamo la procedura di etichettamento sul grafo dato, ottenendo la tabella seguente.

v_1	v_2	v_3	v_4	u_1	u_2	u_3	u_4
$P/-$						Q/v_1	

List
$\cancel{v_1}$
u_3

Visto che il nodo u_3 non appartiene al matching, abbiamo trovato un cammino aumentante. Per mezzo delle etichette in tabella, possiamo dire che il cammino è $P = \{(v_1, u_3)\}$ ed il nuovo matching $M = \{(v_1, u_3)\}$.

2. Il nuovo matching è rappresentato in figura.



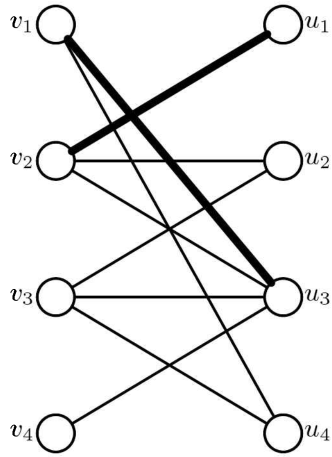
Effettuiamo la procedura di etichettamento sul grafo dato, ottenendo la tabella seguente.

v_1	v_2	v_3	v_4	u_1	u_2	u_3	u_4
	$P/-$			Q/v_2			

List
~~v_2~~
u_1

Visto che il nodo u_1 non appartiene al matching, abbiamo trovato un cammino aumentante. Per mezzo delle etichette in tabella, possiamo dire che il cammino è $P = \{(v_2, u_1)\}$ ed il nuovo matching $M = \{(v_1, u_3), (v_2, u_1)\}$.

3. Il nuovo matching è rappresentato in figura.



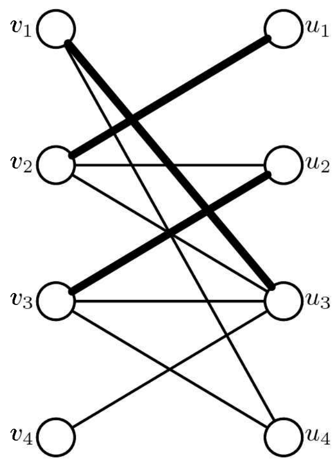
Effettuiamo la procedura di etichettamento sul grafo dato, ottenendo la tabella seguente.

v_1	v_2	v_3	v_4	u_1	u_2	u_3	u_4
		$P/-$			Q/v_3		

List
$\cancel{v_3}$
u_2

Visto che il nodo u_2 non appartiene al matching, abbiamo trovato un cammino aumentante. Per mezzo delle etichette in tabella, possiamo dire che il cammino è $P = \{(v_3, u_2)\}$ ed il nuovo matching $M = \{(v_1, u_3), (v_2, u_1), (v_3, u_2)\}$.

4. Il nuovo matching è rappresentato in figura.

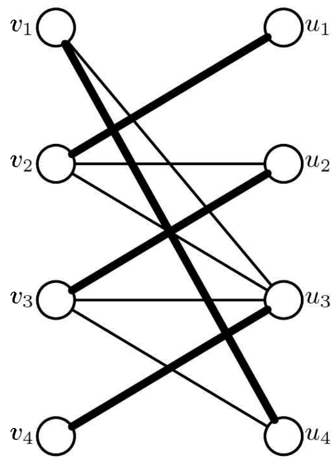


Effettuiamo la procedura di etichettamento sul grafo dato, ottenendo la tabella seguente.

v_1	v_2	v_3	v_4	u_1	u_2	u_3	u_4
P/u_3			$P/-$			D/v_4	Q/v_1

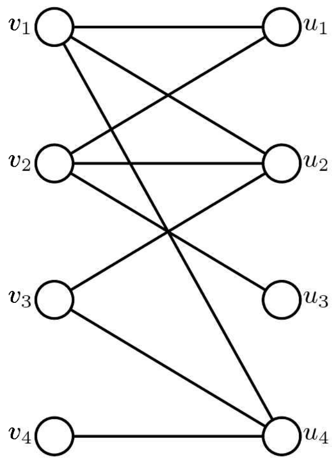
List
$\cancel{v_4}$
$\cancel{u_3}$
$\cancel{v_1}$
u_4

Visto che il nodo u_4 non appartiene al matching, abbiamo trovato un cammino aumentante. Per mezzo delle etichette in tabella, possiamo dire che il cammino è $P = \{(v_4, u_3), (u_3, v_1), (v_1, u_4)\}$. Aggiornando il matching, otteniamo il nuovo matching $M = \{(v_1, u_4), (v_2, u_1), (v_3, u_2), (v_4, u_3)\}$.
Visto che $min\{|N_1|, |N_2|\} = 4 = |M|$, possiamo dire di aver raggiunto l'ottimo, rappresentato nel grafo seguente.



.

ESEMPIO 3.15. Si trovi il massimo matching per il seguente grafo bipartito.



Soluzione

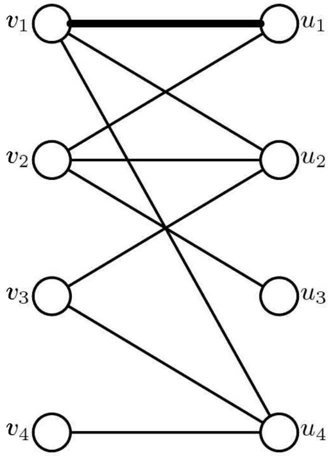
1. Effettuiamo la procedura di etichettamento sul grafo dato, ottenendo la tabella seguente.

v_1	v_2	v_3	v_4	u_1	u_2	u_3	u_4
$P/-$				Q/v_1			

List
$\cancel{v_1}$
u_1

Visto che il nodo u_1 non appartiene al matching, abbiamo trovato un cammino aumentante. Per mezzo delle etichette in tabella, possiamo dire che il cammino è $P = \{(v_1, u_1)\}$ ed il nuovo matching $M = \{(v_1, u_1)\}$.

2. Il nuovo matching è rappresentato in figura.



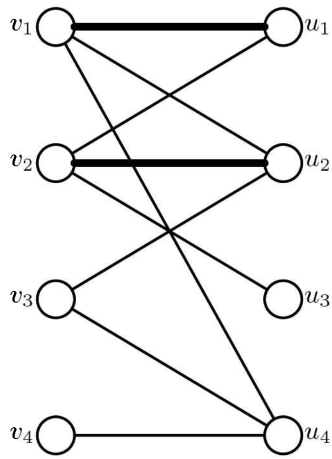
Effettuiamo la procedura di etichettamento sul grafo dato, ottenendo la tabella seguente.

v_1	v_2	v_3	v_4	u_1	u_2	u_3	u_4
	$P/-$			D/v_2	Q/v_2		

List
$\cancel{v_2}$
u_1
u_2

Visto che il nodo u_2 non appartiene al matching, abbiamo trovato un cammino aumentante. Per mezzo delle etichette in tabella, possiamo dire che il cammino è $P = \{(v_2, u_2)\}$ ed il nuovo matching $M = \{(v_1, u_1), (v_2, u_2)\}$.

3. Il nuovo matching è rappresentato in figura.



Effettuiamo la procedura di etichettamento sul grafo dato, ottenendo la tabella seguente.

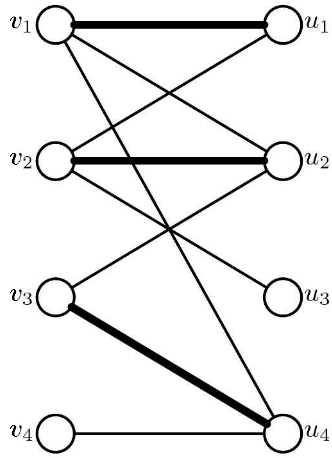
v_1	v_2	v_3	v_4	u_1	u_2	u_3	u_4
		$P/-$			D/v_3		Q/v_3

List

$\cancel{v_3}$
u_2
u_4

Visto che il nodo u_4 non appartiene al matching, abbiamo trovato un cammino aumentante. Per mezzo delle etichette in tabella, possiamo dire che il cammino è $P = \{(v_3, u_4)\}$ ed il nuovo matching $M = \{(v_1, u_1), (v_2, u_2), (v_3, u_4)\}$.

4. Il nuovo matching è rappresentato in figura.



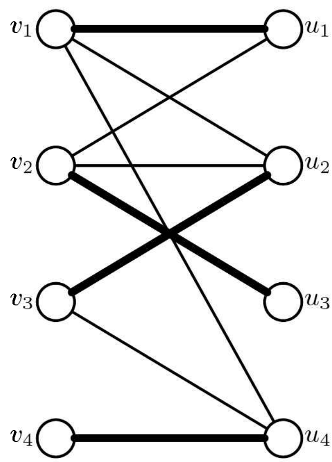
Effettuiamo la procedura di etichettamento sul grafo dato, ottenendo la tabella seguente.

v_1	v_2	v_3	v_4	u_1	u_2	u_3	u_4
	P/u_2	P/u_4	$P/-$	D/v_2	D/v_3	Q/v_2	D/v_4

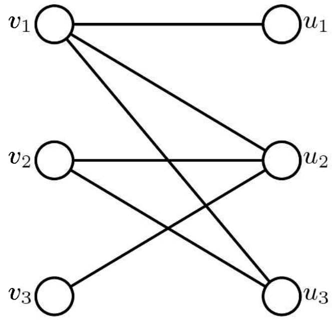
List

$\cancel{v_4}$
$\cancel{u_4}$
$\cancel{v_3}$
$\cancel{u_2}$
$\cancel{v_2}$
u_1
u_3

Visto che il nodo u_3 non appartiene al matching, abbiamo trovato un cammino aumentante. Per mezzo delle etichette in tabella, possiamo dire che il cammino è $P = \{(v_4, u_4), (u_4, v_3), (v_3, u_2), (u_2, v_2), (v_2, u_3)\}$. Il nuovo matching è $M = \{(v_1, u_1), (v_2, u_3), (v_3, u_2), (v_4, u_4)\}$.
Essendo $min\{|N_1|, |N_2|\} = 4 = |M|$, possiamo dire di aver raggiunto l'ottimo, rappresentato nel grafo seguente.



ESEMPIO 3.16. Si trovi il massimo matching per il seguente grafo bipartito.



Soluzione

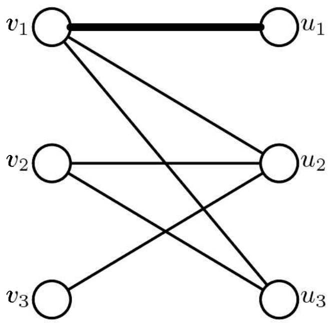
1. Effettuiamo la procedura di etichettamento sul grafo dato, ottenendo la tabella seguente.

v_1	v_2	v_3	u_1	u_2	u_3
$P/-$			Q/v_1		

List
~~v_1~~
u_1

Visto che il nodo u_1 non appartiene al matching, abbiamo trovato un cammino aumentante. Per mezzo delle etichette in tabella, possiamo dire che il cammino è $P = \{(v_1, u_1)\}$ ed il nuovo matching $M = \{(v_1, u_1)\}$.

2. Il nuovo matching è rappresentato in figura.



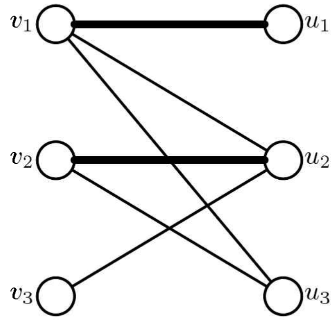
Effettuiamo la procedura di etichettamento sul grafo dato, ottenendo la tabella seguente.

v_1	v_2	v_3	u_1	u_2	u_3
	$P/-$			Q/v_2	

List
~~v_2~~
u_2

Visto che il nodo u_2 non appartiene al matching, abbiamo trovato un cammino aumentante. Per mezzo delle etichette in tabella, possiamo dire che il cammino è $P = \{(v_2, u_2)\}$ ed il nuovo matching $M = \{(v_1, u_1), (v_2, u_2)\}$.

3. Il nuovo matching è rappresentato in figura.



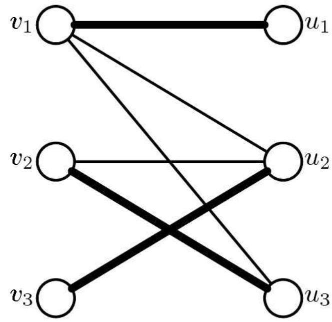
Effettuiamo la procedura di etichettamento sul grafo dato, ottenendo la tabella seguente.

v_1	v_2	v_3	u_1	u_2	u_3
	P/u_2	$P/-$		D/v_3	Q/v_2

List

~~v_3~~
~~u_2~~
~~v_2~~
u_3

Visto che il nodo u_3 non appartiene al matching, abbiamo trovato un cammino aumentante. Per mezzo delle etichette in tabella, possiamo dire che il cammino è $P = \{(v_3, u_2), (u_2, v_2), (v_2, u_3)\}$ ed il nuovo matching $M = \{(v_1, u_1), (v_2, u_3), (v_3, u_2)\}$. Essendo $min\{|N_1|, |N_2|\} = 3 = |M|$, possiamo dire di aver raggiunto l'ottimo, rappresentato nel grafo seguente.



3.6 Metodo del Cammino Critico (CPM) per Project Scheduling

3.6.1 Richiami teorici

Si supponga di avere un progetto del quale si conosce la suddivisione in task ed i vincoli di sequenziamento tra i diversi task. Il problema del cammino critico (CPM) consiste nel trovare l'ordine dei task che minimizzi il tempo di realizzo dell'intero progetto. Si supponga, per ogni task, di conoscere la sua durata e l'elenco dei nodi predecessori. Si supponga di rappresentare il progetto tramite un grafo orientato i cui vertici siano i task e gli archi rappresentino i vincoli di sequenziamento (un arco tra due nodi esiste se e solo se i è nella lista dei predecessori di j). Tale grafo è sicuramente aciclico, infatti, se esistesse un ciclo, ogni task del ciclo dovrebbe precedere se stesso. Data l'aciclicità del grafo, esistono alcuni nodi senza predecessori ed altri che non fanno parte di nessuna lista dei predecessori.
Il progetto può essere rappresentato tramite un grafo orientato, detto *Grafo dei potenziali di task*.

Definizione 3.18 (Grafo dei potenziali di task) Si consideri un progetto caratterizzato da un insieme dei task $T = \{t_1, ..., t_k\}$. Per ogni task t_i si conosce la lista dei task predecessori Γ_i^{-1}. Il *Grafo dei potenziali di task* è un grafo orientato $G(V, A)$ nel quale l'insieme dei vertici $V = T \cup \{\alpha, \omega\}$ è formato dai task del progetto più due task fittizi α ed ω. Un arco (i, j) esiste tra due nodi se $i \in \Gamma_j^{-1}$ e la lunghezza dell'arco è pari a d_i, la durata del task i. Per definizione, α ha durata nulla, è il predecessore di tutti i nodi senza predecessori nel progetto ed ha la lista dei predecessori vuota. Il task fittizio ω ha nella sua lista dei predecessori tutti i nodi che non compaiono in nessuna lista dei predecessori del progetto e non compare come predecessore di nessun nodo del progetto.

Definizione 3.19 (Tempo di inizio al più presto) Dato un progetto rappresentato tramite il un grafo dei potenziali di task, si definisce il *tempo di inizio al più presto* del task i la quantità

$$t_i = \max_{j \in \Gamma_i^{-1}} (t_j + d_j) \,.$$

Il valore di t_i coincide con il cammino massimo $l(\alpha, i)$ da α ad i.

Definizione 3.20 (Tempo di inizio al più tardi) Dato un progetto rappresentato tramite il un grafo dei potenziali di task, si definisce il *tempo di inizio al più tardi* del task i la quantità

$$T_i = \min_{j \in \Gamma_i} T_j - d_i = T_\omega - l\,(i, \omega) = t_\omega - l\,(i, \omega)\,,$$

ove $l(i, \omega)$ è la lunghezza del cammino massimo da i a ω e $t_\omega = T_\omega$.

Definizione 3.21 (Slack di un task) Definiamo *slack* del task i la quantità

$$m_i = T_i - t_i.$$

Un task avente slack nullo è detto *critico*, in quanto un ritardo sul tempo di completamento del task pari a Δ provocherà un aumento del tempo di completamento dell'intero progetto pari a Δ.

Definizione 3.22 (Cammino critico di un progetto) Il *cammino critico di un progetto* è un cammino esistente tra α ed ω composto da soli task critici.

Function CPM per il Project Scheduling
$N = T \cup \{\alpha, \omega\}, A = \{\emptyset\}$
for $(i \in T)$ **do**
    for $(j \in T \,:\, j \in \Gamma_i^1)$ **do**
        $A = A \cup \{(i,j)\}$, $l_{ij} = d_i$
    end for
end for
for $(i \in T \,:\, \Gamma_i = \{\emptyset\})$ **do**
    $A = A \cup \{(i,\omega)\}$, $l_{i\omega} = d_i$
end for
for $(i \in T \,:\, \Gamma_i^{-1} = \{\emptyset\})$ **do**
    $A = A \cup \{(\alpha, i)\}$, $l_{\alpha i} = 0$
end for
Calcolare i rank per i nodi del grafo $G = (N, A)$. Sia $r_{max}(G)$ il massimo valore di rank assunto dai nodi di G e r_i il valore del rank di un singolo nodo.
$r = 0$, $t_\alpha = 0$
while $(r \leq r_{max}(G))$ **do**
    for $(i \in T) \,:\, r_i = r$ **do**
        $t_i = \max_{j \in \Gamma_i^{-1}} \{t_j + d_j\}$
    end for
    $r = r + 1$
end while
$r = r_{max}(G)$, $T_\omega = t_\omega$
while $(r \neq 0)$ **do**
    for $(i \in T) \,:\, r_i = r$ **do**
        $T_i = \min_{j \in \Gamma_i} \{T_j\} - d_i$
    end for
    $r = r - 1$
end while

3.6.2 Esercizi risolti

ESEMPIO 3.17. Si consideri il progetto i cui dati sono riportati in tabella.

Attività	Durata	Attività precedenti
A	2	-
B	3	-
C	4	-
D	6	A, B
E	2	B, C
F	3	A, D
G	4	D, E
H	5	E

Rispondere ai seguenti quesiti:

- determinare la durata minima del progetto, il cammino critico, i tempi di inizio al più presto ed al più tardi di tutte le attività;
- dire di quanto debba diminuire la durata delle attività critiche affinché non siano più critiche;
- dire di quanto debba aumentare la durata delle attività non critiche affinché siano critiche.

Soluzione
La prima cosa da fare è modificare il progetto nella seguente maniera:

- aggiungiamo un'attività α all'inizio del progetto di durata 0 che abbia come successori le attività senza predecessori nel progetto originale;
- aggiungiamo un'attività ω al termine del progetto che abbia come predecessori le attività senza successori nel progetto originale.

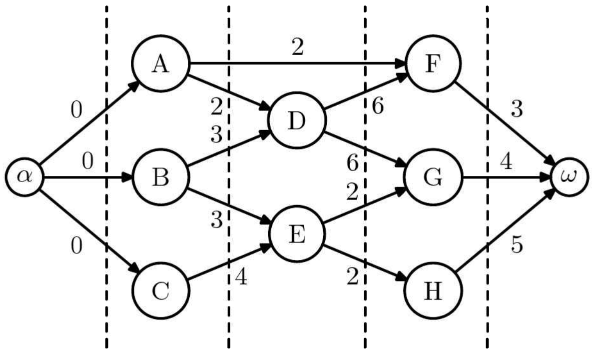
La tabella riassuntiva delle attività diventa quindi la seguente.

Attività	Durata	Attività precedenti
α	0	-
A	2	α
B	3	α
C	4	α
D	6	A, B
E	2	B, C
F	3	A, D
G	4	D, E
H	5	E
ω	-	F, G, H

A partire da questa nuova struttura del progetto, possiamo determinare i valori dei rank delle attività.

Iter	α	A	B	C	D	E	F	G	H	ω	r_i
0	0	1	1	1	2	2	2	2	1	3	α
1		0	0	0	2	2	2	2	1	3	A, B, C
2					0	0	1	2	1	3	D, E
3							0	0	0	3	F, G, H
4										0	ω

Avendo determinato i valori dei rank delle attività, possiamo rappresentare l'intero progetto tramite un grafo disegnato per rank crescenti. Tale grafo ha per nodi le attività del progetto. Dati due nodi i e j, nel grafo esiste un arco (i, j) se e solo se i è un predecessore di j. Ad ogni arco (i, j) viene associato un costo pari alla durata dell'attività i. Per il progetto in esame, si ottiene il grafo riportato di seguito.



Avendo determinato i valori dei rank per le attività del progetto, possiamo ora calcolare i tempi di inizio al più presto t_i ed al più tardi T_i. Per determinare i t_i, fissiamo $t_\alpha = 0$ e, considerando le attività per valori crescenti dei rank, determiniamo iterativamente i rimanenti valori tramite la formula $t_i = \max_{j \in \Gamma_i^{-1}}\{t_j + d_j\}$.

$t_A = \max\{t_\alpha + d_\alpha\} = 0$ $\quad$ $t_B = \max\{t_\alpha + d_\alpha\} = 0$
$t_C = \max\{t_\alpha + d_\alpha\} = 0$ $\quad$ $t_D = \max\{t_A + d_A, t_B + d_B\} = 3$
$t_E = \max\{t_B + d_B, t_C + d_C\} = 4$ $\quad$ $t_F = \max\{t_A + d_A, t_D + d_D\} = 9$
$t_G = \max\{t_D + d_D, t_E + d_E\} = 9$ $\quad$ $t_H = \max\{t_E + d_E\} = 6$
$t_\omega = \max\{t_F + d_F, t_G + d_G, t_H + d_H\} = 13.$

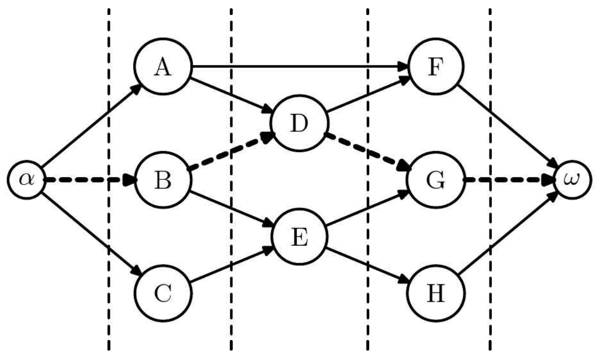
Per ottenere i valori dei tempi di inizio al più tardi T_i, invece, viene utilizzata iterativamente la formula $T_i = \min_{j \in \Gamma_i}\{T_j\} - d_i$, avendo posto $T_\omega = t_\omega = 13$ e considerando le attività per valori decrescenti dei rank.

$T_H = \min\{T_\omega\} - d_H = 8 \qquad T_G = \min\{T_\omega\} - d_G = 9$
$T_F = \min\{T_\omega\} - d_F = 10 \qquad T_E = \min\{T_G, T_H\} - d_E = 6$
$T_D = \min\{T_F, T_G\} - d_D = 3 \qquad T_C = \min\{T_E\} - d_C = 2$
$T_B = \min\{T_D, T_E\} - d_B = 0 \qquad T_A = \min\{T_D\} - d_A = 1.$

Avendo ottenuto i valori di t_i e T_i, possiamo calcolare lo slack di ogni attività come $m_i = T_i - t_i$. La tabella seguente riporta, per tutte le attività del progetto, i valori di t_i, T_i e m_i.

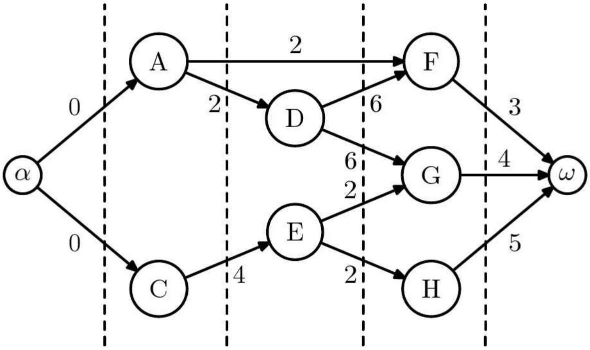
	α	A	B	C	D	E	F	G	H	ω
t_i	0	0	0	0	3	4	9	9	6	13
T_i	0	1	0	2	3	6	10	9	8	13
m_i	0	1	0	2	0	2	1	0	2	0

La durata minima del progetto è pari a $t_\omega = T_\omega = 13$, mentre le attività critiche sono quelle per le quali lo slack è nullo. Il cammino critico è il cammino tra α ed ω che si ottiene considerando solo le attività critiche e gli archi che le interconnettono. Nel caso in esame il cammino critico è $B/D/G$ (vedi gli archi tratteggiati nella figura seguente).



Analizziamo ora le attività critiche in modo da determinare se possano diventare non critiche e, se sì, per quali valori delle durate.

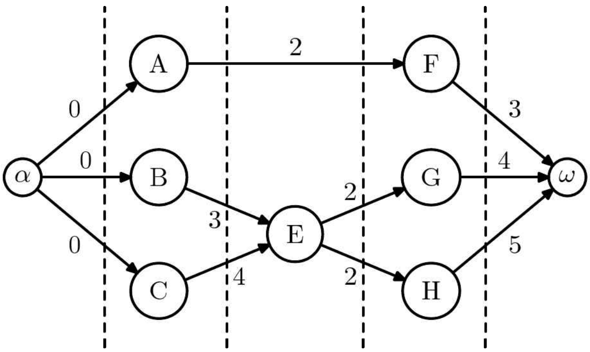
Consideriamo quindi l'attività B. Se tale attività non fosse critica e le altre attività rimanessero immutate, sarebbe come se B non fosse mai stata presente nel progetto, ottenendo, per questo nuovo progetto, il grafo in figura.



Ricalcoliamo i valori dei tempi di inizio al più presto t_i al fine di ricavare la durata minima del nuovo progetto.

	α	A	C	D	E	F	G	H	ω
t'_i	0	0	0	2	4	8	8	6	12

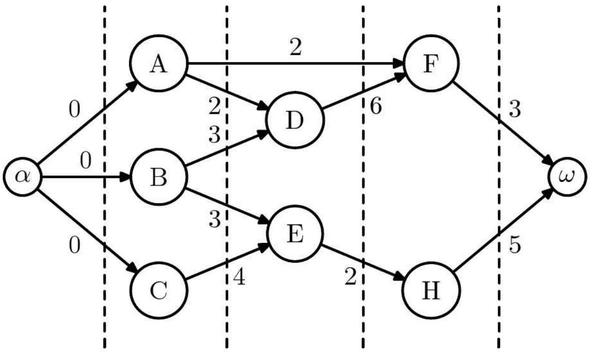
La durata del nuovo progetto è quindi $t'_\omega = 12$, mentre la durata dell'attività B dovrebbe, al fine di diventare non critica, diminuire almeno della quantità $\Delta_B = t\omega - t'\omega + 1 = 2$. Visto che $d_B > \Delta_B$, l'attività B può diventare non critica se la sua durata diminuisce almeno di Δ_B.
Consideriamo ora l'attività D. Eliminando tale attività otteniamo il grafo seguente.



Ricalcoliamo i valori dei tempi di inizio al più presto t_i al fine di ricavare la durata minima del nuovo progetto.

	α	A	B	C	E	F	G	H	ω
t'_i	0	0	0	0	4	2	6	6	11

La durata del nuovo progetto è quindi $t'_\omega = 11$, mentre la durata dell'attività D dovrebbe, al fine di diventare non critica, diminuire almeno della quantità $\Delta_D = t\omega - t'\omega + 1 = 3$. Visto che $d_D > \Delta_D$, l'attività D può diventare non critica se la sua durata diminuisce almeno di Δ_D.
Consideriamo infine l'attività G. Eliminando tale attività dal progetto otteniamo il grafo seguente.



Ricalcoliamo i valori dei tempi di inizio al più presto t_i al fine di ricavare la durata minima del nuovo progetto.

	α	A	B	C	D	E	F	H	ω
t'_i	0	0	0	0	3	4	9	6	12

La durata del nuovo progetto è quindi $t'_\omega = 12$, mentre la durata dell'attività G dovrebbe, al fine di diventare non critica, diminuire almeno della quantità $\Delta_G = t\omega - t'\omega + 1 = 2$. Visto che $d_G > \Delta_G$, l'attività G può diventare non critica se la sua durata diminuisce almeno di Δ_G.
Consideriamo ora le attività non critiche. Affinché esse diventino critiche, la loro durata deve aumentare almeno del valore dello slack. Data, cioè, un'attività non critica, la durata minima per la quale essa diventa critica è $d'_i = d_i + m_i$.
Nel caso specifico, quindi, avremo che l'attività A diventa non critica se la sua durata vale almeno $d'_A = d_A + m_A = 3$. Analogamente, per le altre attività non critiche otterremo i valori seguenti:

$$d'_C = d_C + m_C = 6, \quad d'_E = d_E + m_E = 4,$$
$$d'_F = d_F + m_F = 4, \quad d'_H = d_H + m_H = 7.$$

ESEMPIO 3.18. Si consideri il progetto i cui dati sono riportati in tabella.

Attività	Durata	Attività precedenti
A	2	-
B	3	-
C	4	A, B
D	5	A, C
E	2	B, C
F	4	B
G	6	B, D, E, F

Rispondere ai seguenti quesiti:

- determinare la durata minima del progetto, il cammino critico, i tempi di inizio al più presto ed al più tardi di tutte le attività;
- discutere l'impatto sul progetto di una variazione della durata dell'attività B;
- discutere l'impatto sul progetto di una variazione della durata dell'attività F.

Soluzione
Modifichiamo il progetto nella seguente maniera:

- aggiungiamo un'attività α all'inizio del progetto di durata 0 che abbia come successori le attività senza predecessori nel progetto originale;
- aggiungiamo un'attività ω al termine del progetto che abbia come predecessori le attività senza successori nel progetto originale.

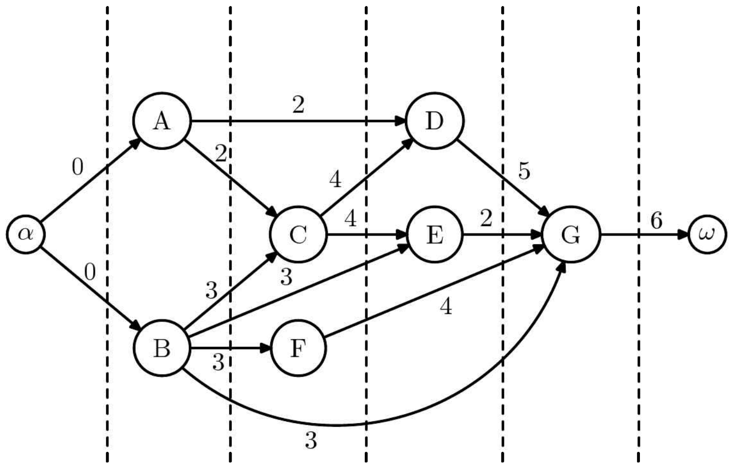
La tabella riassuntiva delle attività diventa la seguente.

Attività	Durata	Attività precedenti
A	2	α
B	3	α
C	4	A, B
D	5	A, C
E	2	B, C
F	4	B
G	6	B, D, E, F
ω	-	G

A partire da questa nuova struttura del progetto, possiamo determinare i valori dei rank delle attività.

Iter	α	A	B	C	D	E	F	G	ω	r_i
0	0	1	1	2	2	2	1	4	1	α
1		0	0	2	2	2	1	4	1	A, B
2				0	1	1	0	3	1	C, F
3					0	0		2	1	D, E
4								0	1	G
4									0	ω

Avendo determinato i valori dei rank delle attività, possiamo rappresentare l'intero progetto tramite un grafo. Disegnando il grafo per valori crescenti dei rank, si ottiene il grafo riportato di seguito.



Avendo determinato i valori dei rank per le attività del progetto, possiamo ora calcolare i tempi di inizio al più presto t_i ed al più tardi T_i. Per determinare i t_i, fissiamo $t_\alpha = 0$ e, considerando le attività per valori crescenti dei rank, determiniamo iterativamente i rimanenti valori tramite la formula $t_i = \max_{j \in \Gamma_i^{-1}}\{t_j + d_j\}$.

$t_A = \max\{t_\alpha + d_\alpha\} = 0$ $\quad$ $t_B = \max\{t_\alpha + d_\alpha\} = 0$
$t_C = \max\{t_A + d_A, t_B + d_B\} = 3$ $\quad$ $t_F = \max\{t_B + d_B\} = 3$
$t_D = \max\{t_A + d_A, t_C + d_C\} = 7$ $\quad$ $t_E = \max\{t_B + d_B, t_C + d_C\} = 7$
$t_G = \max\{t_B + d_B, t_D + d_D, t_E + d_E, t_F + d_F\} = 12$ $\quad$ $t_\omega = \max\{t_G + d_G\} = 18$.

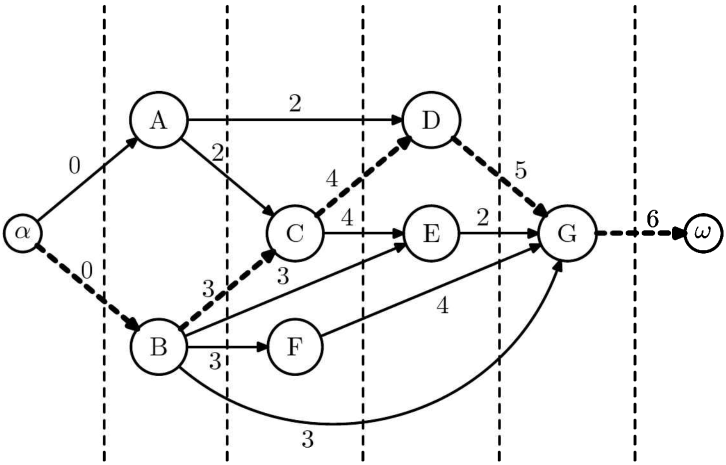
Per ottenere i valori dei tempi di inizio al più tardi T_i, invece, viene utilizzata iterativamente la formula $T_i = \min_{j \in \Gamma_i}\{T_j\} - d_i$, avendo posto $T_\omega = t_\omega = 18$ e considerando le attività per valori decrescenti dei rank.

$T_G = \min\{T_\omega\} - d_G = 12$ $\quad$ $T_E = \min\{T_\omega\} - d_E = 10$
$T_D = \min\{T_\omega\} - d_D = 7$ $\quad$ $T_F = \min\{T_G\} - d_F = 8$
$T_C = \min\{T_D, T_E\} - d_C = 3$ $\quad$ $T_B = \min\{T_C, T_E, T_F\} - d_B = 0$
$T_A = \min\{T_C, T_D\} - d_A = 1$.

Avendo ottenuto i valori di t_i e T_i, possiamo calcolare lo slack di ogni attività come $m_i = T_i - t_i$. La tabella seguente riporta, per tutte le attività del progetto, i valori di t_i, T_i e m_i.

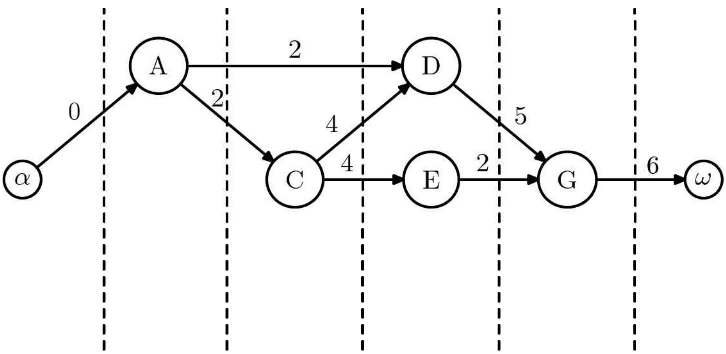
	α	A	B	C	F	D	E	G	ω
t_i	0	0	0	3	3	7	7	12	18
T_i	0	1	0	3	8	7	10	12	18
m_i	0	1	0	0	5	0	3	0	0

La durata minima del progetto è pari a $t_\omega = T_\omega = 18$, mentre le attività critiche sono quelle per le quali lo slack è nullo. Il cammino critico è il cammino tra α ed ω che si ottiene considerando solo le attività critici e gli archi che li interconnettono. Nel caso in esame il cammino critico è $B/C/D/G$ (vedi gli archi tratteggiati nella figura seguente).



Consideriamo ora l'attività B. Tale attività è critica, per cui un aumento della sua durata non farebbe che renderla sempre critica, peggiorando la durata minima del progetto.

Vediamo ora se, diminuendo opportunamente la durata di B, possiamo rendere l'attività non critica. Per far ciò eliminiamo B dal progetto e ricalcoliamone la durata minima. Si noti che, eliminando l'attività B, l'attività F può essere eliminata, visto che aveva come unica precedenza proprio B.



La tabella seguente riporta, per tutte le attività del progetto, i nuovi valori di t_i.

	α	A	C	D	E	G	ω
t'_i	0	0	2	6	6	11	17

L'attività B dovrebbe diminuire di $\delta_B = t_\omega - t'_\omega + 1 = 18 - 17 + 1 = 2$ per diventare un'attività non critica. L'attività può diventare non critica se la sua durata diventa almeno $d'_B = 3 - 2 = 1$.
L'attività F è non critica, per cui rimane tale anche diminuendo la sua durata. Aumentando la durata di F di almeno il suo slack $m_F = 5$, invece, si potrebbe rendere tale attività critica e variare, quindi, la durata minima del progetto.

3.7 Esercizi proposti

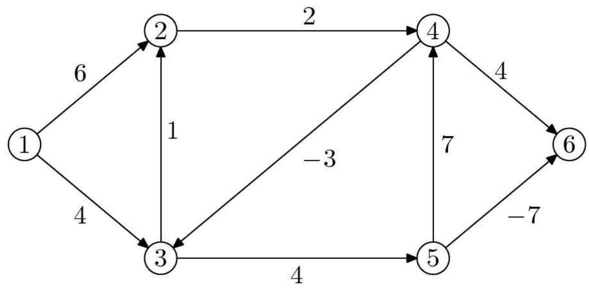
1. Trovare, con l'algoritmo di Bellman, il cammino minimo tra il nodo 1 e tutti gli altri nodi nel grafo dato:

	1	2	3	4	5	6
1	0	10	3	∞	∞	∞
2	∞	0	∞	2	-3	∞
3	∞	1	0	∞	6	∞
4	∞	1	∞	0	∞	4
5	∞	∞	∞	1	0	3
6	∞	∞	∞	∞	∞	0

2. Trovare il cammino minimo tra il nodo 1 ed il nodo 6 del problema indicato in tabella utilizzando l'algoritmo di Dijkstra.

	1	2	3	4	5	6
1	∞	1	2	∞	∞	∞
2	∞	∞	∞	∞	1	∞
3	∞	3	∞	3	∞	∞
4	∞	∞	∞	∞	∞	2
5	∞	∞	∞	2	∞	5
6	∞	∞	∞	∞	∞	∞

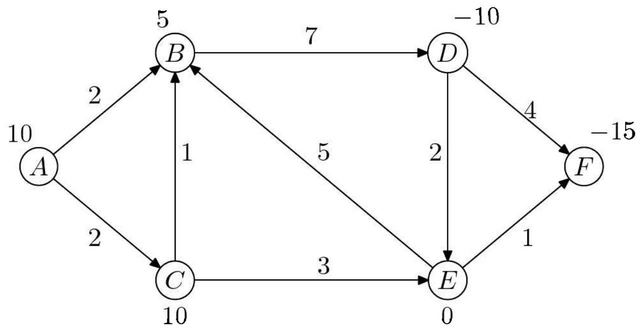
3. Trovare il cammino minimo tra 1 e tutti gli altri nodi nel seguente grafo:



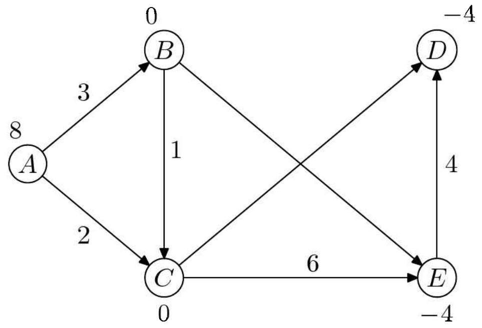
4. Trovare il cammino minimo tra 1 e tutti gli altri nodi nel seguente grafo:

	1	2	3	4	5	6
1	0	5	9	∞	∞	∞
2	∞	0	2	∞	2	∞
3	∞	∞	0	3	∞	∞
4	∞	-4	∞	0	∞	7
5	∞	∞	∞	1	0	8
6	∞	∞	∞	∞	∞	0

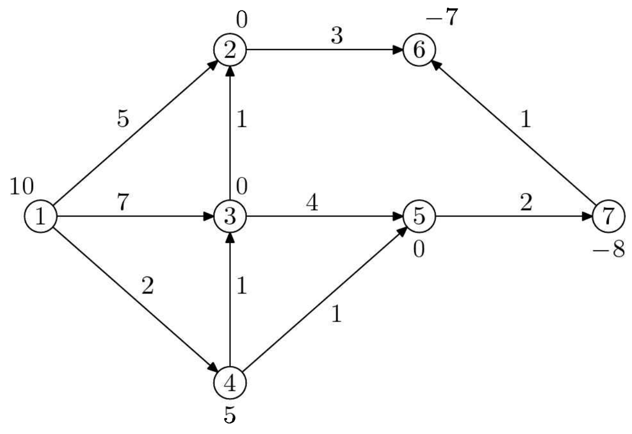
5. Trovare la soluzione ottima e la funzione obiettivo ottima per il problema di flusso di costo minimo rappresentato in figura, in cui agli archi è associato il costo e ai nodi la domanda (< 0) o l'offerta (> 0).



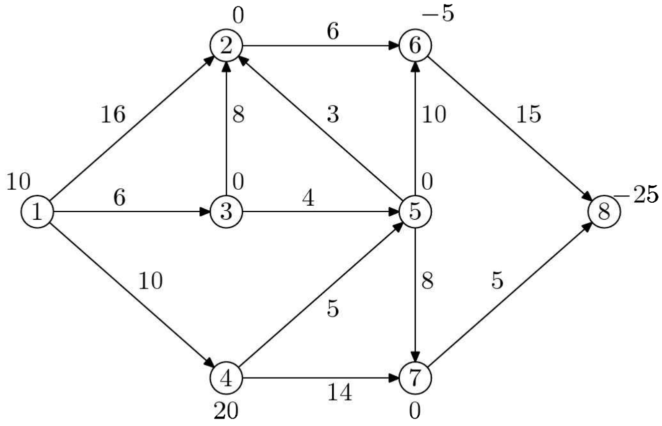
6. Trovare la soluzione ottima e la funzione obiettivo ottima per il problema di flusso di costo minimo rappresentato in figura, in cui agli archi è associato il costo e ai nodi la domanda (< 0) o l'offerta (> 0). L'arco CD ha costo 8, mentre l'arco BE ha costo 6.



7. Trovare la soluzione ottima e la funzione obiettivo ottima per il problema di flusso di costo minimo rappresentato in figura, in cui agli archi è associato il costo e ai nodi la domanda (< 0) o l'offerta (> 0).



8. Trovare la soluzione ottima e la funzione obiettivo ottima per il problema di flusso di costo minimo rappresentato in figura, in cui agli archi è associato il costo e ai nodi la domanda (< 0) o l'offerta (> 0), partendo dalla base: $x_{13} = 10, x_{32} = 0, x_{35} = 10, x_{45} = 20, x_{56} = 5, x_{57} = 25, x_{78} = 25$.



9. Trovare la soluzione ottima e la funzione obiettivo ottima per il seguente problema di flusso di costo minimo partendo dal flusso iniziale indicato:

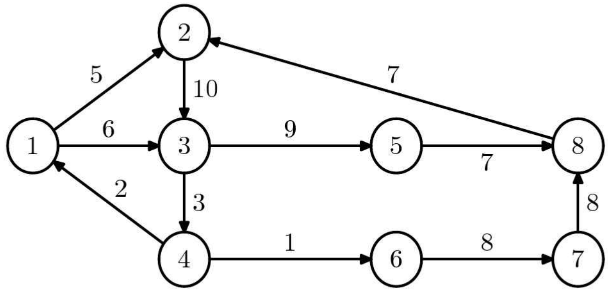
   Nodo A: offerta di 10 unità. Nodo B: offerta di 8 unità.
   Nodo C: domanda di 4 unità. Nodo D: domanda di 7 unità.
   Nodo E: offerta di 1 unità. Nodo F: domanda di 8 unità.

   Arco AB: 4 unità di flusso. Arco AE: 6 unità di flusso.
   Arco BF: 12 unità di flusso. Arco ED: 7 unità di flusso.
   Arco FB: 4 unità di flusso.

   Matrice costi

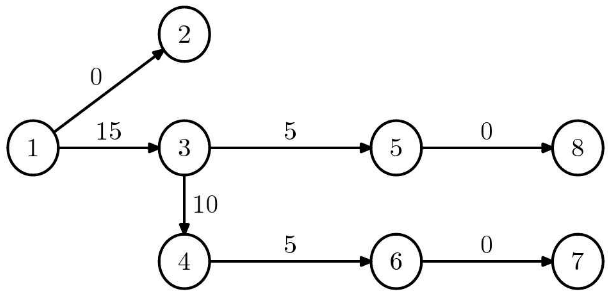
	A	B	C	D	E	F
A	∞	5	∞	∞	4	7
B	∞	∞	2	∞	6	3
C	∞	∞	∞	6	∞	∞
D	∞	∞	∞	∞	∞	∞
E	∞	∞	5	4	∞	∞
F	∞	∞	1	∞	∞	∞

10. Risolvere il seguente problema di flusso di costo minimo.



I valori delle offerte e delle domande di flusso associate agli archi sono le seguenti: $b_1 = 15$, $b_4 = -5$, $b_6 = -5$, $b_5 = -5$.

Si consideri la seguente soluzione ammissibile di base, Dove i numeri presenti sugli archi indicano le quantità di flusso sui rispettivi archi.



11. Dato il seguente problema di flusso massimo dal nodo 1 al nodo 7 con il flusso iniziale indicato fra parentesi nella matrice delle capacità,

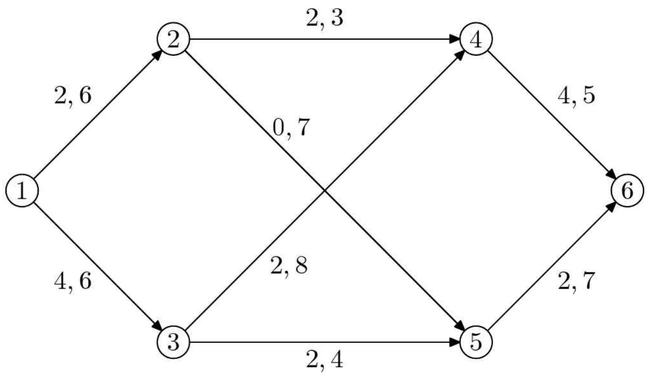
    (a) verificare l'ottimalità della soluzione;

    (b) indicare se sia possibile o meno incrementare il flusso massimo aumentando la capacità dell'arco 6-7; motivare la risposta.

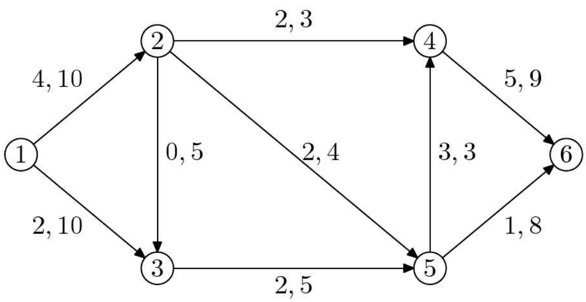
    Matrice delle capacità

	1	2	3	4	5	6	7
1	0	7 (7)	7 (3)	0	0	0	0
2	0	0	0	3 (2)	0	5 (5)	0
3	0	0	0	0	7 (4)	0	0
4	0	0	0	0	0	0	5 (5)
5	0	0	0	4 (3)	0	1 (1)	0
6	0	0	0	0	0	0	6 (6)
7	0	0	0	0	0	0	0

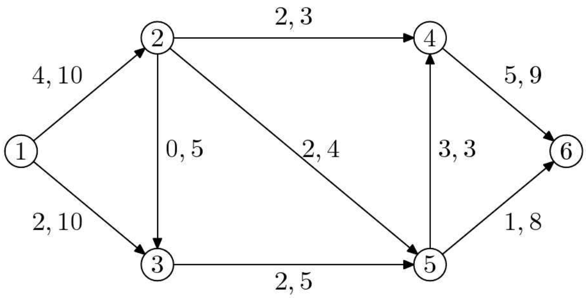
12. Trovare il flusso massimo tra nodo 1 e nodo 6 sul seguente grafo:



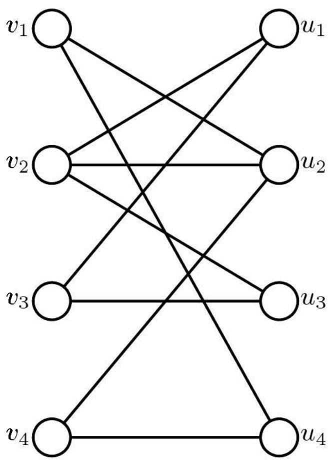
13. Trovare il flusso massimo tra nodo 1 e nodo 6 sul seguente grafo:



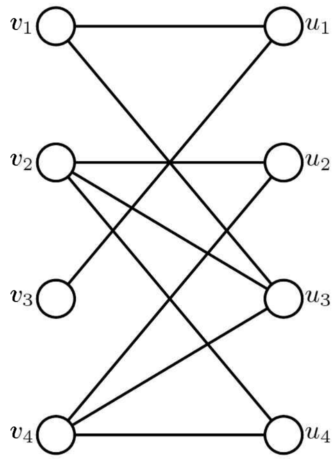
14. Trovare, con l'algoritmo di Ford-Fulkerson, il flusso massimo tra nodo 1 e nodo 6 sul seguente grafo:



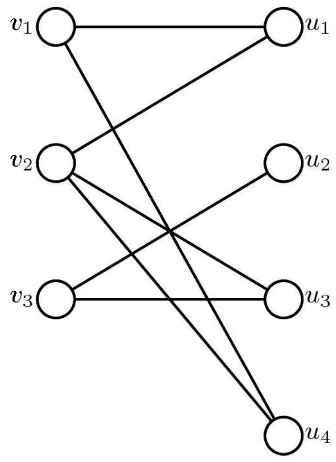
15. Si trovi il massimo matching per il seguente grafo bipartito.



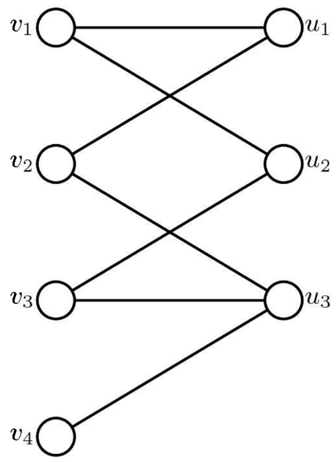
16. Si trovi il massimo matching per il seguente grafo bipartito.



17. Si trovi il massimo matching per il seguente grafo bipartito.



18. Si trovi il massimo matching per il seguente grafo bipartito.


v_1
u_1
v_2
u_2
v_3
u_3
v_4

4

Problemi di allocazione

4.1 Problema dei trasporti

Il problema dei trasporti richiede di inviare opportune quantità di "merce" da un insieme di origini $1, 2, \ldots, m$ a un insieme di destinazioni $1, 2, \ldots, n$ minimizzando il costo totale di trasporto. Sono dati: i costi di trasporto per unità di merce, specificati per ogni coppia origine-destinazione, c_{ij} $i = 1, \ldots, m$, $j = 1, \ldots, m$; le disponibilità di merce ad ogni origine, $a_1, a_2, \ldots, a_m$; le richieste di merce alle destinazioni $b_1, b_2, \ldots, b_n$.
Il problema ammette un modello di programmazione lineare.

$$\min \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{4.1}$$

soggetto a

$$\sum_{j=1}^{n} x_{ij} = a_i \qquad i = 1, \ldots, m, \tag{4.2}$$

$$\sum_{i=1}^{n} x_{ij} = b_j \qquad j = 1, \ldots, n, \tag{4.3}$$

$$x_{ij} \geq 0, \ i = 1, \ldots, m, \ j = 1, \ldots, n. \tag{4.4}$$

Le x_{ij} rappresentano le quantità di merce da inviare dall'origine i alla destinazione j; la funzione obiettivo (4.1) rappresenta il costo totale di trasporto; le serie di vincoli (4.2) e (4.3) impongono rispettivamente che tutte le disponibilità delle sorgenti e tutte le richieste delle destinazioni siano esaurite in ogni soluzione ammissibile.
Il modello assume $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$, cioè la condizione di *problema bilanciato.* Ogni istanza non bilanciata può facilmente essere ricondotta ad un'istanza bilanciata equivalente: $\sum_{i=1}^{m} a_i - \sum_{j=1}^{n} b_j = Q > 0$, si può aggiungere una destinazione "artificiale"

$n+1$, con $b_{n+1} = Q$ e costi di trasporto $c_{i,n+1} = 0$ per ogni $i = 1, \ldots, n$; si può osservare facilmente che l'istanza così ottenuta è equivalente a quella non bilanciata. Se $Q < 0$ occorre aggiungere una sorgente artificiale, in modo simmetrico.
Il problema (4.1)–(4.4) presenta una struttura notevole: esso ha sempre soluzione ottima finita e, poiché i coefficienti dei vincoli formano una matrice totalmente unimodulare, esiste sempre una soluzione ottima a variabili intere se tutti vi valori a_i, b_j sono interi.

Esiste una variante particolarmente semplice dell'algoritmo del simplesso, sviluppata per il problema dei trasporti. L'algoritmo lavora su una tabella di celle $(i,j) \in \{1,\ldots,m\} \times \{1,\ldots,n\}$ che rappresentano le variabili x_{ij}; una base di (4.1)–(4.4) corrisponde ad un insieme B di $m+n-1$ celle per le quali il sistema ristretto

$$\sum_{\substack{j=1 \\ (i,j)\in B}}^{n} x_{ij} = a_i \qquad i = 1, \ldots, m,$$

$$\sum_{\substack{i=1 \\ (i,j)\in B}}^{n} x_{ij} = b_j \qquad j = 1, \ldots, n,$$

ammette soluzione unica.
Pur essendo possibile risolvere il problema dei trasporti mediante tecniche generali di programmazione lineare, il seguente algoritmo, dovuto a Dantzig, risulta essere più efficiente.

Algoritmo del simplesso per il problema dei trasporti.
Determina una soluzione ammissibile $x \in \mathbb{R}^{m\times n}$ corrispondente a una base B
repeat
  Calcola i costi ridotti r_{ij} per ogni $(i,j) \notin B$
  if $\langle r_{ij} < 0$ per qualche $(i,j)\rangle$ **then**
    Scegli $(k,l) \notin B$ con $r_{kl} < 0$
    Determina $(p,q) \in B$ tale che $B \cup \{(k,l)\} \setminus \{(p,q)\}$ sia una base ammissibile
    Poni $B = B \cup \{(k,l)\} \setminus \{(p,q)\}$ e ricalcola la soluzione ammissibile x corrispondente
  end if
until $\langle r_{ij} \geq 0$ per ogni $(i,j) \notin B\rangle$

L'algoritmo presenta la logica classica del simplesso, ma i seguenti tre passi fondamentali sono implementabili con metodi specializzati per il problema del trasporto e altamente efficienti.

- Determinazione di una soluzione ammissibile di base iniziale.

- Calcolo dei costi ridotti.
- Cambio di base.

Determinazione di una soluzione ammissibile iniziale. La determinazione di una soluzione ammissibile di base iniziale può essere svolta per mezzo del seguente

---

Metodo dell'angolo Nord-Ovest.
$x_{ij} := 0$ per ogni (i, j)
$i := j := 1$
while $i < m$ **or** $j < n$ **do**
  $x_{ij} := \min\{a_i, b_j\}$
  $a_i := a_i - x_{ij}$
  $b_j := b_j - x_{ij}$
  if $b_j = 0$ **and** $j < n$ **then**
    $j := j + 1$
  else if $a_i = 0$ **and** $i < m$ **then**
    $i := i + 1$
  end if
end while

Calcolo dei costi ridotti. Il calcolo dei costi ridotti avviene come segue. Si determinano i moltiplicatori del simplesso $u_1, \ldots, u_m$, $v_1, \ldots, v_n$ associati rispettivamente ai vincoli (4.2) e (4.3). Data la base corrente B, un valore dei moltiplicatori è determinato da una soluzione del sistema

$$u_i + v_j = c_{ij} \qquad \text{per ogni } (i, j) \in B, \tag{4.5}$$

Il sistema (4.5) non ha rango pieno, quindi si può scegliere — arbitrariamente — la soluzione con $v_n = 0$. Ottenuti i moltiplicatori, i costi ridotti sono dati da

$$r_{ij} = c_{ij} - (u_i + v_j) \qquad \text{per ogni } (i, j).$$

Cambio di base. Un *ciclo alternante* è una sequenza di celle distinte

$$\left[(i_1, j_1)^{+}_{,} (i_2, j_2)^{-}_{,} \ldots, (i_{2t}, j_{2t})^{-}\right]$$

— le posizioni dispari sono etichettate "+", quelle pari "−" — tale che valga

$$i_r = i_{r+1} \text{ oppure } j_r = j_{r+1} \qquad \text{per } 1 \leq r \leq 2t - 1,$$

e una tra $i_1 = i_{2t}$, $j_1 = j_{2t}$.
La procedura di cambio di base per il problema dei trasporti è la seguente. Sia B una base ammissibile di (4.1)–(4.4) con associata una soluzione ammissibile x, e $(k, l) \notin B$ la cella candidata a entrare in base; l'insieme $B \cup \{(k, l)\}$ contiene un <u>*unico*</u> ciclo

alternante C con (k,l) in prima posizione. La cella uscente (p,q) è scelta in modo tale che

$$x_{pq} = \varepsilon = \min\{(r,s) \in B\colon (r,s) \text{ ha etichetta "}-\text{" in } C\}\,.$$

La base successiva è $B \cup \{(k,l)\} \setminus \{(p,q)\}$, ed i valori di x ad essa associati possono essere ricavati aggiornando le x_{ij} precedenti come segue:

$$x_{ij} = \begin{cases} x_{ij} + \varepsilon, & \text{se } (i,j) \text{ ha etichetta "+"}, \\ x_{ij} - \varepsilon, & \text{se } (i,j) \text{ ha etichetta "}-\text{"}, \\ x_{ij}, & \text{altrimenti.} \end{cases}$$

4.1.1 Esercizi risolti

ESEMPIO 4.1. Si risolva il problema di trasporto associato alla seguente matrice dei costi.

	1	2	3
1	2	2	1
2	3	4	2
3	1	5	3

Le offerte associate alle righe sono rispettivamente $40, 50, 60$; le domande associate alle colonne sono $50, 70, 30$. Risolvere il problema usando l'algoritmo di Dantzig.

Soluzione Il problema è bilanciato ($\sum_i a_i = \sum_j b_j = 150$). Applicando il metodo dell'angolo Nord-Ovest si determina la base iniziale

$$B_0 = \{(1,1), (2,1), (2,2), (3,2), (3,3)\}$$

con la relativa soluzione rappresentata nella tabella x_{ij}.

	1	2	3	a_i
1	40			40
2	10	40		50
3		30	30	60
b_j	50	70	30	

x_{ij}

	1	2	3	u_i
1	**0**	-1	0	1
2	**0**	**0**	0	2
3	-3	**0**	**0**	3
v_j	1	2	0	

r_{ij}

Iterazione 1. Moltiplicatori e costi ridotti della tabella a fianco seguono dal sistema (4.5) impostato per la base B_0:

$$u_1 + v_1 = 2$$

$$u_2 + v_1 = 3$$

$$u_2 + v_2 = 4$$

$$u_3 + v_2 = 5$$

$$u_3 + v_3 = 3.$$

Imponendo arbitrariamente $v_3 = 0$, il sistema diviene completamente determinato, permettendo di ricavare

$$u_1 = 1, \qquad u_2 = 2, \qquad u_3 = 3,$$

$$v_1 = 1, \qquad v_2 = 2, \qquad v_3 = 0.$$

Da questi si ricavano i valori $r_{ij} = c_{ij} - (u_i + v_j)$ sopra riportati — in grassetto sono evidenziate le celle in base. I costi ridotti negativi indiccano che la base corrente B_0 non è ottima; si sceglie quindi la cella $(3,1)$, con costo ridotto $r_{31} = -3$, come cella entrante in base. Il ciclo alternante contenuto in $B_0 \cup \{(3,1)\}$ è $C = [(3,1)^+, (2,1)^-, (2,2)^+, (3,2)^-]$.
Risulta $\varepsilon = \min\{x_{21}, x_{32}\} \min\{10, 30\} = 10$, il che indica in $(2,1)$ la cella destinata a lasciare la base. Quindi la nuova base è

$$B_1 = B_0 \cup \{(3,1)\} \setminus \{(2,1)\} = \{(1,1), (2,2), (3,1), (3,2), (3,3)\},$$

con le variabili aggiornate come segue.

	1	2	3	a_i
1	40			40
2	10^-	40^+		50
3	$^+$	30^-	30	60
b_j	50	70	30	

x_{ij}, B_0

$\Longrightarrow$

	1	2	3	a_i
1	40			40
2		50		50
3	10	20	30	60
b_j	50	70	30	

x_{ij}, B_1

Iterazione 2. Valutando moltiplicatori e costi ridotti si ottiene, per la nuova base B_1:

	1	2	3	a_i
1	40^-	$^+$		40
2		50		50
3	10^+	20^-	30	60
b_j	50	70	30	

x_{ij}

	1	2	3	u_i
1	**0**	−4	−3	1
2	3	**0**	0	2
3	**0**	**0**	**0**	3
v_j	1	2	0	

r_{ij}

Avendo $r_{12}, r_{13} < 0$, la base corrente non è ottima. Si sceglie $(1,2)$ come cella entrante e si identifica il ciclo alternante $C = [(1,2)^+, (1,1)^-, (3,1)^+, (3,2)^-]$. Si aggiorna quindi con $\varepsilon = 20$, $B_2 = B_1 \cup \{(1,2)\} \setminus \{(3,2)\}$.

Iterazione 3. $B_2 = \{(1,1), (1,2), (2,2), (3,1), (3,3)\}$.

	1	2	3	a_i
1	20^-	20^+		40
2		50^-	$+$	50
3	30^+		30^-	60
b_j	50	70	30	

x_{ij}

	1	2	3	u_i
1	**0**	**0**	-3	1
2	-1	**0**	-4	2
3	**0**	4	**0**	3
v_j	1	2	0	

r_{ij}

La cella $(2,3)$ deve entrare in base; $B_2 \cup (2,3)$ contiene $C = [(2,3)^+, (3,3)^-, (3,1)^+, (1,1)^-, (1,2)^+, (2,2)^-]$, che dà $\varepsilon = \min\{20, 50, 30\} = 20$, con cella uscente $(1,1)$. Quindi $B_3 = B_2 \cup \{(2,3)\} \setminus \{(1,1)\}$.

Iterazione 4. $B_3 = \{(1,2), (2,2), (2,3), (3,1), (3,3)\}$.

	1	2	3	a_i
1		40		40
2		20	20	50
3	50		10	60
b_j	50	70	30	

x_{ij}

	1	2	3	u_i
1	4	**0**	1	1
2	3	**0**	**0**	2
3	**0**	0	**0**	3
v_j	1	2	0	

r_{ij}

Tutti i costi ridotti sono non-negativi, quindi

$$x_{12} = 40, \qquad x_{22} = 30, \qquad x_{23} = 20, \qquad x_{31} = 50, \qquad x_{33} = 10,$$

$$x_{ij} = 0 \quad \text{per ogni } (i,j) \notin B_3,$$

è una soluzione ottima di costo $z = \sum_{(i,j) \in B_3} c_{ij} x_{ij} = 320$.

ESEMPIO 4.2. Si risolva il problema di trasporto con tre origini, tre destinazioni e la seguente matrice dei costi— i valori a_i e b_j sono riportati a margine.

	1	2	3	a_i
1	8	7	4	50
2	5	2	3	20
3	1	6	5	20
b_j	25	25	10	

Soluzione Il problema non è bilanciato, in quanto $\sum_{i=1^3} a_i - \sum_{j=1}^{3} b_j = 90 - 60 = 30$; si provvede quindi al bilanciamento aggiungendo una destinazione fittizia 4, con

$b_4 = 30$.

	1	2	3	4	a_i
1	8	7	4	0	50
2	5	2	3	0	20
3	1	6	5	0	20
b_j	25	25	10	30	

Partendo dal problema bilanciato, applicando il metodo dell'angolo Nord-Ovest si ottiene la base iniziale B_0.

$$B_0 = \{(1,1),(1,2),(1,3),(2,3),(2,4),(3,4)\}$$

	1	2	3	4	a_i
1	25	25	0		50
2			10	10	20
3				20	20
b_j	25	25	10	30	

x_{ij}

Iterazione 1. Calcolando i costi ridotti rispetto a B_0 si ottiene quanto segue.

	1	2	3	4	a_i
1	25^-	25	0^+		50
2			10^-	10^+	20
3	$+$			20^-	20
b_j	25	25	10	30	

x_{ij}

	1	2	3	4	u_i
1	**0**	**0**	**0**	-1	1
2	-2	-4	**0**	**0**	0
3	-6	0	2	**0**	0
v_j	7	6	3	0	

r_{ij}

Risulta $C = [(3,1)^+,(1,1)^-,(1,3)^+,(2,3)^-,(2,4)^+,(3,4)^-]$, $\varepsilon = \min\{25,10,20\} = 10 = x_{23}$, quindi $B_1 = B_0 \cup \{(3,1)\} \setminus \{(2,3)\}$.

Iterazione 2. $B_1 = \{(1,1),(1,2),(1,3),(2,4),(3,1),(3,4)\}$.

	1	2	3	4	a_i
1	15	25	10		50
2				20	20
3	10			10	20
b_j	25	25	10	30	

x_{ij}

	1	2	3	4	u_i
1	**0**	**0**	**0**	-7	7
2	4	2	6	**0**	0
3	**0**	6	8	**0**	0
v_j	1	0	-3	0	

r_{ij}

Scelta $(1,4)$ come cella entrante in base risulta $C = [(1,4)^+,(3,4)^-,(3,1)^+,(1,1)^-]$, $\varepsilon = \min\{15, 10 = 10 = x_{34}\}$ e quindi $B_2 = B_1 \cup \{(1,4)\} \setminus \{(3,4)\}$.

Iterazione 3. $B_2 = \{(1,1),(1,2),(1,3),(1,4),(2,4),(3,1)\}$.

	1	2	3	4	a_i
1	5	25^-	10	10^+	50
2		+		20^-	20
3	20				20
b_j	25	25	10	30	

x_{ij}

	1	2	3	4	u_i
1	**0**	**0**	**0**	**0**	0
2	−3	−5	−1	**0**	0
3	**0**	6	8	7	−7
v_j	8	7	4	0	

r_{ij}

Scelta $(2,2)$ come cella entrante risulta $C = [(2,2)^+,(1,2)^-,(1,4)^+,(2,4)^-]$, $\varepsilon = \min\{25,20\} = 20 = x_{24}$, quindi $B_3 = B_2 \cup \{(2,2)\} \setminus \{(2,4)\}$.

Iterazione 4. $B_3 = \{(1,1),(1,2),(1,3),(1,4),(2,2),(3,1)\}$.

	1	2	3	4	a_i
1	5	5	10	30	50
2		20			20
3	20				20
b_j	25	25	10	30	

x_{ij}

	1	2	3	4	u_i
1	**0**	**0**	**0**	**0**	0
2	2	**0**	4	5	−5
3	**0**	6	8	7	−7
v_j	8	7	4	0	

r_{ij}

La soluzione

$$x_{11} = 5, \quad x_{12} = 5, \quad x_{13} = 10, \quad x_{14} = 30$$

$$x_{22} = 20, \quad x_{31} = 20,$$

$$x_{ij} = 0, \text{per ogni } (i,j) \notin B_3,$$

è quindi ottima.

4.2 Problema dell'assegnamento

4.2.1 Modello e proprietà

Siano N, M insiemi con $|N| = |M| = n$; data una matrice di di costi di accoppiamento c_{ij}, $(i,j) \in N \times M$, il *problema dell' assegnamento* richiede di selezionare da $N \times M$ n coppie *disgiunte* il cui costo totale sia minimo. Il problema ammette una formulazione in programmazione lineare.

$$\min \sum_{i \in N} \sum_{j = \in M} c_{ij} x_{ij} \tag{4.6}$$

soggetto a

$$\sum_{j \in M} x_{ij} = 1 \qquad i \in N, \tag{4.7}$$

$$\sum_{i \in N} x_{ij} = 1 \qquad j \in M, \tag{4.8}$$

$$x_{ij} \in \{0,1\} \ \ (i,j) \in N \times M. \tag{4.9}$$

Le (4.6)–(4.9) possono modellare un problema di accoppiamento completo tra gli elementi di due insiemi di uguale cardinalità (di persone e compiti, task e macchine,...) a costo minimo, tra “origini” e “destinazioni”. Alle variabili binarie x_{ij} è attribuito il seguente significato: $x_{ij} = 1$ se e solo se (i, j) sono accoppiati. I vincoli (4.7) impongono che ogni origine partecipi ad uno ed un solo accopiamento tra quelli selezionati, mentre i vincoli (4.8) impongono che ogni destinazione partecipi ad uno ed un solo accoppiamento.

Osservazione. Il problema dell'assegnamento è un caso particolare del problema dei trasporti. Ogni soluzione di base del sistema (4.7)–(4.8) ha quindi valori interi, per le note proprietà delle matrici totalmente unimodulari. Il modello (4.6)–(4.9) può quindi essere risolto rilassando i vincoli di interezza (4.9) a

$$x_{ij} \geq 0, \ \ (i,j) \in N \times M. \tag{4.9$'$}$$

4.2.2 L' Algoritmo Ungherese

Il problema dell'assegnamento può essere risolto mediante tecniche generali di programmazione lineare (ad esempio mediante l'algoritmo del simplesso). L'algoritmo riportato di seguito risulta essere, però più efficiente.

Function Algoritmo ungherese
Creare la matrice dei costi ridotti $\bar{c}_{ij}$
repeat
  Determinare un accoppiamento $\mathcal{M} \subseteq N \times M$ di cardinalità massima e costo ridotto $\bar{c}(\mathcal{M}) = \sum_{(i,j) \in \mathcal{M}} \bar{c}_{ij} = 0$
  if $|\mathcal{M}| < n$ **then**
    Aggiornare i costi ridotti
  end if
until $|\mathcal{M}| = n$

Il passo iniziale dell'algoritmo ⟨Creare la matrice dei costi ridotti ...⟩ è suscettibile di diverse implementazioni; una procedura molto usata è la seguente. Determinare

$$u_i = \min\{c_{ij} : j \in M\},$$

$$v_j = \min\{c_{ij} - u_i : i \in j\},$$

e porre

$$\bar{c}_{ij} = c_{ij} - u_i - v_j.$$

Questo garantisce che fin dall'inizio su ogni riga e ogni colonna appaia almeno una cella con costo ridotto nullo.
L'accoppiamento di cardinalità massima $\mathcal{M}$ è determinato sul grafo bipartito $G(N \cup M, E)$, con

$$E = \{(i,j) : i \in N, j \in M, \bar{c}_{ij} = 0\}.$$

La regola di aggiornamento dei costi ridotti fa uso delle etichette generate dall'ultima iterazione dell'algoritmo per la determinazione dell'accoppiamento di cardinalità massima. Se $N' \subseteq N$, $M' \subseteq M$ sono gli insiemi di nodi etichettati su G in corrispondenza dell'accoppiamento di cardinalità massima $\mathcal{M}$, si considera l'insieme

$$S = N' \cup (M \setminus M')$$

. Si determina

$$\varepsilon = \min\{\bar{c}_{ij} : i, j \in S\}$$

e si aggiornano i costi ridotti come

$$\bar{c}_{ij} := \begin{cases} \bar{c}_{ij} - \varepsilon, & \text{se } i \in N', j \in M \setminus M', \\ \bar{c}_{ij} + \varepsilon, & \text{se } i \in N \setminus N', j \in M', \\ \bar{c}_{ij}, & \text{altrimenti.} \end{cases}$$

4.2.3 Esercizi risolti

ESEMPIO 4.3. Si consideri il problema di assegnamento associato alla seguente matrice dei costi, con $N = \{1, 2, 3, 4, 5\}$, $M = \{a, b, c, d, e\}$.

$$\begin{array}{c c} & \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{pmatrix} 5 & 3 & 2 & 6 & 1 \\ 4 & 2 & 8 & 11 & 7 \\ 4 & 3 & 5 & 6 & 6 \\ 7 & 2 & 3 & 7 & 4 \\ 3 & 2 & 1 & 3 & 4 \end{pmatrix} \end{array}$$

Inizializzazione. Applicando la procedura di riduzione si ottiene la matrice $\bar{c}_{ij}$.

	a	b	c	d	e	u_i
1	5	3	2	6	1	1
2	4	2	8	11	7	2
3	4	3	5	6	6	3
4	7	2	3	7	4	2
5	3	2	1	3	4	1

c_{ij}

$\Longrightarrow$

	a	b	c	d	e	u_i
1	4	2	1	5	0	1
2	2	0	6	9	5	2
3	1	0	2	3	3	3
4	5	0	1	5	2	2
5	2	1	0	2	3	1

$c_{ij} - u_i$

	a	b	c	d	e	u_i
1	4	2	1	5	0	1
2	2	0	6	9	5	2
3	1	0	2	3	3	3
4	5	0	1	5	2	2
5	2	1	0	2	3	1
v_j	1	0	0	2	0	

$c_{ij} - u_i$

$\Longrightarrow$

	a	b	c	d	e
1	3	2	1	3	0
2	1	0	6	7	5
3	0	0	2	1	3
4	4	0	1	3	2
5	1	1	0	0	3

$\bar{c}_{ij}$

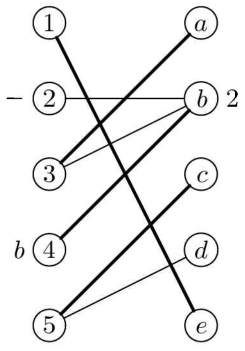
Iterazione 1. Il grafo bipartito corrispondente di costi ridotti determinati è $G(N \cup M, E)$, con

$$E = \{(1, e), (2, b), (3, a), (3, b), (4, b), (5, c), (5, d)\}.$$

Applicando l'algoritmo per il problema del massimo matching su G si ottiene $\mathcal{M} = \{(1, e), (3, a), (4, b), (5, c)\}$, con l'etichettatura riportata qui di seguito.

	a	b	c	d	e
1	3	2	1	3	**0**
2	1	0	6	7	5
3	**0**	0	2	1	3
4	4	**0**	1	3	2
5	1	1	**0**	0	3

$\bar{c}_{ij}$



Essendo $|\mathcal{M}| = 4$, si procede all'aggiornamento dei costi ridotti. In forma tabellare, l'esecuzione dell'aggiornamento può essere ricondotta alle seguenti regole pratiche.

1. Barrare tutte le righe $i \in N$ corrispondenti a nodi non etichettati e tutte le colonne $j \in M$ corrispondenti a nodi etichettati.

2. Il valore ε cercato corrisponde al minimo $\bar{c}_{ij}$ rimasto <u>non</u> barrato.

3. Aggiornare i costi ridotti:

$$\bar{c}_{ij} := \begin{cases} \bar{c}_{ij} - \varepsilon, & \text{se la cella } (i,j) \text{ non è barrata,} \\ \bar{c}_{ij} + \varepsilon, & \text{se la cella } (i,j) \text{ è doppiamente barrata,} \\ \bar{c}_{ij}, & \text{altrimenti.} \end{cases}$$

L'aggiornamento produce quindi

	a	~~b~~	c	d	e
~~1~~	~~3~~	~~2~~	~~1~~	~~3~~	~~0~~
2	1	~~0~~	6	7	5
~~3~~	~~0~~	~~0~~	~~2~~	~~1~~	~~3~~
4	4	~~0~~	1	3	2
~~5~~	~~1~~	~~1~~	~~0~~	~~0~~	~~3~~

$\bar{c}_{ij}$

$$\Longrightarrow \varepsilon = 1 \Longrightarrow$$

	a	b	c	d	e
1	3	3	1	3	0
2	0	0	5	6	4
3	0	1	2	1	3
4	3	0	0	2	1
5	1	2	0	0	3

$\bar{c}_{ij}$

Iterazione 2. Si ridefinisce il grafo bipartito $G(N \cup M, E)$ in funzione dei costi ridotti aggiornati:

$$E = \{(1,e), (2,a), (2,b), (3,a), (4,b), (4,c), (5,c), (5,d)\}.$$

Applicando l'algoritmo per il massimo matching sul nuovo G si ottiene

$$\mathcal{M} = \{(1,e), (2,b), (3,a), (4,c), (5,d)\},$$

di cardinalità $|\mathcal{M}| = 5$ e quindi corrispondente ad un assegnamento ottimo, di costo $c(\mathcal{M}) = 1 + 2 + 4 + 3 + 3 = 13$.

ESEMPIO 4.4. Si consideri il problema di assegnamento associato alla seguente matrice dei costi, con $N = \{1,2,3,4,5\}$, $M = \{a,b,c,d,e\}$.

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{pmatrix} 3 & 6 & 7 & 10 & 6 \\ 5 & 6 & 3 & 8 & 9 \\ 2 & 8 & 4 & 16 & 3 \\ 8 & 6 & 5 & 9 & 10 \\ 6 & 8 & 4 & 12 & 12 \end{pmatrix} \end{array}$$

Inizializzazione. Riducendo la matrice dei costi si ottiene quanto segue.

	a	b	c	d	e	u_i
1	3	6	7	10	6	3
2	5	6	3	8	9	3
3	2	8	4	16	3	2
4	8	6	5	9	10	5
5	6	8	4	12	12	4

c_{ij}

$\Longrightarrow$

	a	b	c	d	e	u_i
1	0	3	4	7	3	3
2	2	3	0	5	6	3
3	0	6	2	14	1	2
4	3	1	0	4	5	5
5	2	4	0	6	6	4

$c_{ij} - u_i$

	a	b	c	d	e	u_i
1	0	3	4	7	3	3
2	2	3	0	5	6	3
3	0	6	2	14	1	2
4	3	1	0	4	5	5
5	2	4	0	6	6	4
v_j	0	1	0	4	1	

$c_{ij} - u_i$

$\Longrightarrow$

	a	b	c	d	e
1	0	2	4	3	2
2	2	2	0	1	5
3	0	5	2	10	0
4	3	0	0	0	4
5	2	3	0	2	5

$\bar{c}_{ij}$

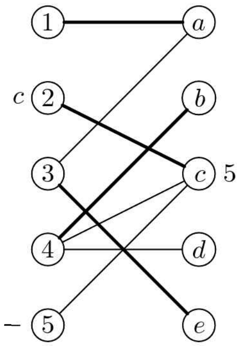
Iterazione 1. Determinando un accoppiamento di cardinalità massima su $G(N \cup M, E)$, si ottiene

$$\mathcal{M} = \{(1,a),(2,c),(3,e),(4,b)\}\,,$$

con la relativa etichettatura.

	a	b	c	d	e
1	**0**	2	4	3	2
2	2	2	**0**	1	5
3	0	5	2	10	**0**
4	3	**0**	0	0	4
5	2	3	0	2	5

$\bar{c}_{ij}$



Avendo $|\mathcal{M}| < 5$ si procede ad aggiornare i costi ridotti.

	a	b	c	d	e
1	0	2	4	3	2
2	2	2	0	1	5
3	0	5	2	10	0
4	3	0	0	0	4
5	2	3	0	2	5

$\bar{c}_{ij}$

$\Longrightarrow \varepsilon = 1 \Longrightarrow$

	a	b	c	d	e
1	0	2	5	3	2
2	1	1	0	0	4
3	0	5	3	10	0
4	3	0	1	0	4
5	1	2	0	1	4

$\bar{c}_{ij}$

Iterazione 2. Sul grafo bipartito indotto dai costi ridotti aggiornati si ottiene l'accoppiamento $\mathcal{M} = \{(1,a),(2,d),(3,e),(4,b),(5,c)\}$, corrispondente ad un assegnamento ottimo di costo $c(\mathcal{M}) = 3+8+3+6+4 = 24$.

ESEMPIO 4.5. Risolvere il problema di assegnamento associato alla seguente matrice dei costi, con $N = \{1,2,3,4,5\}$, $M = \{a,b,c,d,e\}$.

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{pmatrix} 2 & 1 & 3 & 8 & 7 \\ 3 & 1 & 5 & 4 & 9 \\ 5 & 1 & 8 & 4 & 7 \\ 8 & 1 & 3 & 6 & 8 \\ 7 & 7 & 8 & 2 & 1 \end{pmatrix} \end{array}$$

Inizializzazione. Riducendo la matrice dei costi si ottiene quanto segue.

	a	b	c	d	e	u_i
1	2	1	3	8	7	1
2	3	1	5	4	9	1
3	5	1	8	4	7	1
4	8	1	3	6	8	1
5	7	7	8	2	1	1

c_{ij}

$\Longrightarrow$

	a	b	c	d	e	u_i
1	1	0	2	7	6	1
2	2	0	4	3	8	1
3	4	0	7	3	6	1
4	7	0	2	5	7	1
5	6	6	7	1	0	1

$c_{ij} - u_i$

	a	b	c	d	e	u_i
1	1	0	2	7	6	1
2	2	0	4	3	8	1
3	4	0	7	3	6	1
4	7	0	2	5	7	1
5	6	6	7	1	0	1
v_j	1	0	2	1	0	

$c_{ij} - u_i$

$\Longrightarrow$

	a	b	c	d	e
1	0	0	0	6	6
2	1	0	2	2	8
3	3	0	5	2	6
4	6	0	0	4	7
5	5	6	5	0	0

$\bar{c}_{ij}$

Iterazione 1. Definendo $G(N \cup M, E)$ con

$$E = \{(1,a),(1,b),(1,c),(2,b),(3,b),(4,b),(4,c),(5,d),(5,e)\}$$

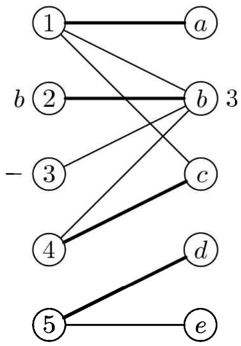
si arriva all'accoppiamento di cardinalità massima

$$\mathcal{M} = \{(1,a),(2,b),(4,c),(5,d)\}\,,$$

con la relativa etichettatura.

	a	b	c	d	e
1	**0**	0	0	6	6
2	1	**0**	2	2	8
3	3	0	5	2	6
4	6	0	**0**	4	7
5	5	6	5	**0**	0

$\bar{c}_{ij}$



Si applica quindi la regola di aggiornamento dei costi ridotti.

	a	b	c	d	e
1	0	0	0	6	6
2	1	0	2	2	8
3	3	0	5	2	6
4	6	0	0	4	7
5	5	6	5	0	0

$\bar{c}_{ij}$

$$\Longrightarrow \ \varepsilon = 1 \ \Longrightarrow$$

	a	b	c	d	e
1	0	1	0	6	6
2	0	0	1	1	7
3	2	0	4	1	5
4	6	1	0	4	7
5	5	7	5	0	0

$\bar{c}_{ij}$

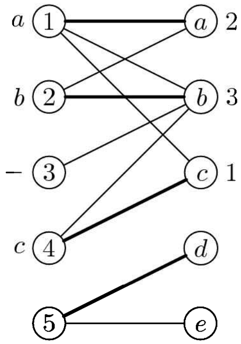
Iterazione 2. Il nuovo grafo bipartito $G(N \cup M, E)$ ha archi

$$E = \{(1,a),(1,c),(2,b),(3,b),(4,c),(5,d),(5,e)\}\,;$$

$\mathcal{M} = \{(1,a),(2,b),(4,c),(5,d)\}$ risulta ancora essere un accoppiamento di cardinalità massima: ciò che cambia è l'etichettatura finale.

	a	b	c	d	e
1	**0**	1	0	6	6
2	0	**0**	1	1	7
3	2	0	4	1	5
4	6	1	**0**	4	7
5	5	7	5	**0**	0

$\bar{c}_{ij}$



Si procede quindi ad un nuovo aggiornamento.

	a	b	c	d	e
1	0	1	0	5	6
2	0	0	1	0	7
3	2	0	4	1	5
4	6	1	0	3	7
5	5	7	5	0	0

$\bar{c}_{ij}$

$$\Longrightarrow \varepsilon = 1 \Longrightarrow$$

	a	b	c	d	e
1	0	1	0	5	5
2	0	0	1	0	6
3	2	0	4	0	5
4	6	1	0	3	6
5	6	8	6	0	0

$\bar{c}_{ij}$

Iterazione 3. Sul nuovo grafo si può identificare l'accoppiamento

$$\mathcal{M} = \{(1,a),(2,b),(3,d),(4,c),(5,e)\},$$

che corrisponde ad un assegnamento ottimo.

4.3 Esercizi proposti

1. Le tabelle seguenti riportano i costi di trasporto c_{ij} e, a margine, i valori di a_i e b_j. Risolvere i problemi di trasporto corrispondenti (applicare la procedura di bilanciamento quando necessario).

	1	2	3	4	a_i
1	8	5	8	9	90
2	4	6	10	4	50
3	2	11	4	3	60
b_j	80	60	30	50	

(a)

	1	2	3	a_i
1	2	8	4	20
2	3	4	9	20
3	8	7	10	40
b_j	20	30	20	

(b)

	1	2	3	4	a_i
1	4	7	12	8	50
2	9	2	6	11	50
3	7	9	2	1	20
b_j	40	60	10	30	

(c)

	1	2	3	4	a_i
1	5	8	15	9	50
2	7	4	2	2	60
3	4	2	3	1	20
b_j	50	40	30	10	

(d)

	1	2	3	4	a_i
1	10	8	4	3	40
2	7	9	8	8	20
3	9	9	7	8	80
4	4	2	1	9	60
b_j	50	50	60	40	

(e)

	1	2	3	4	a_i
1	10	7	2	15	150
2	4	8	9	8	50
3	2	3	4	6	50
b_j	100	100	25	25	

(f)

2. Risolvere i problemi di assegnamento corrispondenti alle seguenti matrici, dove $N = \{1, 2, 3, 4, 5\}$, $M = \{a, b, c, d, e\}$.

$$(a) \quad \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{pmatrix} 5 & 3 & 2 & 6 & 1 \\ 4 & 2 & 8 & 11 & 7 \\ 4 & 3 & 5 & 6 & 6 \\ 7 & 2 & 3 & 7 & 4 \\ 3 & 2 & 1 & 3 & 4 \end{pmatrix} \end{array} \qquad (b) \quad \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{pmatrix} 3 & 6 & 7 & 10 & 6 \\ 5 & 6 & 3 & 8 & 9 \\ 2 & 2 & 4 & 16 & 3 \\ 8 & 6 & 5 & 9 & 10 \\ 6 & 8 & 4 & 12 & 12 \end{pmatrix} \end{array}$$

$$(c) \quad \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{pmatrix} 2 & 5 & 3 & 8 & 1 \\ 3 & 6 & 2 & 7 & 5 \\ 2 & 4 & 6 & 3 & 8 \\ 10 & 8 & 5 & 2 & 5 \\ 4 & 5 & 2 & 3 & 1 \end{pmatrix} \end{array} \qquad (d) \quad \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{pmatrix} 7 & 2 & 1 & 3 & 5 \\ 3 & 5 & 2 & 4 & 1 \\ 1 & 2 & 4 & 6 & 3 \\ 2 & 16 & 5 & 3 & 5 \\ 4 & 7 & 2 & 3 & 6 \end{pmatrix} \end{array}$$

$$(e) \quad \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{pmatrix} 5 & 2 & 4 & 3 & 3 \\ 3 & 4 & 3 & 8 & 2 \\ 3 & 2 & 5 & 4 & 6 \\ 7 & 3 & 4 & 6 & 5 \\ 5 & 1 & 5 & 4 & 2 \end{pmatrix} \end{array} \qquad (f) \quad \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{pmatrix} 5 & 3 & 4 & 3 & 2 \\ 3 & 4 & 3 & 8 & 3 \\ 3 & 2 & 5 & 4 & 5 \\ 4 & 3 & 4 & 6 & 5 \\ 5 & 6 & 5 & 4 & 2 \end{pmatrix} \end{array}$$

5

Metodi esatti per l'Ottimizzazione Combinatoria

5.1 Metodo del Branch & Bound

5.1.1 Richiami teorici

Il metodo del Branch & Bound è un metodo di *enumerazione implicita* che decompone l'insieme delle soluzioni ammissibili del problema di partenza in sottoinsiemi via via più ristretti. Tale processo è detto *processo di branch.* Esso, applicato recursivamente ad ogni sottoinsieme, costruisce una struttura ad albero, detta *albero delle decisioni* o *albero di branch*, i cui nodi sono i sottoinsiemi e gli archi sono i branch effettuati. Il nodo radice di un albero di branch rappresenta il problema di cui si vuole trovare la soluzione e solitamente è indicato come nodo 0. Dato un problema, i sottoinsiemi di soluzioni ammissibili derivanti da un suo branch sono detti *nodi figli.* Inoltre, il problema che ha per soluzioni ammissibili solo e soltanto quelle del sottoinsieme associato ad un nodo è detto *problema di nodo.*

Dato un nodo ed il problema ad esso associato, può essere calcolata una stima (inferiore, o *lower bound*, nel caso di problema di minimo, superiore o *upper bound*, nel caso di problema di massimo) del valore ottimo della sua funzione obiettivo. Il processo di calcolo di tale stima è detto processo di *rilassamento* o *bounding.* Negli esercizi proposti di seguito verranno utilizzati i seguenti metodi di calcolo del bound:

- rilassamento continuo: si ignorano i vincoli di interezza delle variabili intere. Il problema che si ottiene è un normale problema di Programmazione Lineare (PL), risolvibile all'ottimo efficientemente con un qualsiasi metodo tra quelli presentati nel Capitolo 2;

- rilassamento lagrangiano: uno o più vincoli vengono tolti dalla formulazione e la loro eventuale violazione viene presa in considerazione come penalizzazione della funzione obiettivo del problema rilassato.

La dimensione dell'albero delle decisioni cresce esponenzialmente con l'aumentare del numero di variabili, per cui ben presto diventa troppo elevato per poter generare tali soluzioni in tempi computazionali ragionevoli. Per tale motivo, l'algoritmo del Branch & Bound affianca alla fase di branch un insieme di regole che permettono di chiudere un nodo senza generare il rispettivo sottoalbero. Tali regole sfruttano proprio il calcolo del bound e sono dette *regole di fathoming.*

1. Chiusura per inammissibilità. Dato un problema di nodo P_i ed il suo rilassamento P_{Ri}, si ottiene che il problema P_{Ri} non ammette soluzione. Visto che tale problema ha uno spazio delle soluzioni più grande di quello di P_i, ciò implica che anche P_i non ammette soluzioni ammissibili.

2. Chiusura per ottimalità. Dato un problema di nodo P_i ed il suo rilassamento P_{Ri}, si supponga che la soluzione ottima di P_{Ri} sia anche soluzione ammissibile di P_i (rispetti, cioè, i vincoli rilassati). Ciò implica che la soluzione ottima di P_{Ri} è anche soluzione ottima di P_i. Essa, inoltre, è anche soluzione ammissibile del problema associato al nodo radice dell'albero di branch.

3. Chiusura per non migliorabilità. Si supponga di trovarsi in un problema di minimo e si indichi con $\overline{z}$ la funzione obiettivo della migliore soluzione ammissibile del problema da risolvere P. Se, ad un generico nodo i, si ha che il lower bound LB_i, cioè la funzione obiettivo ottima del problema rilassato associato al nodo i, è tale che $LB_i \geq \overline{z}$, allora il nodo preso in considerazione e tutti i nodi del sottoalbero che lo vedono come nodo radice non possono contenere soluzioni migliori di quella già nota, per cui è inutile esplorare ulteriormente tale nodo ed i suoi discendenti. Nel caso di problema di massimo il ragionamento è lo stesso, ma, considerando che la funzione obiettivo del problema rilassato costituisce un upper bound, la regola di fathoming diventa $UB_i \leq \overline{z}$.

Accanto a questa struttura principale, numerose altre componenti sono presenti in un algoritmo di Branch & Bound. Tra di esse due rivestono particolare importanza:

1. regola di esplorazione dell'albero: regola con cui, dati i nodi ancora non valutati, viene scelto il prossimo nodo a cui applicare le regole di fathoming e di branch. Tra le principali ricordiamo:

   - Depth First: visita i profondità;
   - Best First: si valuta il nodo con il lower bound minore nel caso di problema di minimo ed upper bound maggiore nel caso di problema di massimo.

2. regola di branch: regola che genera il partizionamento in sottoinsiemi del problema. La più classica è la cosiddetta regola di branch binario associata al rilassamento continuo. Data una variabile $x_i = p$ che viola il vincolo di interezza , si effettua una suddivisione del problema di nodo in due sottoproblemi caratterizzati dal considerare in aggiunta ai vincoli presenti rispettivamente i vicoli $x_i \leq \lfloor p \rfloor$ e $x_i \geq \lceil p \rceil$. In tal modo, si escludono non solo le soluzioni aventi $x_i = p$, ma anche tutte le soluzioni con $\lfloor p \rfloor < x_i < \lceil p \rceil$.

5.1.2 Esercizi risolti

ESEMPIO 5.1.
Si consideri un problema a variabili intere con funzione obiettivo di massimo. Si supponga, ad un certo passo, di aver ottenuto l'albero di ricerca in figura 5.1 (ad ogni nodo i sono associati i relativi valori di lower ed upper bound $[LB_i, UB_i]$).

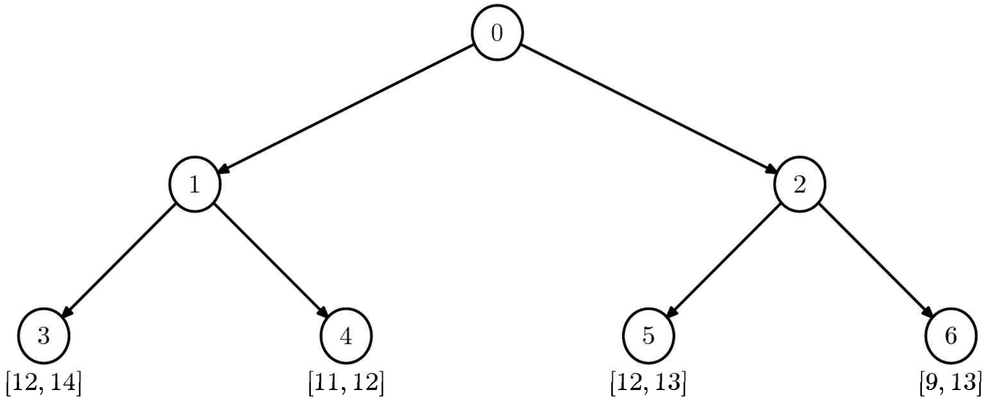


Figura 5.1: Albero di branch relativo all'esercizio 5.1

1. Supponendo di utilizzare la politica di visita dei nodi Best First, chi è il prossimo nodo da considerare tra quelli ancora aperti? Perché?
2. Se la migliore soluzione trovata fino a questo momento ha valore z=13, quali tra i nodi ancora aperti possono essere chiusi? Perché?

   Si indichi inoltre se le seguenti affermazioni siano vere o false, motivando le risposte.
3. L'upper bound di un nodo figlio è sempre minore o uguale all'upper bound del nodo padre.
4. Il lower bound di un nodo figlio è sempre maggiore o uguale del lower bound del nodo padre.

Soluzione

1. Nel caso di un problema di massimo, la politica Best First esplora per primi i nodi più promettenti in base al loro valore di upper bound. In particolare, il prossimo nodo da considerare è dunque il nodo con massimo upper bound, ovvero il nodo 3 con upper bound 14.
2. Possono essere chiusi tutti i nodi con un upper bound minore o uguale alla miglior soluzione trovata fino a questo momento, ovvero i nodi 4, 5 e 6.
3. Il problema del nodo figlio è lo stesso problema rappresentato dal nodo padre con un vincolo in più. In un problema di massimo l'upper bound è la soluzione del rilassamento continuo del problema. Il problema più vincolato (problema del nodo figlio) avrà dunque upper bound minore o uguale a quella del problema meno vincolato (problema del nodo padre). L'affermazione è pertanto vera.
4. In un problema di massimo il lower bound è una soluzione ammissibile del problema di nodo. I valori di lower bound sono dunque fortemente dipendenti dal metodo di calcolo del lower bound stesso (euristiche, arrotondamenti, etc.). Non esiste dunque in generale nessuna correlazione tra i valori di lower bound per un nodo figlio ed il nodo padre. L'affermazione è quindi falsa.

ESEMPIO 5.2.

Si consideri un problema a variabili intere scritto nella forma

$$\begin{aligned}
&\min z = \textstyle\sum_{i\in I} c_i x_i \\
&s.t. \\
&\textstyle\sum_i a_i^1 x_i >= b_1 \\
&\textstyle\sum_i a_i^2 x_i = b_2 \\
&x_i \in \mathbb{Z}^+,\ \forall i \in I.
\end{aligned}$$

Si è deciso di risolvere tale problema mediante Branch & Bound con rilassamento continuo. Si supponga di trovarsi ad un generico passo dell'algoritmo e di aver ottenuto l'albero di ricerca in figura 5.2. Ad ogni nodo aperto sono associati i valori [LB, UB, V_c=val], rappresentanti rispettivamente lower bound, upper bound, variabile critica sulla quale si è deciso di effettuare l'eventuale branch e valore assunto dalla stessa nel rilassamento.
Supponendo di utilizzare la politica di visita dei nodi Best First, si discuta come procederà l'algoritmo del Branch & Bound, specificando:

- quale sarà il prossimo nodo esplorato secondo la politica Best First;
- quali nodi possono essere chiusi e perché, supponendo di conoscere una soluzione ammissibile del problema avente funzione obiettivo pari a 12;
- quali saranno i nodi generati nel caso di branch binario;
- quali saranno i problemi rilassati dei nodi generati;

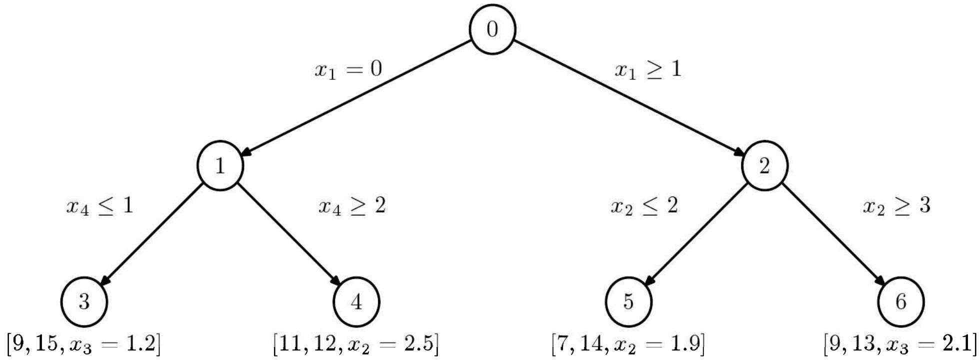


Figura 5.2: Albero di branch relativo all'esercizio 5.2

- eventuali restrizioni ai valori assunti dai rilassamenti.

Soluzione

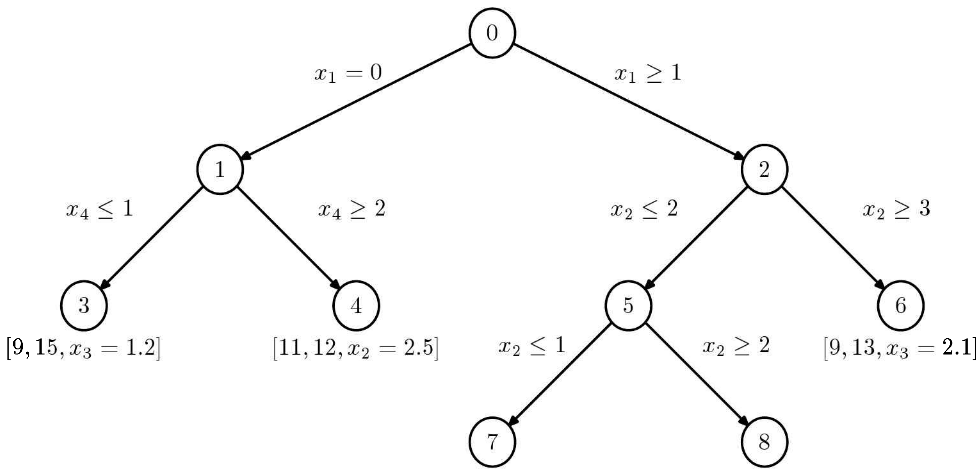


Figura 5.3: Soluzione dell'esercizio 5.2: albero di branch

La politica Best First esplora per primi i nodi più promettenti in base al valore della soluzione ottima del loro problema rilassato. Trattandosi di un problema di minimo,

nel nostro caso la politica Best First sceglierà il nodo con il lower bound minimo, cioè il nodo 5.
Possono essere chiusi tutti i nodi con un lower bound maggiore o uguale alla miglior soluzione trovata fino a questo momento. Nessun nodo rispetta tale condizione, per cui nessun nodo può essere chiuso.
Come si è detto, il prossimo nodo considerato sarà il nodo 5. Dai dati forniti, sappiamo che la variabile scelta come variabile critica è la variabile x_2. Volendo effettuare un branch di tipo binario, il nodo 5 genererà due nodi figli, i nodi 7 ed 8. I problemi dei nodi 7 ed 8 sono ottenuti a partire dal problema del nodo 5 ed aggiungendo a tale problema i vincoli $x_2 <= 1$ per il nodo 7 e $x_2 >= 2$ per il nodo 8. In tal modo possiamo eliminare le soluzione continue in cui la variabile x_2 assume valori continui nell'intervallo $(1, 2)$.
Più precisamente, i problemi rilassati relativi ai nodi 7 ed 8 saranno i seguenti:

$$
\begin{array}{ll}
\min z = \sum_{i \in I} c_i x_i & \\
s.t. & (R_7) \\
\sum_i a_i^1 x_i >= b_1 & \\
\sum_i a_i^2 x_i = b_2 & \\
x_1 \geq 1 & \\
x_2 \leq 2 & \\
x_2 \leq 1 & \\
x_i \geq 0,\ \forall i \in I, &
\end{array}
$$

$$
\begin{array}{ll}
\min z = \sum_{i \in I} c_i x_i & \\
s.t. & (R_8) \\
\sum_i a_i^1 x_i >= b_1 & \\
\sum_i a_i^2 x_i = b_2 & \\
x_1 \geq 1 & \\
x_2 \leq 2 & \\
x_2 \geq 2 & \\
x_i \geq 0,\ \forall i \in I. &
\end{array}
$$

Si noti come nel problema R_8 la coppia di vincoli $x_2 \leq 2$ e $x_2 \geq 2$ di fatto forzi la variabile ad assumere il valore 2, mentre il vincolo $x_2 \leq 1$ presente nel problema R_7 renda superfluo il vincolo $x_2 \leq 2$ imposto dal branch del nodo 2. Il nuovo albero di branch è illustrato in figura 5.3.
Circa i valori dei lower bound dei nodi 7 ed 8, si ricordi che i problemi figli sono più vincolati dei rispettivi problemi padre, per cui, indicando con LB_i in lower bound di un generico nodo, avremo che LB_7 ed LB_8 dovranno essere non inferiori ai lower bound dei nodi LB_5, LB_2 e LB_0.

ESEMPIO 5.3.

Risolvere, mediante l'algoritmo del Branch & Bound, il seguente problema:

$$\begin{aligned}
&\max 3x_1 + 5x_2 + 2x_3 - 5x_4 \\
&s.t. \qquad\qquad\qquad\qquad\qquad (P) \\
&10x_1 + 12x_2 + 7x_3 \leq 35 \\
&12x_2 + 6x_3 + 10x_4 \geq 18 \\
&x_i \in \mathbb{Z}^+ \quad \forall x_i.
\end{aligned}$$

Si risolva il problema mediante l'algoritmo del Branch & Bound supponendo:

- utilizzare per il calcolo del lower bound il rilassamento continuo;
- politica di visita dei nodi Best First;
- branch binario.

Soluzione Trattandosi di un problema di massimo, la soluzione del rilassamento continuo è un upper bound, mentre una qualsiasi soluzione ammissibile del problema è un lower bound.
Avendo supposto di rilassare i vincoli di interezza, i problemi di PL che avremo ai diverso nodi sono risolvibili mediante l'uso di un qualsiasi algoritmo per la PL (ad esempio il simplesso revisionato). Supponiamo che, al generico nodo i, la funzione obiettivo ottima del problema rilassato sia pari z_{PL_i}. Osservando che il problema ha solo variabili intere e che tutti i coefficienti sono interi, si può dedurre che una qualsiasi funzione obiettivo del problema avrà valori interi. Di conseguenza si può ricavare un rilassamento più efficace ponendo $UB_i = \lfloor z_{PL_i} \rfloor$, dove UB_i è l'upper bound del nodo i.

Nodo 0. Al nodo radice il problema utilizzato per il calcolo del lower bound è il seguente

$$\begin{aligned}
&\max 3x_1 + 5x_2 + 2x_3 - 5x_4 \\
&s.t. \qquad\qquad\qquad\qquad\qquad (R_0) \\
&10x_1 + 12x_2 + 7x_3 \leq 35 \\
&12x_2 + 6x_3 + 10x_4 \geq 18 \\
&x_i \geq 0 \quad \forall x_i.
\end{aligned}$$

Risolvendo il problema, si ottiene una funzione obiettivo $z_{PL_0} = 14.583$ e la soluzione continua associata è $x_2 = 2.916$. Il rilassamento del nodo vale quindi $UB_0 = \lfloor 14.58 \rfloor = 14$.
La scelta della variabile sulla quale effettuare il branch ricade su x_2, unica variabile continua. Il nodo 0 genera quindo due nodi 1 e 2, imponendo rispettivamente i vincoli $x_2 \leq 2$ e $x_2 \geq 3$. Al fine di scegliere quale sia il prossimo nodo da considerare, occorre calcolare gli upper bound dei nodi appena creati. Per il nodo 1, l'upper bound si ottiene risolvendo il problema

$$\begin{array}{ll} \max 3x_1 + 5x_2 + 2x_3 - 5x_4 & \\ s.t. & (R_1) \\ 10x_1 + 12x_2 + 7x_3 \leq 35 & \\ 12x_2 + 6x_3 + 10x_4 \geq 18 & \\ x_2 \leq 2 & \\ x_i \geq 0 \quad \forall x_i, & \end{array}$$

che porta alla soluzione $x_1 = 1.1, x_2 = 2$, corrispondente ad un valore della funzione obiettivo $z_{LP_1} = 13.3$. A partire dalla soluzione ottima del problema (R_1), si ricava il valore dell'upper bound $UB_1 = \lfloor 13.3 \rfloor = 13$.
Analogamente, per il nodo 2 il problema rilassato è il seguente

$$\begin{array}{ll} \max 3x_1 + 5x_2 + 2x_3 - 5x_4 & \\ s.t. & (R_2) \\ 10x_1 + 12x_2 + 7x_3 \leq 35 & \\ 12x_2 + 6x_3 + 10x_4 \geq 18 & \\ x_2 \geq 3 & \\ x_i \geq 0 \quad \forall x_i. & \end{array}$$

Il problema (R_2) è non ammissibile, infatti imponendo $x_2 \geq 3$ è facile vedere che il primo vincolo del problema non può essere soddisfatto. Per tale motivo, il nodo 2 può essere chiuso per non ammissibilità.
L'unico nodo ancora aperto è il nodo 1, per cui esso sarà utilizzato dall'algoritmo come prossimo nodo da processare. L'albero di branch parziale al termine del nodo 0 è mostrato in figura 5.4.

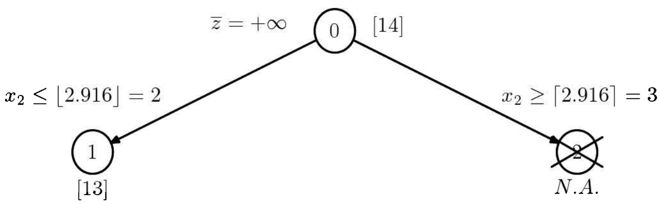


Figura 5.4: Esercizio 5.3: albero di branch parziale relativo al nodo 0

Nodo 1: $x_2 \leq 2$. Il nodo non può essere chiuso per alcun criterio di chiusura, per cui si procede con il branch binario. La soluzione ottima del problema (R_1) presenta una sola variabile con valore non intero, $x_1 = 1.1$, per cui essa è la variabile critica sulla quale effettuare il branch. Il nodo 1 genera quindi due figli, i nodi 3 e 4, aventi rispettivamente i vincoli aggiuntivi $x_1 \leq 1$ e $x_1 \geq 2$.

Al fine di scegliere quale sia il prossimo nodo da considerare, occorre calcolare gli upper bound dei nodi appena creati. Per il nodo 3, l'upper bound si ottiene risolvendo il problema

$$\begin{array}{ll} \max 3x_1 + 5x_2 + 2x_3 - 5x_4 & \\ s.t. & (R_3) \\ 10x_1 + 12x_2 + 7x_3 \leq 35 & \\ 12x_2 + 6x_3 + 10x_4 \geq 18 & \\ x_2 \leq 2 & \\ x_1 \leq 1 & \\ x_i \geq 0 \quad \forall x_i, & \end{array}$$

che porta alla soluzione ottima $x_1 = 1, x_2 = 2, x_3 = 0.143$, corrispondente ad un valore della funzione obiettivo $z_{LP_3} = 13.286$. A partire dalla soluzione ottima del problema (R_3), si ricava il valore dell'upper bound $UB_3 = \lfloor 13.286 \rfloor = 13$.
Analogamente, per il nodo 4 si ottiene il problema continuo

$$\begin{array}{ll} \max 3x_1 + 5x_2 + 2x_3 - 5x_4 & \\ s.t. & (R_4) \\ 10x_1 + 12x_2 + 7x_3 \leq 35 & \\ 12x_2 + 6x_3 + 10x_4 \geq 18 & \\ x_2 \leq 2 & \\ x_1 \geq 2 & \\ x_i \geq 0 \quad \forall x_i, & \end{array}$$

avente soluzione ottima $x_1 = 2, x_2 = 1.25, x_3 = 0.3$, corrispondente ad un valore della funzione obiettivo $z_{LP_4} = 10.75$ ed upper bound $UB_4 = \lfloor 10.75 \rfloor = 10$.
La politica di visita dei nodi implica che il prossimo nodo da considerare sarà il nodo 3, avendo tale nodo l'upper bound più grande. L'albero di branch parziale al termine del nodo 1 è mostrato in figura 5.5.

Nodo 3: $x_2 \leq 2, x_1 \leq 1$. Il nodo non può essere chiuso per alcun criterio di chiusura, per cui si procede con il branch binario. La soluzione ottima del problema (R_3) presenta una sola variabile con valore non intero, la variabile $x_3 = 0.143$, per cui tale variabile è la variabile critica sulla quale effettuare il branch. Il nodo 3 genera quindi due figli, i nodi 5 e 6, aventi rispettivamente i vincoli aggiuntivi $x_3 \leq 0$ e $x_3 \geq 1$. Al fine di scegliere quale sia il prossimo nodo da considerare, occorre calcolare gli upper bound dei nodi appena creati.
Per il nodo 5, l'upper bound si ottiene risolvendo il problema

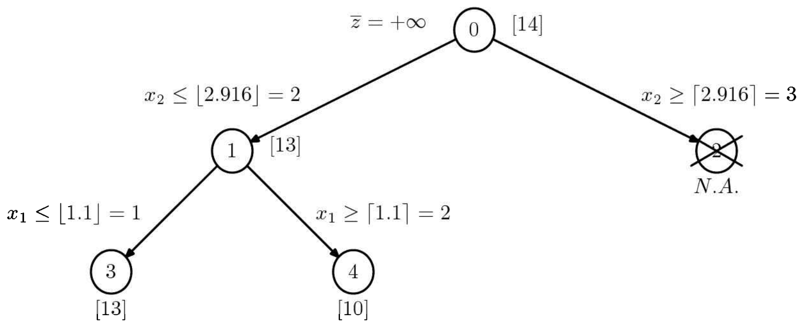


Figura 5.5: Esercizio 5.3: albero di branch parziale relativo al nodo 1

$$
\begin{array}{ll}
\max 3x_1 + 5x_2 + 2x_3 - 5x_4 & \\
s.t. & (R_5) \\
10x_1 + 12x_2 + 7x_3 \leq 35 & \\
12x_2 + 6x_3 + 10x_4 \geq 18 & \\
x_2 \leq 2 & \\
x_1 \leq 1 & \\
x_3 \leq 0 & \\
x_i \geq 0 \quad \forall x_i, &
\end{array}
$$

che porta alla soluzione ottima $x_1 = 1, x_2 = 2$, corrispondente ad un valore della funzione obiettivo $z_{LP_5} = 13$, che risulta quindi anche essere il valore dell'upper bound U_5.
Per il nodo 6, invece, il problema da risolvere è

$$
\begin{array}{ll}
\max 3x_1 + 5x_2 + 2x_3 - 5x_4 & \\
s.t. & (R_6) \\
10x_1 + 12x_2 + 7x_3 \leq 35 & \\
12x_2 + 6x_3 + 10x_4 \geq 18 & \\
x_2 \leq 2 & \\
x_1 \leq 1 & \\
x_3 \geq 1 & \\
x_i \geq 0 \quad \forall x_i, &
\end{array}
$$

con soluzione ottima $x_1 = 0.4, x_2 = 2, x_3 = 1$, $z_{LP_6} = 13.2$ e $U_6 = 13$.
La politica di visita dei nodi implica che il prossimo nodo da considerare sarà il nodo con l'upper bound maggiore, cioè upper bound pari a 13. I nodi che soddisfano tale criterio sono sia il nodo 5 che il nodo 6, per cui si effettua una scelta tra di essi in

base al criterio lessicografico. Il prossimo nodo da esaminare sarà pertanto il nodo 5. L'albero di branch parziale al termine del nodo 3 è mostrato in figura 5.6.

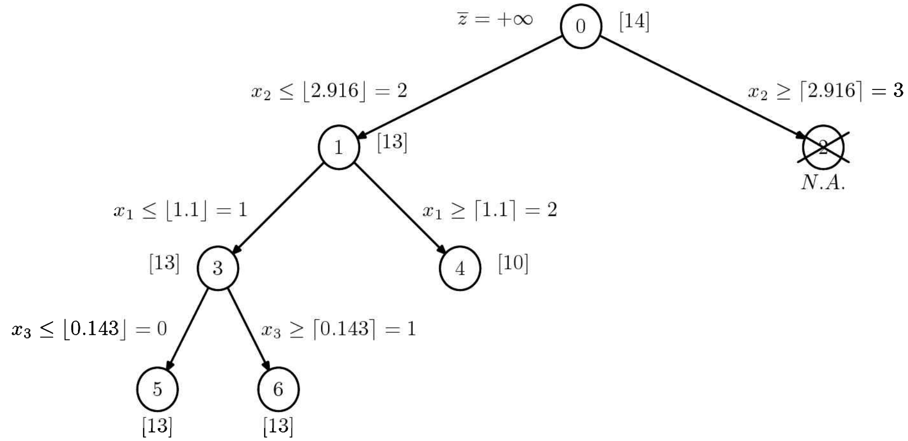


Figura 5.6: Esercizio 5.3: albero di branch parziale relativo al nodo 3

Nodo 5: $x_2 \leq 2, x_1 \leq 1, x_3 \leq 0$. Ispezionando la soluzione ottima continua si nota che essa è anche intera, per cui possiamo chiudere il nodo 5 per raggiunta ottimalità del problema di nodo. La soluzione ottima intera del nodo 5 è inoltre anche soluzione ammissibile del problema (P), per cui possiamo aggiornare il valore della soluzione ottimale come $\overline{z} = 13$ (vedi figura 5.7).
I nodi ancora aperti risultano esser i nodi 4 e 6. Avendo l'upper bound maggiore, il prossimo nodo da considerare è il nodo 6.

Nodo 6: $x_2 \leq 2, x_1 \leq 1, x_3 \geq 1$. Il nodo presenta un valore del rilassamento pari a 13. Essendo $UB_6 = 13 \leq \overline{z}$, il nodo può essere chiuso in quanto non migliorante.
L'unico nodo ancora aperto risulta essere il nodo 4, il quale verrà considerato come prossimo nodo da analizzare.

Nodo 4: $x_2 \leq 2, x_1 \geq 2$. Il nodo presenta un valore del rilassamento pari a 10. Essendo $UB_4 = 10 \leq \overline{z}$, il nodo può essere chiuso in quanto non migliorante.
Non essendoci ulteriori nodi aperti, l'algoritmo si considera terminato. La soluzione ottima al problema risulta avere funzione obiettivo pari a 13 e soluzione $x_1 = 1, x_2 = 2$.
L'albero di branch finale è rappresentato in figura 5.8.
Si supponga di cambiare la politica di branch dalla Best First alla Depth First. Si supponga inoltre che, dati i due nodi figli, ogni volta l'algoritmo scelga di esplorare

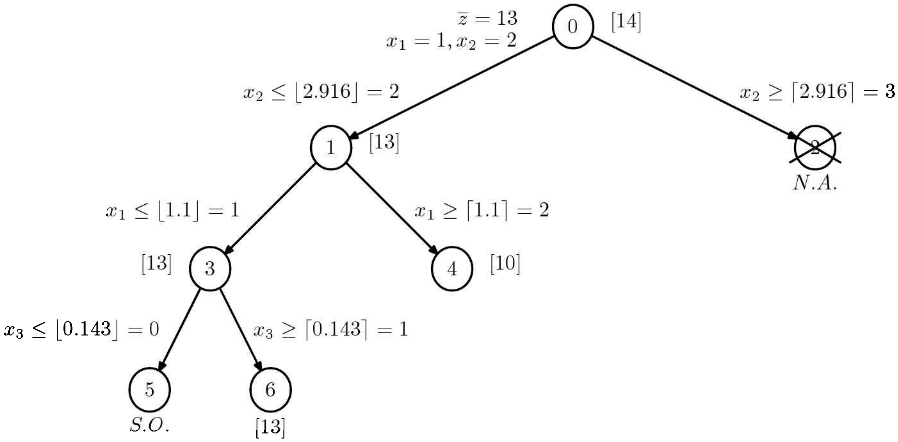


Figura 5.7: Esercizio 5.3: albero di branch parziale relativo al nodo 5

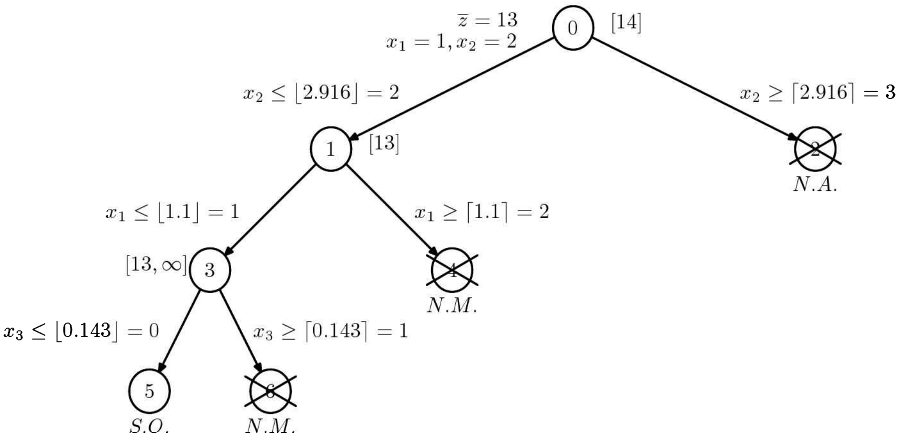


Figura 5.8: Esercizio 5.3: albero di branch finale

prima il nodo di sinistra. In tal caso si ottiene in nuovo albero che è, come struttura, uguale a quello in figura 5.9, ma presenta una diversa etichettatura dei nodi a causa della diversa politica di visita.
Supponiamo ora di mantenere la politica di visita dei nodi Depth First, ma di prediligere ora il nodo destro a quello sinistro. In tal caso avremo una visita completamente diversa, corrispondente all'albero in figura 5.10. Come si può notare in tal caso non solo cambia l'ordine di visita, ma anche il numero di nodi visitato, che sale a 36. Analogamente all'esempio citato, si possono avere casi inversi in cui è la politica Best First ad aprire una quantità maggiore di nodi. In generale, quindi, la politica da utilizzare è da valutare tipologia di problema per tipologia di problema, dipendendo essa dalla struttura del problema da risolvere e dalla relativa distribuzione delle soluzioni nell'albero di ricerca.

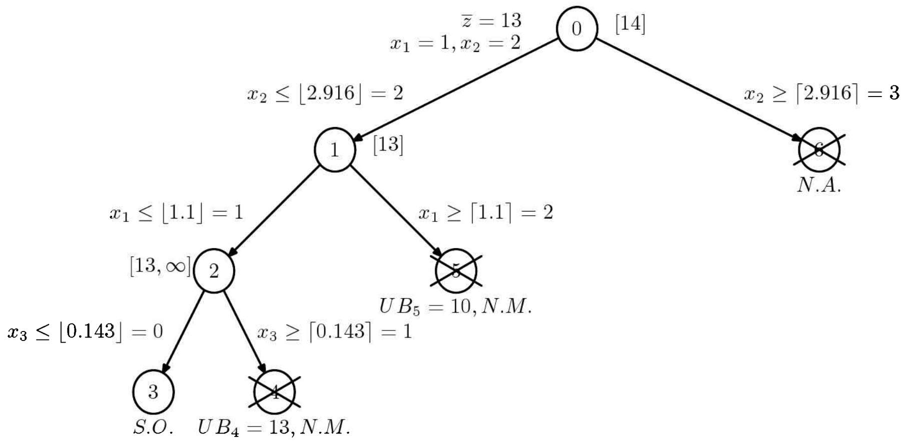


Figura 5.9: Esercizio 5.3: albero di branch finale nel caso di visita Depth First - navigazione a sinistra

ESEMPIO 5.4. Risolvere, mediante l'algoritmo del Branch & Bound, il seguente problema.

$$\begin{aligned}
&\max 2x_1 + x_2 + 3x_3 \\
&s.t. \\
&2x_2 + 3x_3 \le 7 \\
&2x_1 + x_2 + 2x_3 \le 8 \\
&3x_1 + 2x_2 + 3x_3 = 10 \\
&x_i \in \mathbb{Z}^+ \quad \forall x_i
\end{aligned}$$

Si utilizzi, per il calcolo del lower bound, il rilassamento continuo.

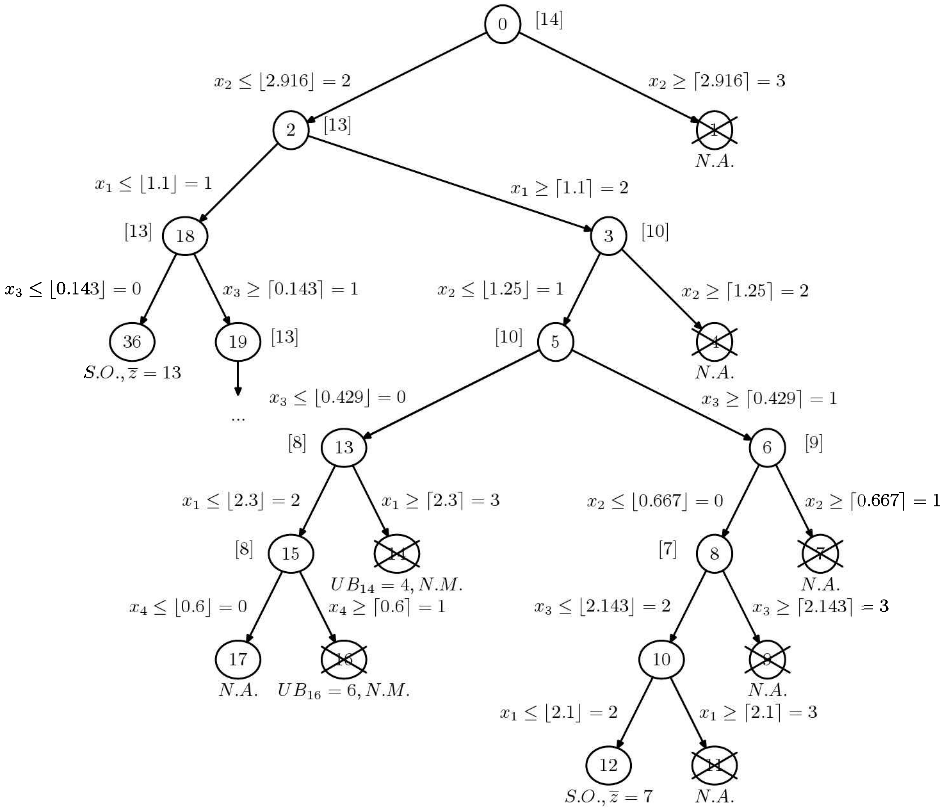


Figura 5.10: Esercizio 5.3: albero di branch finale nel caso di visita Depth First - navigazione a destra

Soluzione
Calcoliamo il rilassamento continuo (lower bound) al nodo radice con il metodo del simplesso. Portando il PLI in forma standard, si ottiene:

$$
\begin{aligned}
&\min -2x_1 - x_2 - 3x_3 \\
&2x_2 + 3x_3 + x_4 = 7 \\
&2x_1 + x_2 + 2x_3 + x_5 = 8 \\
&3x_1 + 2x_2 + 3x_3 = 10 \\
&x_i \in \mathbb{Z}^+ \quad \forall x_i
\end{aligned}
$$

Non è immediatamente visibile una soluzione di base, per cui si applica la fase I del simplesso:

	x_1	x_2	x_3	x_4	x_5	y	
x_4	0	2	3	1	0	0	7
x_5	2	1	2	0	1	0	8
y	$\boxed{3}$	2	3	0	0	1	10
	-3	-2	-3	0	0	0	-10

$\Rightarrow$

	x_1	x_2	x_3	x_4	x_5	y	
x_4	0	2	3	1	0	0	7
x_5	0	$-\frac{1}{3}$	0	0	1	$-\frac{2}{3}$	$\frac{4}{3}$
x_1	1	$\frac{2}{3}$	1	0	0	$\frac{1}{3}$	$\frac{10}{3}$
	0	0	0	0	0	1	0

Si ritorna quindi al problema di partenza, calcolando il lower bound al nodo radice:

	x_1	x_2	x_3	x_4	x_5	
x_4	0	2	3	1	0	7
x_5	0	$-\frac{1}{3}$	0	0	1	$\frac{4}{3}$
x_1	1	$\frac{2}{3}$	1	0	0	$\frac{10}{3}$
	-2	-1	-3	0	0	0

$\Rightarrow$

	x_1	x_2	x_3	x_4	x_5	
x_4	0	2	$\boxed{3}$	1	0	7
x_5	0	$-\frac{1}{3}$	0	0	1	$\frac{4}{3}$
x_1	1	$\frac{2}{3}$	1	0	0	$\frac{10}{3}$
	0	$\frac{1}{3}$	-1	0	0	$\frac{20}{3}$

	x_1	x_2	x_3	x_4	x_5	
x_3	0	$\frac{2}{3}$	1	$\frac{1}{3}$	0	$\frac{7}{3}$
x_5	0	$-\frac{1}{3}$	0	0	1	$\frac{4}{3}$
x_1	1	0	0	$-\frac{1}{3}$	0	1
	0	1	0	$\frac{1}{3}$	0	9

$$
\begin{aligned}
x_1 &= 1 \\
x_2 &= 0 \\
x_3 &= \tfrac{7}{3} \\
x_4 &= 0 \\
x_5 &= \tfrac{4}{3} \\
z &= -9.
\end{aligned}
$$

Il lower bound al nodo radice vale $LB_0 = -9$. La soluzione non è intera. Selezioniamo una variabile con valore frazionario e operiamo un branch su di essa. Nel nostro caso consideriamo la variabile x_3, generando due nodi. Nel nodo 1 imponiamo $x_3 \leq 2$ e nel nodo 2 $x_3 \geq 3$.

Analizziamo il nodo 1:

	x_1	x_2	x_3	x_4	x_5	x_6	
x_3	0	$\frac{2}{3}$	1	$\frac{1}{3}$	0	0	$\frac{7}{3}$
x_5	0	$-\frac{1}{3}$	0	0	1	0	$\frac{4}{3}$
x_1	1	0	0	$-\frac{1}{3}$	0	0	1
x_6	0	0	1	0	0	1	2
	0	1	0	$\frac{1}{3}$	0	0	9

Ricordando che la variabile x_3 è il base e che la sua colonna deve essere composta da un unico coefficiente unitario e tutti gli altri coefficienti a zero, al fine di mantenere tale struttura si effettua una operazione di pivot su $y_{1,3}$.

	x_1	x_2	x_3	x_4	x_5	x_6	
x_3	0	$\frac{2}{3}$	1	$\frac{1}{3}$	0	0	$\frac{7}{3}$
x_5	0	$-\frac{1}{3}$	0	0	1	0	$\frac{4}{3}$
x_1	1	0	0	$-\frac{1}{3}$	0	0	1
x_6	0	$-\frac{2}{3}$	0	$\boxed{-\frac{1}{3}}$	0	1	$-\frac{1}{3}$
	0	1	0	$\frac{1}{3}$	0	0	9

La soluzione è ottima ($r_i \geq 0, \forall i$) ma non ammissibile ($x_6 < 0$), per cui applichiamo il simplesso duale: $\varepsilon = \min\left\{\frac{1}{\frac{2}{3}}, \frac{\frac{1}{3}}{\frac{1}{3}}\right\} = \min\left\{\frac{3}{2}, 1\right\} = 1$

	x_1	x_2	x_3	x_4	x_5	x_6	
x_3	0	0	1	0	0	1	2
x_5	0	$-\frac{1}{3}$	0	0	1	0	$\frac{4}{3}$
x_1	1	$\frac{2}{3}$	0	0	0	-1	$\frac{4}{3}$
x_4	0	2	0	1	0	-3	1
	0	$\frac{1}{3}$	0	0	0	1	$\frac{26}{3}$

$$
\begin{aligned}
x_1 &= \tfrac{4}{3}\\
x_2 &= 0\\
x_3 &= 2\\
x_4 &= 1\\
x_5 &= \tfrac{4}{3}\\
x_6 &= 0\\
z &= -\tfrac{26}{3}.
\end{aligned}
$$

La soluzione non è intera. Si noti che il problema in questione è un problema a variabili intere con tutti i coefficienti della funzione obiettivo interi. E' dunque evidente che qualsiasi soluzione al problema assumerà valore della funzione obiettivo intero. E' allora possibile considerare come lower bound la parte intera superiore del valore della funzione obiettivo della soluzione ottima del problema continuo. Il lower bound del nodo 1 vale dunque vale $LB_1 = \lceil -\frac{26}{3} \rceil$=-8.

Nodo 2:
Aggiungiamo il vincolo $x_3 \geq 3$, che, portato in forma standard, diventa $x_3 - x_6 = 3$ ed ottimizziamo il tableau tramite il simplesso duale.

	x_1	x_2	x_3	x_4	x_5	x_6	
x_3	0	$\frac{2}{3}$	1	$\frac{1}{3}$	0	0	$\frac{7}{3}$
x_5	0	$-\frac{1}{3}$	0	0	1	0	$\frac{4}{3}$
x_1	1	0	0	$-\frac{1}{3}$	0	0	1
x_6	0	0	-1	0	0	1	-3
	0	1	0	$\frac{1}{3}$	0	0	9

	x_1	x_2	x_3	x_4	x_5	x_6	
x_3	0	$\frac{2}{3}$	1	$\frac{1}{3}$	0	0	$\frac{7}{3}$
x_5	0	$-\frac{1}{3}$	0	0	1	0	$\frac{4}{3}$
x_1	1	0	0	$-\frac{1}{3}$	0	0	1
x_6	0	$\frac{2}{3}$	0	$\boxed{\frac{1}{3}}$	0	1	$-\frac{2}{3}$
	0	1	0	$\frac{1}{3}$	0	0	9

Il nodo può essere chiuso per inammissibilità della soluzione. La situazione attuale è rappresentata in figura 5.11: l'unico nodo aperto rimane il nodo 1. Nella soluzione ottima del rilassamento continuo del problema a tale nodo, la variabile x_1 compare con valore frazionario. Operiamo un branch su di essa, generando i nodi 3 e 4.

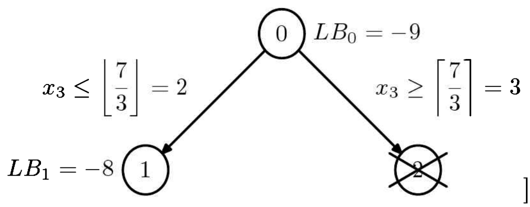


Figura 5.11: Albero di branch parziale relativo all'esercizio 5.4

Nodo 3:
Aggiungiamo il vincolo
$x_1 \leq 1 \rightarrow x_1 + x_7 = 1$.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
x_3	0	0	1	0	0	1	0	2
x_5	0	$-\frac{1}{3}$	0	0	1	0	0	$\frac{4}{3}$
x_1	1	$\frac{2}{3}$	0	0	0	-1	0	$\frac{4}{3}$
x_4	0	2	0	1	0	-3	0	1
x_7	1	0	0	0	0	0	1	1
	0	$\frac{1}{3}$	0	0	0	1	0	$\frac{26}{3}$

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
x_3	0	0	1	0	0	1	0	2
x_5	0	$-\frac{1}{3}$	0	0	1	0	0	$\frac{4}{3}$
x_1	1	$\frac{2}{3}$	0	0	0	-1	0	$\frac{4}{3}$
x_4	0	2	0	1	0	-3	0	1
x_7	0	$\boxed{-\frac{2}{3}}$	0	0	0	1	1	$-\frac{1}{3}$
	0	$\frac{1}{3}$	0	0	0	1	0	$\frac{26}{3}$

Applicando il simplesso duale, esce x_7 entra x_2

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
x_3	0	0	1	0	0	1	0	2
x_5	0	0	0	0	1	$-\frac{1}{2}$	$-\frac{1}{2}$	$\frac{3}{2}$
x_1	1	0	0	0	0	0	1	1
x_4	0	0	0	1	0	0	3	0
x_2	0	1	0	0	0	$-\frac{3}{2}$	$-\frac{3}{2}$	$\frac{1}{2}$
	0	0	0	0	0	$\frac{3}{2}$	$\frac{1}{2}$	$\frac{17}{3}$

La soluzione ottima è:

$$
\begin{gathered}
x_1 = 1 \\
x_2 = \frac{1}{2} \\
x_3 = 2 \\
x_4 = 0 \\
x_5 = \frac{3}{2} \\
x_6 = 0 \\
x_7 = 0 \\
z = -\frac{17}{3} \\
LB_3 = \lceil z \rceil = -8.
\end{gathered}
$$

Nodo 4:
Aggiungiamo il vincolo

$x_1 \geq 2 \rightarrow x_1 - x_7 = 2.$

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
x_3	0	0	1	0	0	1	0	2
x_5	0	$-\frac{1}{3}$	0	0	1	0	0	$\frac{4}{3}$
x_1	1	$\frac{2}{3}$	0	0	0	-1	0	$\frac{4}{3}$
x_4	0	2	0	1	0	-3	0	1
x_7	1	0	0	0	0	0	-1	-2
	0	$\frac{1}{3}$	0	0	0	1	0	$\frac{26}{3}$

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
x_3	0	0	1	0	0	1	0	2
x_5	0	$-\frac{1}{3}$	0	0	1	0	0	$\frac{4}{3}$
x_1	1	$\frac{2}{3}$	0	0	0	-1	0	$\frac{4}{3}$
x_4	0	2	0	1	0	-3	0	1
x_7	0	$\frac{2}{3}$	0	0	0	$\boxed{-1}$	1	$-\frac{1}{3}$
	0	$\frac{1}{3}$	0	0	0	1	0	$\frac{26}{3}$

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
x_3	0	$\frac{2}{3}$	1	0	0	0	1	$\frac{4}{3}$
x_5	0	$-\frac{1}{3}$	0	0	1	0	0	$\frac{4}{3}$
x_1	1	0	0	0	0	0	-1	2
x_4	0	0	0	1	0	0	-3	3
x_6	0	$-\frac{2}{3}$	0	0	0	1	-1	$\frac{2}{3}$
	0	1	0	0	0	0	1	8

$$
\begin{aligned}
x_1 &= 2 \\
x_2 &= 0 \\
x_3 &= \tfrac{4}{3} \\
x_4 &= 3 \\
x_5 &= \tfrac{4}{3} \\
x_6 &= \tfrac{2}{3} \\
x_7 &= 0 \\
z &= -8 \\
LB_4 &= -8.
\end{aligned}
$$

La situazione attuale è rappresentata in figura 5.12: sono aperti i nodi 3 e 4. Consideriamo il nodo 3 e operiamo un branch su una delle variabili che compaiono con valore frazionario. Nel nostro caso scegliamo x_3 e generiamo i nodi 5 e 6.

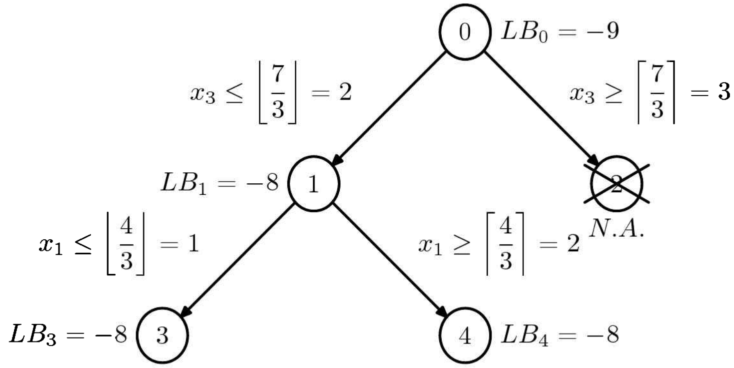


Figura 5.12: Albero di branch parziale relativo all'esercizio 5.4

Nodo 5: Aggiungiamo il vincolo $x_2 \leq 0$ e calcoliamo il lower bound:

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	
x_3	0	0	1	0	0	1	0	0	2
x_5	0	0	0	0	1	$-\frac{1}{2}$	$-\frac{1}{2}$	0	$\frac{3}{2}$
x_1	1	0	0	0	0	0	1	0	1
x_4	0	0	0	1	0	0	3	0	0
x_2	0	1	0	0	0	$-\frac{3}{2}$	$-\frac{3}{2}$	0	$\frac{1}{2}$
x_8	0	1	0	0	0	0	0	1	0
	0	0	0	0	0	$\frac{3}{2}$	$\frac{1}{2}$	0	$\frac{17}{3}$

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	
x_3	0	0	1	0	0	1	0	0	2
x_5	0	0	0	0	1	$-\frac{1}{2}$	$-\frac{1}{2}$	0	$\frac{3}{2}$
x_1	1	0	0	0	0	0	1	0	1
x_4	0	0	0	1	0	0	3	0	0
x_2	0	1	0	0	0	$-\frac{3}{2}$	$-\frac{3}{2}$	0	$\frac{1}{2}$
x_8	0	0	0	0	0	$\frac{3}{2}$	$\frac{3}{2}$	1	$-\frac{1}{2}$
	0	0	0	0	0	$\frac{3}{2}$	$\frac{1}{2}$	0	$\frac{17}{3}$

Visto che è verificata la condizione di illimitatezza duale, il nodo può essere chiuso per non ammissibilità.

Nodo 6: Aggiungiamo il vincolo $x_2 \geq 1$ e calcoliamo il lower bound

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	
x_3	0	0	1	0	0	1	0	0	2
x_5	0	0	0	0	1	$-\frac{1}{2}$	$-\frac{1}{2}$	0	$\frac{3}{2}$
x_1	1	0	0	0	0	0	1	0	1
x_4	0	0	0	1	0	0	3	0	0
x_2	0	1	0	0	0	$-\frac{3}{2}$	$-\frac{3}{2}$	0	$\frac{1}{2}$
x_8	0	-1	0	0	0	0	0	1	-1
	0	0	0	0	0	$\frac{3}{2}$	$\frac{1}{2}$	0	$\frac{17}{3}$

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	
x_3	0	0	1	0	0	1	0	0	2
x_5	0	0	0	0	1	$-\frac{1}{2}$	$-\frac{1}{2}$	0	$\frac{3}{2}$
x_1	1	0	0	0	0	0	1	0	1
x_4	0	0	0	1	0	0	3	0	0
x_2	0	1	0	0	0	$-\frac{3}{2}$	$-\frac{3}{2}$	0	$\frac{1}{2}$
x_8	0	0	0	0	0	$-\frac{3}{2}$	$\boxed{-\frac{3}{2}}$	1	$-\frac{1}{2}$
	0	0	0	0	0	$\frac{3}{2}$	$\frac{1}{2}$	0	$\frac{17}{3}$

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	
x_3	0	0	1	0	0	1	0	0	2
x_5	0	0	0	0	1	0	0	$-\frac{1}{3}$	$\frac{5}{3}$
x_1	1	0	0	0	0	-1	0	$\frac{2}{3}$	$\frac{2}{3}$
x_4	0	0	0	1	0	$\boxed{-3}$	0	2	-1
x_2	0	1	0	0	0	0	0	-1	1
x_7	0	0	0	0	0	1	1	$-\frac{3}{2}$	$\frac{1}{3}$
	0	0	0	0	0	1	0	$\frac{1}{3}$	$\frac{25}{3}$

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	
x_3	0	0	1	$\frac{1}{3}$	0	0	0	$\frac{2}{3}$	$\frac{5}{3}$
x_5	0	0	0	0	1	0	0	$-\frac{1}{3}$	$\frac{5}{3}$
x_1	1	0	0	$-\frac{1}{3}$	0	0	0	0	1
x_6	0	0	0	$-\frac{1}{3}$	0	1	0	$-\frac{2}{3}$	$\frac{1}{3}$
x_2	0	1	0	0	0	0	0	-1	1
x_7	0	0	0	$\frac{1}{3}$	0	0	1	0	0
	0	0	0	$\frac{1}{3}$	0	0	0	1	8

$$
\begin{aligned}
x_1 &= 1 \\
x_2 &= 1 \\
x_3 &= \tfrac{5}{3} \\
x_4 &= 0 \\
x_5 &= \tfrac{5}{3} \\
x_6 &= \tfrac{1}{3} \\
x_7 &= 0 \\
x_8 &= 0 \\
z &= -8 \\
LB_6 &= -8.
\end{aligned}
$$

A questo punto sono aperti i nodi 4 e 6 (figura 5.13). Scegliamo il nodo 4 e operiamo un branch sulla variabile x_3.

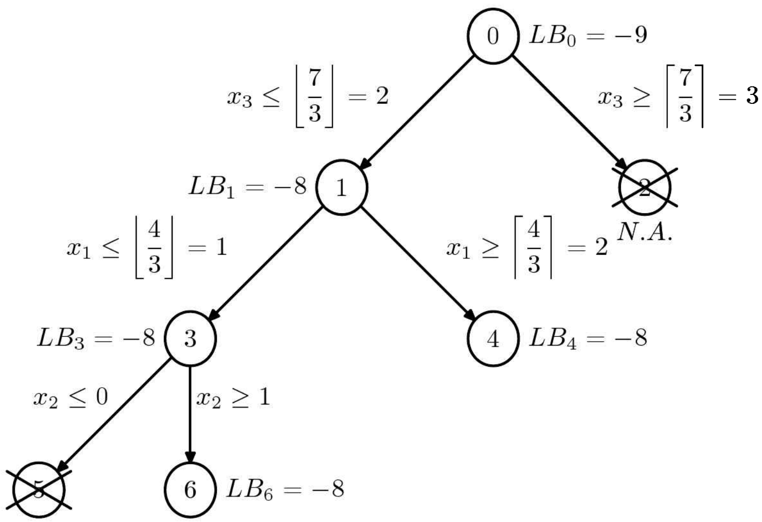


Figura 5.13: Albero di branch parziale relativo all'esercizio 5.4

Nodo 7: Aggiungiamo al problema relativo al nodo padre (nodo 4) il vincolo $x_3 \leq 1$.

			x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_3	0	$\frac{2}{3}$	1	0	0	0	1	0	$\frac{4}{3}$	
x_5	0	$-\frac{1}{3}$	0	0	1	0	0	0	$\frac{4}{3}$	
x_1	1	0	0	0	0	0	-1	0	2	
x_4	0	0	0	1	0	0	-3	0	3	
x_6	0	$-\frac{2}{3}$	0	0	0	1	-1	0	$\frac{2}{3}$	
x_8	0	0	1	0	0	0	0	1	1	
	0	1	0	0	0	0	1	0	8	

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	
x_3	0	$\frac{2}{3}$	1	0	0	0	1	0	$\frac{4}{3}$
x_5	0	$-\frac{1}{3}$	0	0	1	0	0	0	$\frac{4}{3}$
x_1	1	0	0	0	0	0	-1	0	2
x_4	0	0	0	1	0	0	-3	0	3
x_6	0	$-\frac{2}{3}$	0	0	0	1	-1	0	$\frac{2}{3}$
x_8	0	$-\frac{2}{3}$	0	0	0	0	$\boxed{-1}$	1	$-\frac{1}{3}$
	0	1	0	0	0	0	1	0	8

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	
x_3	0	0	1	0	0	0	0	1	1
x_5	0	$-\frac{1}{3}$	0	0	1	0	0	0	$\frac{4}{3}$
x_1	1	$\frac{2}{3}$	0	0	0	0	0	-1	$\frac{7}{3}$
x_4	0	2	0	1	0	0	0	-3	4
x_6	0	0	0	0	0	1	0	-1	1
x_7	0	$\frac{2}{3}$	0	0	0	0	1	-1	$\frac{1}{3}$
	0	$\frac{1}{3}$	0	0	0	0	0	1	$\frac{23}{3}$

$$x_1 = \tfrac{7}{3}$$
$$x_2 = 0$$
$$x_3 = 1$$
$$x_4 = 4$$
$$x_5 = \tfrac{4}{3}$$
$$x_6 = 1$$
$$x_7 = \tfrac{1}{3}$$
$$x_8 = 0$$
$$z = -\tfrac{23}{3}$$
$$LB_7 = \lceil -\tfrac{23}{3} \rceil = -7.$$

Nodo 8: Aggiungiamo al problema del nodo padre (nodo 4) il vincolo $x_3 \geq 2$.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	
x_3	0	$\frac{2}{3}$	1	0	0	0	1	0	$\frac{4}{3}$
x_5	0	$-\frac{1}{3}$	0	0	1	0	0	0	$\frac{4}{3}$
x_1	1	0	0	0	0	0	-1	0	2
x_4	0	0	0	1	0	0	-3	0	3
x_6	0	$-\frac{2}{3}$	0	0	0	1	-1	0	$\frac{2}{3}$
x_8	0	0	-1	0	0	0	0	1	-2
	0	1	0	0	0	0	1	0	8

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	
x_3	0	$\frac{2}{3}$	1	0	0	0	1	0	$\frac{4}{3}$
x_5	0	$-\frac{1}{3}$	0	0	1	0	0	0	$\frac{4}{3}$
x_1	1	0	0	0	0	0	-1	0	2
x_4	0	0	0	1	0	0	-3	0	3
x_6	0	$-\frac{2}{3}$	0	0	0	1	-1	0	$\frac{2}{3}$
x_8	0	$\frac{2}{3}$	0	0	0	0	1	1	$-\frac{2}{3}$
	0	1	0	0	0	0	1	0	8

Soluzione non ammissibile. Il nodo può essere chiuso.
Rimangono aperti i nodi 6 e 7 (figura 5.14). Consideriamo per primo, secondo una politica best first, il nodo con minimo valore di lower bound, ovvero il nodo 6. Il branch viene effettuato sulla variabile x_3.

Nodo 9: Aggiungiamo il vincolo $x_3 \leq 1$.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	
x_3	0	0	1	$\frac{1}{3}$	0	0	0	$\frac{2}{3}$	0	$\frac{5}{3}$
x_5	0	0	0	0	1	0	0	$-\frac{1}{3}$	0	$\frac{5}{3}$
x_1	1	0	0	$-\frac{1}{3}$	0	0	0	0	0	1
x_6	0	0	0	$-\frac{1}{3}$	0	1	0	$-\frac{2}{3}$	0	$\frac{1}{3}$
x_2	0	1	0	0	0	0	0	-1	0	1
x_7	0	0	0	$\frac{1}{3}$	0	0	1	0	0	0
x_9	0	0	1	0	0	0	0	0	1	1
	0	0	0	$\frac{1}{3}$	0	0	0	1	0	8

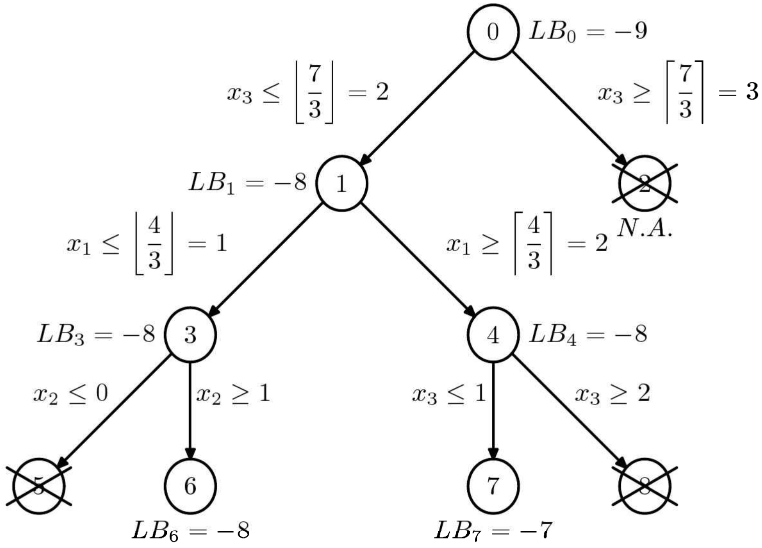


Figura 5.14: Albero di branch parziale relativo all'esercizio 5.4

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	
x_3	0	0	1	$\frac{1}{3}$	0	0	0	$\frac{2}{3}$	0	$\frac{5}{3}$
x_5	0	0	0	0	1	0	0	$-\frac{1}{3}$	0	$\frac{5}{3}$
x_1	1	0	0	$-\frac{1}{3}$	0	0	0	0	0	1
x_6	0	0	0	$-\frac{1}{3}$	0	1	0	$-\frac{2}{3}$	0	$\frac{1}{3}$
x_2	0	1	0	0	0	0	0	-1	0	1
x_7	0	0	0	$\frac{1}{3}$	0	0	1	0	0	0
x_9	0	0	0	$\boxed{-\frac{1}{3}}$	0	0	0	$-\frac{2}{3}$	1	$-\frac{2}{3}$
	0	0	0	$\frac{1}{3}$	0	0	0	1	0	8

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	
x_3	0	0	1	0	0	0	0	0	0	1
x_5	0	0	0	0	1	0	0	$-\frac{1}{3}$	0	$\frac{5}{3}$
x_1	1	0	0	0	0	0	0	$\frac{2}{3}$	-1	$\frac{5}{3}$
x_6	0	0	0	0	0	1	0	0	0	1
x_2	0	1	0	0	0	0	0	-1	0	1
x_7	0	0	0	0	0	0	0	$\boxed{-\frac{2}{3}}$	1	$-\frac{2}{3}$
x_4	0	0	0	1	0	0	1	2	-3	2
	0	0	0	0	0	0	0	$\frac{1}{3}$	1	$\frac{22}{3}$

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	
x_3	0	0	1	0	0	0	0	0	0	1
x_5	0	0	0	0	1	0	$-\frac{1}{2}$	0	$-\frac{1}{2}$	2
x_1	1	0	0	0	0	0	1	0	0	1
x_6	0	0	0	0	0	1	0	0	0	1
x_2	0	1	0	0	0	0	0	-1	0	2
x_8	0	0	0	0	0	0	$-\frac{3}{2}$	$\boxed{1}$	$-\frac{3}{2}$	1
x_4	0	0	0	1	0	0	3	0	0	0
	0	0	0	0	0	0	$\frac{1}{2}$	0	$\frac{3}{2}$	7

$$
\begin{aligned}
x_1 &= 1\\
x_2 &= 2\\
x_3 &= 1\\
x_4 &= 0\\
x_5 &= 2\\
x_6 &= 1\\
x_7 &= 0\\
x_8 &= 1\\
x_9 &= 0\\
z &= -7.
\end{aligned}
$$

Abbiamo trovato una soluzione intera. Memorizziamo $z^* = -7$.

Nodo 10: Aggiungiamo il vincolo $x_3 \geq 2$.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	
x_3	0	0	1	$\frac{1}{3}$	0	0	0	$\frac{2}{3}$	0	$\frac{5}{3}$
x_5	0	0	0	0	1	0	0	$-\frac{1}{3}$	0	$\frac{5}{3}$
x_1	1	0	0	$-\frac{1}{3}$	0	0	0	0	0	1
x_6	0	0	0	$-\frac{1}{3}$	0	1	0	$-\frac{2}{3}$	0	$\frac{1}{3}$
x_2	0	1	0	0	0	0	0	-1	0	1
x_7	0	0	0	$\frac{1}{3}$	0	0	1	0	0	0
x_9	0	0	-1	0	0	0	0	0	1	-2
	0	0	0	$\frac{1}{3}$	0	0	0	1	0	8

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	
x_3	0	0	1	$\frac{1}{3}$	0	0	0	$\frac{2}{3}$	0	$\frac{5}{3}$
x_5	0	0	0	0	1	0	0	$-\frac{1}{3}$	0	$\frac{5}{3}$
x_1	1	0	0	$-\frac{1}{3}$	0	0	0	0	0	1
x_6	0	0	0	$-\frac{1}{3}$	0	1	0	$-\frac{2}{3}$	0	$\frac{1}{3}$
x_2	0	1	0	0	0	0	0	-1	0	1
x_7	0	0	0	$\frac{1}{3}$	0	0	1	0	0	0
x_9	0	0	0	$\frac{1}{3}$	0	0	0	$\frac{2}{3}$	1	$\frac{2}{3}$
	0	0	0	$\frac{1}{3}$	0	0	0	1	0	8

Soluzione non ammissibile: il nodo può essere chiuso.

A questo punto rimane aperto il solo nodo 7, ma visto che $LB_7 \geq z^* = -7$ non è possibile trovare soluzioni miglioranti. Posso dunque chiudere anche tale nodo (figura 5.15).
La soluzione ottima del problema intero è dunque quella trovata al nodo 9, ovvero:

$$x_1 = 1$$
$$x_2 = 2$$
$$x_3 = 1$$

funzione obiettivo del problema (max): $z = 7$.

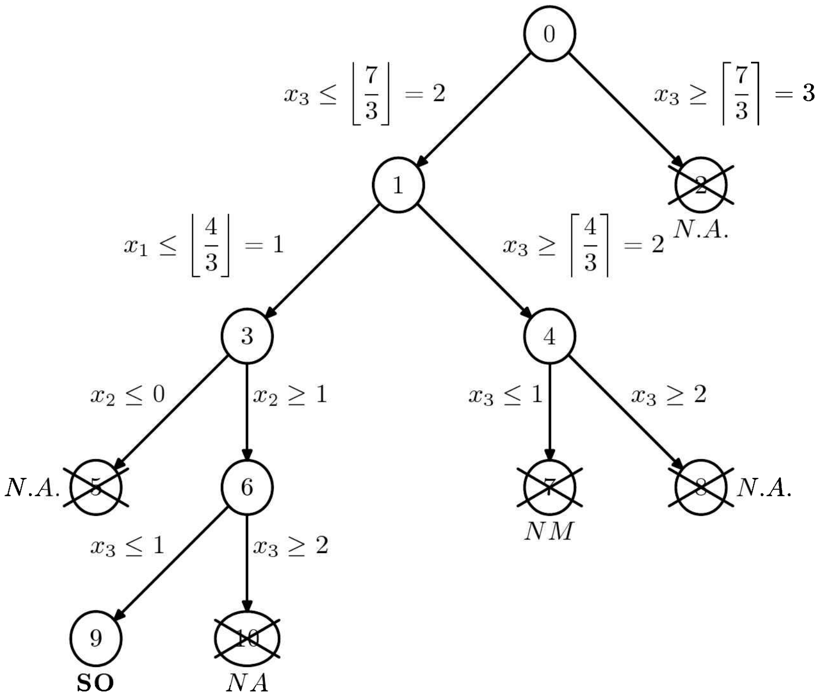


Figura 5.15: Albero di branch relativo all'esercizio 5.4

5.2 Branch & Bound per KP0/1

5.2.1 Richiami teorici

Il problema dello zaino con variabili 0/1 è formulato in PLI nel modo seguente:

$$\begin{array}{ll} \max & \sum_{i=1}^{n} p_i x_i \\ & \sum_{i=1}^{n} w_i x_i \leq W \\ & x_i \in \{0,1\} \quad i = 1, ..., n \end{array}$$

dove la variabile $x_i \in \{0,1\}$ rappresenta la presenza ($x_i = 1$) o l'assenza ($x_i = 0$) nello zaino dell'oggetto i.mo mentre W, p_i, w_i, $i = 1, ..., n$, sono interi positivi.
Un metodo per calcolare un lower bound (soluzione ammissibile) ed un upper bound (soluzione ottima del rilassamento continuo) per questo problema sono descritti nel seguente algoritmo:

Function KP0/1 - Lower bound e Upper bound
Si ordinino le variabili in ordine decrescente del rapporto $\frac{p_i}{w_i}$.
Si carichi lo zaino progressivamente con gli oggetti secondo l'ordine indicato. Sia s il primo oggetto a sforare la capacità dello zaino.
Il lower bound è la somma dei profitti degli oggetti che è stato possibile inserire nello zaino:

$$\begin{array}{ll} x_i = 1 & \text{per } i = 1, ..., s-1 \\ x_i = 0 & \text{per } i = s, ..., n \end{array}$$

L'upper bound è calcolabile attraverso la seguente assegnazione di valori alle variabili:

$$\begin{array}{ll} x_i = 1 & \text{per } i = 1, ..., s-1 \\ x_i = 0 & \text{per } i = s+1, ..., n \\ x_s = \frac{W^*}{w_s} & \text{con } W^* = W - \sum_{i=1}^{s-1} w_i. \end{array}$$

E' possibile calcolare un upper bound più stringente attraverso la seguente formula:

$$UB^* = \sum_{i=1}^{s-1} p_i + \max\{W^* \frac{p_{s+1}}{w_{s+1}}, p_s - (w_s - W^*) \frac{p_{s-1}}{w_{s-1}}\}.$$

nota come "Upper bound di Martello & Toth".
Si ricorda inoltre che se tutte le variabili sono intere (binarie nel nostro caso) ed i coefficienti della funzione obiettivo sono interi, la funzione obiettivo di qualsiasi soluzione sarà intera. Sotto queste ipotesi è dunque possibile approssimare l'upper

bound con il suo valore intero inferiore, ottenedo l'espressione seguente.

$$UB^* = \sum_{i=1}^{s-1} p_i + \max\{\lfloor W^* \frac{p_{s+1}}{w_{s+1}} \rfloor, \lfloor p_s - (w_s - W^*) \frac{p_{s-1}}{w_{s-1}} \rfloor\}.$$

5.2.2 Esercizi risolti

ESEMPIO 5.5. Risolvere mediante il metodo del Branch & Bound il seguente problema dello zaino:

$$\max 7x_1 + 5x_2 + 3x_3 + 4x_4 + 5x_5 + 6x_6 + 8x_7$$

soggetto a

$$3x_1 + 10x_2 + 3x_3 + 6x_4 + x_5 + 2x_6 + 8x_7 \leq 15$$

$$x_i \in \{0, 1\} \quad \forall i$$

Per la soluzione di questo esercizio, così come dei successivi, si farà uso di entrambi i metodi introdotti per il calcolo degli UB: il primo metodo (metodo con rilassamento continuo), che prevede di saturare la capacità dello zaino con parti frazionarie dell'oggetto costituente in quel nodo la variabile critica. Viceversa, il metodo con upper bound Martello & Toth, che ricorre ad una determinata formula specifica. Come primo passo procediamo ad ordinare le variabili secondo valori decrescenti del rapporto $\frac{p_i}{w_i}$.

x_i	x_1	x_2	x_3	x_4	x_5	x_6	x_7
p_i	7	5	3	4	5	6	8
w_i	3	10	3	6	1	2	8
$\frac{p_i}{w_i}$	2.33	0.5	1	0.67	5	3	1

Dalla tabella di ottiene che l'ordine con il quale devono essere considerate le variabili risulta: x_5, x_6, x_1, x_3, x_7, x_4, x_2.

Metodo con rilassamento continuo

Nodo 0: Si ottiene: $UB = 27$ e $LB = 21$ con variabile critica x_7. Si pone quindi $z = 21$ e si procede ad esplorare il nodo.

`Branch del nodo 0.`

Nodo 1: $x_7 = 0$. Si ottiene: $UB = LB = 25$ per cui si è ottenuto un nodo foglia di cui se ne memorizza la soluzione ($z' = 25$).

Nodo 2: $x_7 = 1$. Si ottiene: $UB = 27$ e $LB = 26$ con variabile critica x_3. Si aggiorna la soluzione ammissibile, $z = 26$, e si procede ad esplorare il nodo.

`Branch del nodo 2.`

Nodo 3: $x_7 = 1$, $x_3 = 0$. Si ottiene: $UB = \lfloor 26.67 \rfloor = 26$ e $LB = 26$, per cui si ottiene un nodo foglia e la soluzione intera corrispondente ($z'' = 26$) non è memorizzata, in quanto ne è stata trovata una equivalente al nodo 2.

Nodo 4: $x_7 = 1$, $x_3 = 1$. Si ottiene: $UB = \lfloor 24.33 \rfloor = 24$ e $LB = 22$ con variabile critica x_1. Il nodo non verrà esplorato perchè il suo UB è minore della soluzione ammissibile trovata.

Alla fine quindi si ottiene $z^* = 26$, individuata dal nodo 2 con la seguente soluzione:

$$x_1 = 1,\ x_2 = 0,\ x_3 = 0,\ x_4 = 0,\ x_5 = 1,\ x_6 = 1,\ x_7 = 1.$$

L'albero di ricerca completo è illustrato in figura 5.16.

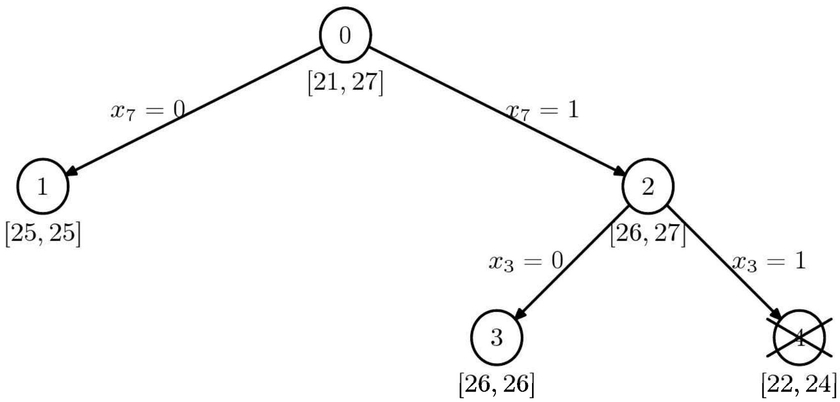


Figura 5.16: Albero di branch relativo all'esercizio 5.5

Metodo con upper bound Martello & Toth

Nodo 0: Si ottiene: $UB^* = 27$ e $LB = 21$ con variabile critica x_7. Si pone quindi $z = 21$ e si procede ad esplorare il nodo.

`Branch del nodo 0.`

Nodo 1: $x_7 = 0$. Si ottiene: $UB^* = LB = 25$ per cui si è ottenuto un nodo foglia di cui se ne memorizza la soluzione ($z' = 25$).

Nodo 2: $x_7 = 1$. Si ottiene: $UB^* = \lfloor 26.66 \rfloor = 26$ e $LB = 26$, per cui otteniamo un nodo foglia e memorizziamo la soluzione corrispondente ($z'' = 26$).

Alla fine quindi si ottiene $z^* = 26$, individuata dal nodo 2 con la seguente soluzione:

$$x_1 = 1,\ x_2 = 0,\ x_3 = 0,\ x_4 = 0,\ x_5 = 1,\ x_6 = 1,\ x_7 = 1.$$

L'albero di ricerca completo è illustrato in figura 5.17.

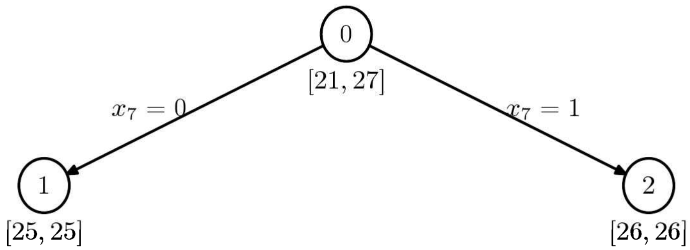


Figura 5.17: Albero di branch relativo all'esercizio 5.5

ESEMPIO 5.6.

Risolvere il seguente problema dello zaino mediante branch and bound.

$$\max 9x_1 + 8x_2 + 7x_3 + 6x_4 + 3x_5$$

soggetto a

$$2x_1 + 3x_2 + 5x_3 + 4x_4 + 9x_5 \leq 12$$
$$x_i \in \{0, 1\}\ \forall i$$

Soluzione

Come primo passo si procede ad ordinare le variabili secondo valori decrescenti del rapporto $\frac{p_i}{w_i}$.

x_i	x_1	x_2	x_3	x_4	x_5
p_i	9	8	7	6	3
w_i	2	3	5	4	9
$\frac{p_i}{w_i}$	4.5	2.33	1.4	1.5	0.33

Dalla tabella di ottiene che l'ordine con il quale devono essere considerate le variabili risulta: x_1, x_2, x_4, x_3, x_5.

Metodo con rilassamento continuo

Nodo 0: Si ottiene: $UB = \lfloor 27.2 \rfloor = 27$ e $LB = 23$ con variabile critica x_3. Si pone quindi $z = 23$ e si procede ad esplorare il nodo.

`Branch del nodo 0.`

Nodo 1: $x_3 = 0$. Si ottiene: $UB = 24$ e $LB = 23$ con variabile critica x_5.

Nodo 2: $x_3 = 1$. Si ottiene: $UB = 27$ e $LB = 24$ con variabile critica x_4. Si aggiorna la soluzione ammissibile, $z = 24$, e si procede ad esplorare il nodo.

`Branch del nodo 2 (strategia Best First).`

Nodo 3: $x_3 = 1$, $x_4 = 0$. Si ottiene: $UB = \lfloor 24.67 \rfloor = 24$ e $LB = 24$ per cui otteniamo un nodo foglia e memorizziamo la soluzione corrispondente ($z' = 24$).

Nodo 4: $x_3 = 1$, $x_4 = 1$. Si ottiene: $UB = \lfloor 24.67 \rfloor = 24$ e $LB = 22$ con variabile critica x_2.

A questo punto risulta inutile procede all'esplorazione del nodo 4 così pure come del nodo 1 perchè sicuramente non potranno portare a soluzioni miglioranti.

Si è dunque ottenuto l'albero di ricerca illustrato in 5.18 e la soluzione $z^* = 24$ individuata al nodo 5:

$$x_1 = 1,\ x_2 = 1,\ x_3 = 1,\ x_4 = 0,\ x_5 = 0.$$

Metodo con upper bound Martello & Toth

Nodo 0: Si ottiene: $UB^* = 27$ e $LB = 23$ con variabile critica x_3. Si pone quindi $z = 23$ e si procede ad esplorare il nodo.

`Branch del nodo 0.`

Nodo 1: $x_3 = 0$. Si ottiene: $UB^* = LB = 23$. per cui si è ottenuto un nodo foglia di cui se ne memorizza la soluzione ($z' = 23$).

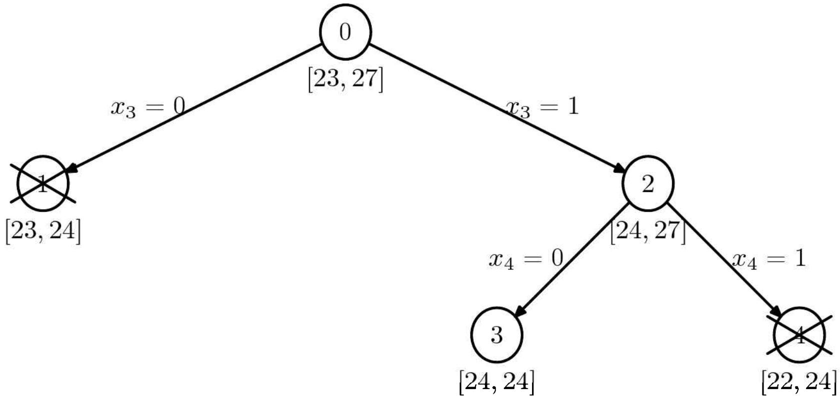


Figura 5.18: Albero di branch relativo all'esercizio 5.6

Nodo 2: $x_3 = 1$. Si ottiene: $UB^* = \lfloor 24.67 \rfloor = 24$. e $LB = 24$ per cui si è determinato un nodo foglia di cui si provvede a memorizzare il valore ($z'' = 24$).

Alla fine quindi si ottiene $z^* = 24$, individuata dal nodo 2 con la seguente soluzione (l'albero di ricerca è rappresentato in 5.19):

$$x_1 = 1,\ x_2 = 1,\ x_3 = 1,\ x_4 = 0,\ x_5 = 0.$$

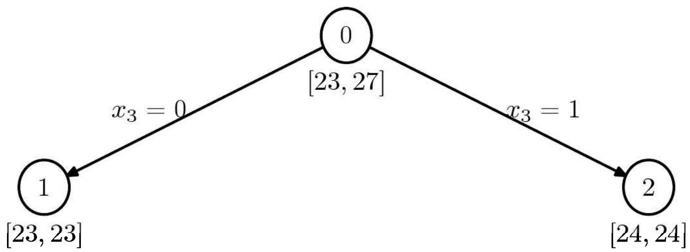


Figura 5.19: Albero di branch relativo all'esercizio 5.6

ESEMPIO 5.7. Risolvere con il metodo del Branch & Bound.

$$\max 4x_1 + 9x_2 + 6x_3 + 7x_4 + 5x_5 + y$$

soggetto a

$$3x_1 + 7x_2 + 6x_3 + 9x_4 + 10x_5 + 3y \leq 20$$

$$x_i \in \{0, 1\} \ \forall i, \ \ y \geq 0$$

Soluzione
Come primo passo si procede ad ordinare le variabili secondo valori decrescenti del rapporto $\frac{p_i}{w_i}$.

x_i	x_1	x_2	x_3	x_4	x_5	y
p_i	4	9	6	7	5	1
w_i	3	7	6	9	10	3
$\frac{p_i}{w_i}$	1.34	1.28	1	0.78	0.5	0.34

Le variabili risultano ordinate. La presenza della variabile reale reale y può essere gestita considerando il fatto che è la variabile meno conveniente, per cui in qualsiasi soluzione al problema tale variabile verrà utilizzata semplicemente per riempire la parte inutilizzata della capacità dello zaino. Si applica dunque il branch and bound per $KP0/1$ normalmente e ad ogni soluzione ottenuta si assegna ad y il valore $\frac{W^*}{3}$. Ovviamente, a causa della presenza di una variabile non intera, non è piu' possibile approssimare all'intero inferiore l'upper bound di ogni nodo.

Metodo con rilassamento continuo

Nodo 0: Si ottiene: $UB = 22.11$ e $LB = 19$ con variabile critica x_4. Si pone quindi $z = 19$ e si procede ad esplorare il nodo.

`Branch del nodo 0.`

Nodo 1: $x_4 = 0$. Si ottiene: $UB = 21$ e $LB = 19$ con variabile critica x_5.

Nodo 2: $x_4 = 1$. Si ottiene: $UB = 21$ e $LB = 20$ con variabile critica x_3.

`Branch del nodo 1.`

Nodo 3: $x_4 = 0$, $x_5 = 0$. Si ottiene: $UB = LB = 20.34$. Si memorizza la soluzione trovata ($z' = 20.34$)

Nodo 4: $x_4 = 0$, $x_5 = 1$. Si ottiene: $UB = LB = 18$. La soluzione appena trovata viene scartata.

`Branch del nodo 2.`

Nodo 5: $x_4 = 1$, $x_3 = 0$. Si ottiene: $UB = 20.5$ e $LB = 20$ con variabile critica x_5.

Nodo 6: $x_4 = 1$, $x_3 = 1$. Si ottiene: $UB = 19.57$ e $LB = 17$ Il nodo viene chiuso e non viene esplorato in quanto non può portare a nessun miglioramento della soluzione.

`Branch del nodo 5.`

Nodo 7: $x_4 = 1$, $x_3 = 0$, $x_5 = 0$. Si ottiene: $UB = LB = 20.34$. Si procede a memorizzare anche questa soluzione.

Nodo 8: $x_4 = 1$, $x_3 = 0$, $x_5 = 1$. Si ottiene: $UB = 13.34$ e $LB = 12$. Per cui il nodo non verrà esplorato.

Si è dunque ottenuto l'albero di ricerca in figura 5.20 e $z^* = 20.34$, individuata al nodo 3 ed al nodo 7 con le seguenti soluzioni:

$$x_1 = 1, x_2 = 1, x_3 = 1, x_4 = 0, x_5 = 0, y = 1.34.$$
$$x_1 = 1, x_2 = 1, x_3 = 0, x_4 = 1, x_5 = 0, y = 0.34.$$

Metodo con upper bound Martello & Toth

Nodo 0: Si ottiene: $UB^* = 21$ e $LB = 19$ con variabile critica x_4. Si pone quindi $z = 19$ e si procede ad esplorare il nodo.

`Branch del nodo 0.`

Nodo 1: $x_4 = 0$. Si ottiene: $UB^* = 20.34$ e $LB = 19$ con variabile critica x_5.

Nodo 2: $x_4 = 1$. Si ottiene: $UB^* = 20.5$ e $LB = 20$ con variabile critica x_3. Si pone quindi $z = 20$.

`Branch del nodo 2.`

Nodo 3: $x_4 = 1$, $x_3 = 0$. Si ottiene: $UB^* = 20.34$ e $LB = 20$ con variabile critica x_5.

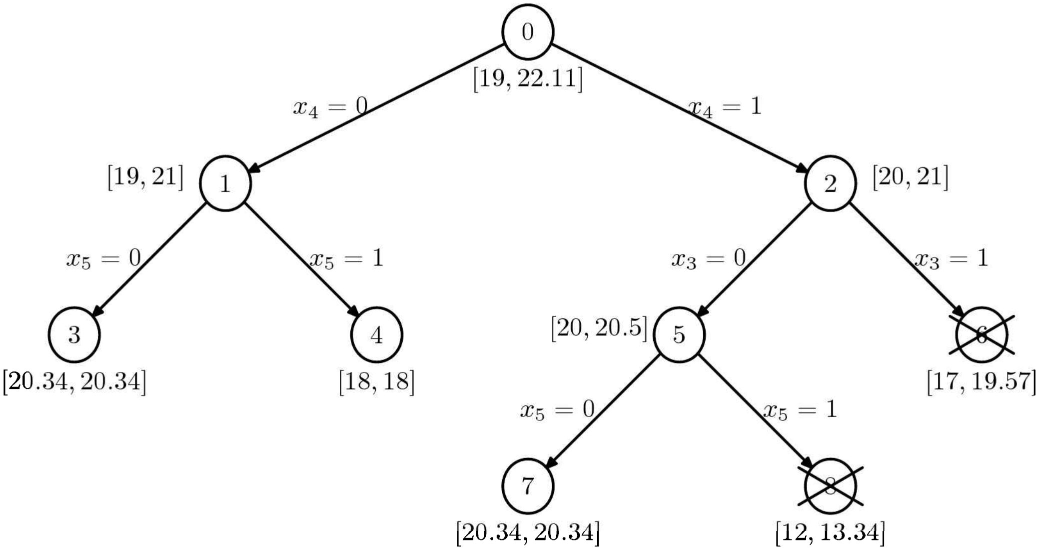


Figura 5.20: Albero di branch relativo all'esercizio 5.7

Nodo 4: $x_4 = 1$, $x_3 = 1$. Si ottiene: $UB^* = 19.33$ e $LB = 17$.Il nodo quindi non verrà esplorato.

`Branch del nodo 1.`

Nodo 5: $x_4 = 0$, $x_5 = 0$. Si ottiene: $UB^* = LB = 20.34$. Si memorizza questa soluzione ($z' = 20.34$).

Nodo 6: $x_4 = 0$, $x_5 = 1$. Si ottiene: $UB^* = LB = 18$. Il nodo è una foglia e la soluzione associata verrà scartata.

Risulta inutile a questo punto esplorare il nodo 3 per cui si è ottenuto l'albero di ricerca in figura 5.21 e $z^* = 20.34$, individuata al dal nodo 5:

$$x_1 = 1,\, x_2 = 1,\, x_3 = 1,\, x_4 = 0,\, x_5 = 0,\, y = 1.34.$$

ESEMPIO 5.8. Risolvere con il metodo di Branch & Bound:

$$\max 6x_1 + 4x_2 + 9x_3 + 5x_4 + x_5$$

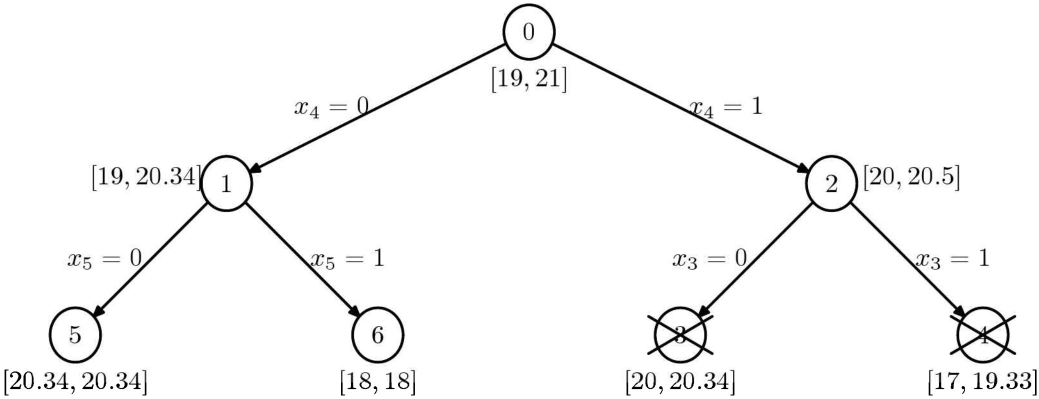


Figura 5.21: Albero di branch relativo all'esercizio 5.7

soggetto a

$$2x_1 + 3x_2 + 10x_3 + 7x_4 + 2x_5 \leq 14$$

$$x_i \in \{0, 1\} \quad \forall i$$

Soluzione
Come primo passo si procede ad ordinare le variabili secondo valori decrescenti del rapporto $\frac{p_i}{w_i}$.

x_i	x_1	x_2	x_3	x_4	x_5
p_i	6	4	9	5	1
w_i	2	3	10	7	2
$\frac{p_i}{w_i}$	3	1.34	0.9	0.71	0.5

Come si può notare dalla tabella come già le variabili risultano ordinate.

Metodo con rilassamento continuo

Nodo 0: Si ottiene: $UB = \lfloor 18.11 \rfloor = 18$ e $LB = 10$ con variabile critica x_3. Si pone quindi $z = 10$ e si procede ad esplorare il nodo.

`Branch del nodo 0.`

Nodo 1: $x_3 = 0$. Si ottiene: $UB = LB = 16$. Si procede quindi a memorizzare questa soluzione ($z' = 16$).

Nodo 2: $x_3 = 1$. Si ottiene: $UB = \lfloor 17.67 \rfloor = 17$ e $LB = 15$ con variabile critica x_2.

`Branch del nodo 2.`

Nodo 3: $x_3 = 1$, $x_2 = 0$. Si ottiene: $UB = \lfloor 16.42 \rfloor = 16$ e $LB = 15$ con variabile critica x_4. Il nodo quindi non verrà esplorato perchè non può portare a soluzioni migliori rispetto a quella trovate.

Nodo 4: $x_3 = 1$, $x_2 = 1$. Si ottiene: $UB = 16$ e $LB = 13$.Il nodo non verrà esplorato perchè non potrà portare nessun miglioramento alla soluzione.

Alla fine quindi si ottiene $z^* = 16$, individuata dal nodo 1 con la seguente soluzione:

$$x_1 = 1,\ x_2 = 1,\ x_3 = 0,\ x_4 = 1,\ x_5 = 1.$$

L'albero di ricerca ottenuto è rappresentato in figura 5.22.

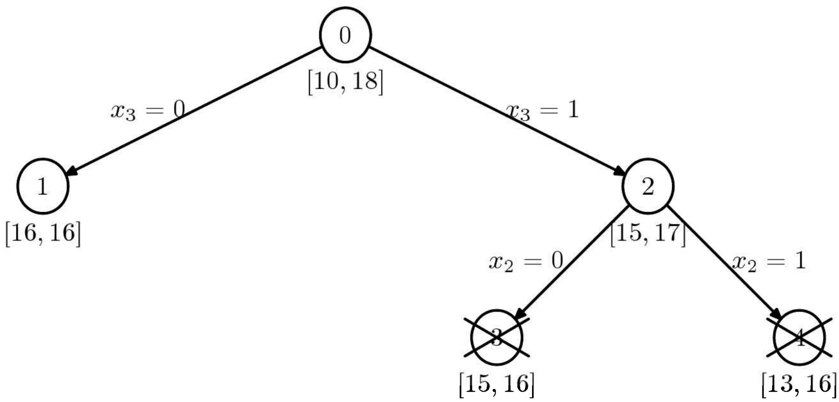


Figura 5.22: Albero di branch relativo all'esercizio 5.8

Metodo con upper bound Martello & Toth

Nodo 0: Si ottiene: $UB^* = \lfloor 17.66 \rfloor = 17$ e $LB = 10$ con variabile critica x_3. Si pone quindi $z = 10$ e si procede ad esplorare il nodo.

`Branch del nodo 0.`

Nodo 1: $x_3 = 0$. Si ottiene: $UB^* = LB = 16$. Si procede quindi a memorizzare questa soluzione ($z' = 16$).

Nodo 2: $x_3 = 1$. Si ottiene: $UB^* = \lfloor 16.42 \rfloor = 16$ e $LB = 15$ con variabile critica x_2.

Risulta pertanto inutile andare a esplorare il nodo 2.
Si è dunque ottenuto l'albero di ricerca in figura 5.23 e la soluzione $z^* = 16$, individuata al nodo 1:

$$x_1 = 1,\, x_2 = 1,\, x_3 = 0,\, x_4 = 1,\, x_5 = 1.$$

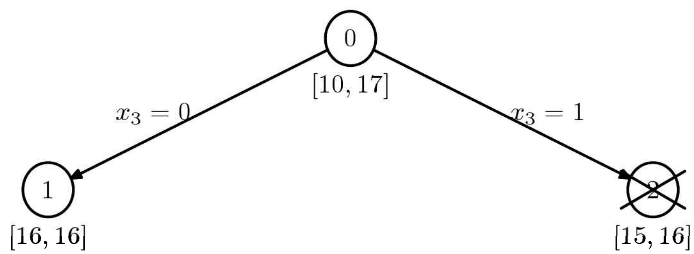


Figura 5.23: Albero di branch relativo all'esercizio 5.8

ESEMPIO 5.9. Applicando il metodo del branch and bound, risolvere:

$$\max 9x_1 + 16x_2 + 18x_3 + 8x_4 + 7x_5$$

soggetto a

$$2x_1 + 4x_2 + 5x_3 + 7x_4 + 8x_5 \leq 15$$

$$x_i \in \{0, 1\}\ \forall i$$

Soluzione
Questo esercizio può essere risolto senza effettuare alcun calcolo, considerando un semplice *criterio di dominanza* per il problema dello zaino: se esistono due variabili x_i è x_j tali che $p_i \geq p_j$ e $w_i \leq w_j$ si dice che *i domina j.* In tal caso in qualsiasi soluzione ottima al problema, valgono le seguenti relazioni:

- Se $x_i = 0$ allora $x_j = 0$.
- Se $x_j = 1$ allora $x_i = 1$.

Intuitivamente la prima relazione afferma che se non viene inserito l'oggetto piu' conveniente nello zaino, sicuramente non sarà inserito neppure quello meno conveniente. La seconda relazione afferma invece che se viene inserito l'oggetto meno conveniente nello zaino, dovrà necessariamente essere stato inserito anche quello piu' conveniente. Nell'esercizio in questione si possono individuare le seguenti relazioni di dominanza:

- x_1 domina x_4 ed x_5.
- x_2 domina x_4 ed x_5.
- x_3 domina x_4 ed x_5.
- x_4 domina x_5.

Nella soluzione al problema, se ponessimo $x_5 = 1$ dovremmo per le relazioni di dominanza, porre a 1 anche x_1, x_2, x_3 ed x_4, generando una soluzione non ammissibile. Dunque dovrà necessariamente essere $x_5 = 0$.
Ora consideriamo la variabile x_4. Se assegnassimo $x_4 = 1$ dovremmo porre a 1 anche x_1, x_2 ed x_3, generando nuovamente una soluzione non ammissibile. Dunque dovrà necessariamente essere $x_5 = 0$.
Rimangono dunque le variabili x_1, x_2 ed x_3, alle quali può essere assegnato valore 1. Si ottiene dunque $z^* = 43$ con la seguente soluzione:

$$x_1 = 1,\ x_2 = 1,\ x_3 = 1,\ x_4 = 0,\ x_5 = 0.$$

ESEMPIO 5.10. Si voglia risolvere il seguente problema Knapsack 0/1:

$$\begin{array}{ll} \max & 9x_1 + 11x_2 + 2x_3 + 2x_4 + 3x_5 + x_6 \\ & 3x_1 + 5x_2 + x_3 + x_4 + 4x_5 + 7x_6 \leq 12 \\ & x_i \in \{0, 1\}, \forall i \end{array}$$

Soluzione
Le variabili sono già ordinate in ordine decrescente del rapporto $\dfrac{p_i}{w_i}$.

Si supponga di avere predeterminato con una semplice procedura costruttiva la soluzione ammissibile di partenza ($x_1 = 1$, $x_2 = 1$, $x_3 = 0$, $x_4 = 0$, $x_5 = 1$, $x_6 = 0$) di valore $LB_0 = 23$ dove $LB_0 = LB^*$ rappresenta anche il *lower bound corrente.* Si calcoli l'upper bound dato dal rilassamento continuo del problema. Si ottiene $UB_0 = 25.5$ dove la variabile critica $x_s = x_5$. Si calcoli ora UB_1. Si ottiene $UB_1 = \lfloor 24.28 \rfloor = 24$. A questo punto, prima di effettuare l'operazione di branch, è possibile cercare di verificare se sia possibile determinare a priori il valore di alcune variabili. Si valutano prima le variabili il cui rapporto $\frac{p_i}{w_i}$ è maggiore di quello della variabile critica: se, ponendo a 0 una di queste variabili, si ottiene, tramite rilassamento continuo, un upper bound minore del lower bound corrente, è possibile fissare univocamente a 1 quella variabile. Allo stesso modo, se, ponendo a 1 una variabile con rapporto $\frac{p_i}{w_i}$ minore di quello della variabile critica, l'upper bound trovato è minore del lower bound corrente, è possibile fissare univocamente a 0 quella variabile.
Nell'esempio si ha

$$\begin{array}{ll} UB_1 = 18.14 < LB_0 & \Longrightarrow \quad x_1 = 1 \\ UB_2 = 16.42 < LB_0 & \Longrightarrow \quad x_2 = 1 \\ UB_3 = 24.25 > LB_0 & \text{nessun assegnamento per } x_3 \\ UB_4 = 24.25 > LB_0 & \text{nessun assegnamento per } x_4 \\ UB_6 = 14.4 < LB_0 & \Longrightarrow \quad x_6 = 0. \end{array}$$

Avendo determinato il valore di 3 delle 6 variabili, il problema si riduce al seguente sottoproblema

$$\begin{array}{ll} \max & 2x_3 + 2x_4 + 3x_5 \\ & x_3 + x_4 + 4x_5 \leq 4 \\ & x_i \in \{0,1\}, \ \forall i. \end{array}$$

Per questo nuovo problema è quasi immediato determinare la soluzione ottima che è data da $x_3 = 1, x_4 = 1$ e $x_5 = 0$. Unendo questo risultato con la parte di soluzione ottenuta prima si ottiene come soluzione ottima del problema la soluzione ($x_1 = 1$, $x_2 = 1$, $x_3 = 1$, $x_4 = 1$, $x_5 = 0$ e $x_6 = 0$) di valore 24.

5.3 Rilassamento lagrangiano

5.3.1 Richiami teorici

Il rilassamento lagrangiano è uno dei metodi più utilizzati per calcolare un lower bound (in caso di problemi di minimo) per un problema di programmazione intera .

Si consideri seguente problema:

$$(P) \quad \begin{array}{l} \min c^T x \\ Ax \geq b \\ Bx \geq d \\ x \in \{0,1\} \end{array}$$

Supponiamo che eliminando i vincoli $Ax \geq b$ il problema diventi risolvibile in tempo polinomiale. Il *rilassamento Lagrangiano* consiste nell'eliminazione di tali vincoli dal problema e dal loro inserimento in funzione obiettivo, al fine di penalizzare soluzioni non ammissibili per il problema (P) di partenza. Si costruisce quindi il seguente problema *Lagrangiano Primale (LP)*

$$(LP) \quad \begin{array}{l} z(\lambda) = \min c^T x - \lambda(Ax - b) \\ Bx \geq d \\ x \in \{0,1\} \\ \lambda \geq 0 \end{array}$$

dove il vettore λ è detto vettore dei moltiplicatori lagrangiani e pesa il contributo dato in funzione obiettivo della inammissibilità dei singoli vincoli. Dato un valore qualsiasi ai moltiplicatori $\lambda \geq 0$, la soluzione ottima $z(\lambda)$ del problema (LP) è un lower bound per il problema (P).

Per ottenere un lower bound che sia il più stringente possibile, si vuole ottimizzare il valore del vettore λ nella funzione Lagrangiana, ossia si vuole risolvere il seguente problema *Lagrangiano Duale (LD)*

$$(LD) \quad \begin{array}{l} \max z(\lambda) \\ \lambda \geq 0 \end{array}$$

Al fine di ottimizzare i valori associati ai moltiplicatori lagrangiani, è possibile utilizzare un metodo iterativo chiamato metodo del subgradiente. Data una qualche inizializzazione λ_0 del vettore dei moltiplicatori, viene risolto all'ottimo il problema LP e viene calcolato il corrispondente vettore $x(\lambda)$. Si calcola il gradiente della funzione obiettivo di LD rispetto ai moltiplicatori λ nell'intorno di $x(\lambda)$ e il vettore dei moltiplicatori viene modificato nella direzione del gradiente stesso. Il procedimento generalmente itera fino a quando non raggiunge una convergenza od un limite predeterminato sul numero totale di iterazioni.

Function Algoritmo del subgradiente
$\lambda = \lambda_0$
$z(LB) = -\infty$
$z(UB) = \infty$
while $TerminationTest ==$ **false do**
  Sia $x(\lambda)$ la soluzione ottima del problema LP, e $z(\lambda)$ la sua funzione obiettivo.
  if $z(\lambda) > z(LB)$ **then**
    $z(LB) = z(\lambda)$

end if
if $x(\lambda)$ è ammissibile per il problema (P) con f.o. $z_{x(\lambda)} < z(UB)$ **then**
$z(UB) = z_{x(\lambda)}$
end if
Calcolare il subgradiente S della f.o. di (LD) in $x(\lambda)$.
$\lambda = \lambda + TS$.
end while

return LB

I componenti del vettore subgradiente S della f.o. del problema (LD) sono:
$S_i = \frac{\partial}{\partial \lambda_i}(c^T x - \lambda(Ax - b)) = b_i - \sum_j a_{ij} x_j$

Il parametro T indica la lunghezza dello step di aggiornamento dei moltiplicatori, e viene normalmente ridotto con il passare delle iterazioni. Una sua tipica caratterizzazione è la seguente:

$T = \frac{t(z(UB) - z(\lambda))}{\|S\|^2}$
$t \in [0, 2]$
$t = t/2$ ogni k iterazioni ($k = 20$).

Infine, l'algoritmo viene normalmente fatto terminare quando un numero limite di iterazioni è stato raggiunto, quando si è trovato l'ottimo ($z(UB) = z(LB)$) o quando il gap tra lower e upper bound è sceso sotto ad una certa soglia.

5.3.2 Esercizi risolti

ESEMPIO 5.11. Calcolare il rilassamento lagrangiano del seguente problema a variabili binarie, supponendo di rilassare tutti i vincoli e utilizzando il metodo del subgradiente per ottimizzare il problema lagrangiano duale. Si supponga di utilizzare come valori iniziali $t = \frac{1}{7}, \lambda = (1, 1), z(UB) = 4$ e di effettuare due iterazioni dell'algoritmo.

$$\min 4x_1 - 3x_2 + 4x_3 - x_4$$

soggetto a

$$x_1 + x_2 + 2x_3 + x_4 \geq 2$$

$$2x_1 + 3x_2 - x_3 + x_4 \geq 5$$

$$x_i \in \{0, 1\} \quad \forall i$$

Soluzione
Il problema lagrangiano risultante dal rilassamento dei due vincoli è:

$$\min 4x_1 - 3x_2 + 4x_3 - x_4 - \lambda_1(x_1 + x_2 + 2x_3 + x_4 - 2) - \lambda_2(2x_1 + 3x_2 - x_3 + x_4 - 5)$$

soggetto a

$$x_i \in \{0,1\} \quad i = 1..4 \lambda_j \geq 0 \; j = 1,2$$

Prima iterazione

Sostituendo $\lambda = (1,1)$ e raccogliendo si ottiene il problema lagrangiano:

$$\min x_1 - 7x_2 + 3x_3 - 3x_4 + 7$$

soggetto a

$$x_i \in \{0,1\} \quad i = 1..4$$

La soluzione ottima è immediatamente ricavabile mettendo a uno le variabili che hanno un coeffieciente negativo e a zero le altre. Dunque:

$$x_1(\lambda) = 0, x_2(\lambda) = 1, x_3(\lambda) = 0, x_4(\lambda) = 1, z(\lambda) = -3$$

Si noti che tale soluzione non è soluzione ammissibile del problema di partenza (sostituendo $x(\lambda) = (0,1,0,1)$ nei vincoli del problema primale il secondo dei due vincoli non è verificato).

Calcoliamo il vettore subgradiente:

$S_1 = 2 - x_1(\lambda) - x_2(\lambda) - 2x_3(\lambda) - x_4(\lambda) = 2 - 1 - 1 = 0$
$S_2 = 5 - 2x_1(\lambda) - 3x_2(\lambda) + x_3(\lambda) - x_4(\lambda) = 5 - 3 - 1 = 1.$

e lo step di aggiornamento dei moltiplicatori:

$T = \frac{t(z(UB) - z(\lambda)}{\|S\|^2} = \frac{\frac{1}{7}(4-(-3))}{(\sqrt{0^2+1^2})^2} = 1$

I nuovi moltiplicatori saranno dunque:

$\lambda_1 = \lambda_1 + TS_1 = 1 + 1 \cdot 0 = 1$
$\lambda_2 = \lambda_2 + TS_2 = 1 + 1 \cdot 1 = 2$

Seconda iterazione

Sostituendo $\lambda = (1, 2)$ si ottiene il nuovo problema lagrangiano:

$$\min -x_1 - 10x_2 + 4x_3 - 4x_4 + 12$$

soggetto a

$$x_i \in \{0, 1\} \quad i = 1..4$$

La soluzione ottima è nuovamente ricavabile mettendo a uno le variabili che hanno un coeffieciente negativo, ovvero:

$$x_1 = 1, x_2 = 1, x_3 = 0, x_4 = 1, z(\lambda) = -3$$

La soluzione trovata è in questo caso ammissibile anche per il problema primale (i due vincoli sono soddisfatti), con valore della funzione obiettivo $z = 4x_1 - 3x_2 + 4x_3 - x_4 = 0$.
Dunque $z(UB) = 0, z(LB) = z(\lambda) = -3$.
Calcoliamo il vettore subgradiente:

$S_1 = 2 - x_1(\lambda) - x_2(\lambda) - 2x_3(\lambda) - x_4(\lambda) = 2 - 1 - 1 - 1 = -1$
$S_2 = 5 - 2x_1(\lambda) - 3x_2(\lambda) + x_3(\lambda) - x_4(\lambda) = 5 - 3 - 2 - 1 = -1.$

e lo step di aggiornamento dei moltiplicatori:

$T = \frac{t(z(UB)-z(\lambda)}{\|S\|^2} = \frac{\frac{1}{7}(0-(-3))}{(\sqrt{(-1)^2+(-1)^2})^2} = \frac{3}{14}$

I nuovi moltiplicatori saranno dunque:

$\lambda_1 = \lambda_1 + TS_1 = 1 + \frac{3}{14} \cdot (-1) = \frac{11}{14}$
$\lambda_2 = \lambda_2 + TS_2 = 1 + \frac{3}{14} \cdot (-1) = \frac{25}{14}$

Avendo raggiunto la fine della seconda iterazione, l'esercizio è completato.

ESEMPIO 5.12. Calcolare il rilassamento lagrangiano del seguente problema a variabili binarie, supponendo di rilassare tutti i vincoli e utilizzando il metodo del subgradiente per ottimizzare il problema lagrangiano duale. Si supponga di utilizzare come valori iniziali $t = \frac{1}{5}, \lambda = (1, 1), z(UB) = 16$ e di effettuare due iterazioni dell'algoritmo.

$$\min 4x_1 + 5x_2 + 3x_3 + 4x_4$$

soggetto a

$$3x_1 + x_2 + 2x_4 \geq 4$$

$$4x_1 + 2x_3 + x_4 \geq 5$$
$$x_i \in \{0,1\} \quad \forall i$$

Soluzione
Il problema lagrangiano risultante dal rilassamento dei due vincoli è:

$$\min 4x_1 + 5x_2 + 3x_3 + 4x_4 - \lambda_1(3x_1 + x_2 + 2x_4 - 4) - \lambda_2(4x_1 + 2x_3 + x_4 - 5)$$

soggetto a

$$x_i \in \{0,1\} \quad i = 1..4\lambda_j \geq 0 \; j = 1,2$$

Prima iterazione

Sostituendo $\lambda = (1,1)$ si ottiene:

$$\min -3x_1 + 4x_2 + x_3 + x_4 + 9$$

soggetto a

$$x_i \in \{0,1\} \quad i = 1..4$$

La soluzione ottima è immediatamente ricavabile mettendo a uno le variabili che hanno un coeffieciente negativo:

$$x_1(\lambda) = 1, x_2(\lambda) = x_3(\lambda) = x_4(\lambda) = 0, z(\lambda) = -3 + 9 = 6$$

Tale soluzione non è soluzione ammissibile del problema di partenza e non fornisce dunque contributi per l'upper bound. Calcoliamo il vettore subgradiente, lo step di aggiornamento dei moltiplicatori e i moltiplicatori per la prossima iterazione:

$S_1 = 4 - 3x_1(\lambda) - x_2(\lambda) - 2x_4(\lambda) = 4 - 3 = 1$
$S_2 = 5 - 4x_1(\lambda) - 2x_3(\lambda) - x_4(\lambda) = 5 - 4 = 1.$

$T = \frac{t(z(UB) - z(\lambda)}{\|S\|^2} = \frac{\frac{1}{5}(16-6)}{(\sqrt{1^2+1^2})^2} = 1$

$\lambda_1 = \lambda_1 + TS_1 = 1 + 1 \cdot 1 = 2$
$\lambda_2 = \lambda_2 + TS_2 = 1 + 1 \cdot 1 = 2$

Seconda iterazione

Sostituendo $\lambda = (2,2)$ si ottiene il nuovo problema lagrangiano:

$$\min -10x_1 + 3x_2 - x_3 - 2x_4 + 18$$

soggetto a

$$x_i \in \{0,1\} \quad i = 1..4$$

la cui soluzione ottima è:

$$x_1 = 1, x_2 = 0, x_3 = 1, x_4 = 1, z(\lambda) = -13 + 18 = 5$$

La soluzione trovata è in questo caso ammissibile anche per il problema primale (i due vincoli sono soddisfatti), con valore della funzione obiettivo $z = 11$.
Dunque $z(UB) = 11, z(LB) = z(\lambda) = 5$.
Calcoliamo il vettore subgradiente:

$S_1 = 4 - 3x_1(\lambda) - x_2(\lambda) - 2x_4(\lambda) = 4 - 3 - 2 = -1$
$S_2 = 5 - 4x_1(\lambda) - 2x_3(\lambda) - x_4(\lambda) = 5 - 4 - 2 - 1 = -2.$

e lo step di aggiornamento dei moltiplicatori:

$T = \frac{t(z(UB)-z(\lambda))}{\|S\|^2} = \frac{\frac{1}{5}(11-5)}{(\sqrt{(-1)^2+(-2)^2})^2} = \frac{6}{25}$

I nuovi moltiplicatori saranno dunque:

$\lambda_1 = \lambda_1 + TS_1 = 2 + \frac{6}{25} \cdot (-1) = \frac{44}{25}$
$\lambda_2 = \lambda_2 + TS_2 = 2 + \frac{6}{25} \cdot (-2) = \frac{38}{25}$

Avendo raggiunto la fine della seconda iterazione, l'esercizio è completato.

5.4 Metodo della Programmazione Dinamica

5.4.1 Richiami teorici

Consideriamo come esempio di applicazione del metodo della programmazione dinamica il problema dello zaino a variabili logiche KP0/1:

$$\begin{array}{ll} \max & \sum_{i=1}^{n} p_i x_i \\ & \sum_{i=1}^{n} w_i x_i \leq W \\ & x_i \in \{0,1\} \quad i = 1, ..., n \end{array}$$

Si consideri la seguente famiglia di problemi:

$$[\mathrm{KP}_i(\mathrm{E})] \quad \begin{array}{ll} F_j(E) = & \max \sum_{i=j}^{n} p_i \cdot x_i \\ & \sum_{i=j}^{n} w_i \cdot x_i \leq W - E \\ & x_i \in \{0,1\} \; i = j, ..., n \end{array}$$

Ogni problema di tale famiglia considera solo un sottoinsieme delle variabili del problema originale (da j a n) ed è parametrizzata da E, definita *variabile di stato.* Si osservi che il problema di partenza è un membro di tale famiglia di problemi, per $j = 1$ ed $E = 0$. Il metodo della programmazione dinamica per KP0/1 opera come segue:

Function Programmazione dinamica per KP0/1
for all $E \in (0..W))$ **do**

$$F_n(E) = \begin{cases} 0 & \text{per } E > W - w_n \\ p_n & \text{per } E \leq W - w_n \text{ e } p_n \geq 0 \\ 0 & \text{per } E \leq W - w_n \text{ e } p_n < 0 \end{cases}$$

end for
for all $j \in n-1, ..., 2$ **do**
  for all $E \in (0..W))$ **do**
    $F_j(E) = \max\{\mathrm{F}_{j+1}(E); p_j + \mathrm{F}_{j+1}(E + w_j)\}$
  end for
end for
$F_1(0) = \max\{\mathrm{F}_2(0); p_1 + \mathrm{F}_2(w_1)\}$

return $F_1(0)$

5.4.2 Esercizi risolti

ESEMPIO 5.13. Risolvere mediante programmazione dinamica il seguente problema dello zaino:

$$\max 3x_1 + 2x_2 + 2x_3 + 5x_4 + x_5$$

soggetto a

$$2x_1 + 4x_2 + 2x_3 + 7x_4 + 8x_5 \leq 12$$

$$x_i \in \{0,1\} \quad \forall i$$

Soluzione

$$F_5(E) = \begin{cases} 1 \textit{ per } \mathrm{E} \leq 4 \\ 0 \textit{ per } 5 \leq \mathrm{E} \leq 12 \end{cases}$$

$$F_4(E) = \max\{F_5(E), 5 + F_5(E+7)\} = \begin{cases} 5 \text{ per } E \leq 5 \\ 0 \text{ per } 6 \leq E \leq 12 \end{cases}$$

$$F_3(E) = \max\{F_4(E), 2 + F_4(E+2)\} = \begin{cases} 7 \text{ per } E \leq 3 \\ 5 \text{ per } 4 \leq E \leq 5 \\ 2 \text{ per } 6 \leq E \leq 10 \\ 0 \text{ per } 11 \leq E \leq 12 \end{cases}$$

$$F_2(E) = \max\{F_3(E), 2 + F_3(E+4)\} = \begin{cases} 7 \text{ per } 0 \leq E \leq 3 \\ 5 \text{ per } 4 \leq E \leq 5 \\ 4 \text{ per } E = 6 \\ 2 \text{ per } 7 \leq E \leq 10 \\ 0 \text{ per } 11 \leq E \leq 12 \end{cases}$$

$$F_1(E) = \max\{F_2(E), 3 + F_2(E+2)\} = \begin{cases} 10 \text{ per } 0 \leq E \leq 1 \\ 8 \text{ per } 2 \leq E \leq 3 \\ 7 \text{ per } E = 4 \\ 5 \text{ per } 5 \leq E \leq 8 \\ 3 \text{ per } 9 \leq E \leq 10 \\ 0 \text{ per } 11 \leq E \leq 12 \end{cases}$$

Riassumendo tutto sotto forma tabellare si ha:

	0	1	2	3	4	5	6	7	8	9	10	11	12
F_1	10	10	8	8	7	5	5	5	5	3	3	0	0
F_2	7	7	7	7	5	5	4	2	2	2	2	0	0
F_3	7	7	7	7	5	5	2	2	2	2	2	0	0
F_4	5	5	5	5	5	5	0	0	0	0	0	0	0
F_5	1	1	1	1	1	0	0	0	0	0	0	0	0

Si ottiene quindi come soluzione ottima $F^* = F_1(0) = 10$.
Procediamo adesso alla determinazione della soluzione ottima che porta a questo risultato della funzione obiettivo. Si parte dalla casella $x_1(0)$ e se ne legge il valore (in questo caso pari a 1). Il valore appena detto determina il fatto che l'oggetto in questione venga inserito oppure no nello zaino. Quindi se $x_1(0) = 1$ (come nel nostro caso) l'oggetto viene inserito dentro lo zaino per cui nella riga successiva, ovvero quella relativa a x_2, ci si sposterà in orizzontale di un numero di caselle pari al valore di w_1(proprio per questo nella seconda riga vado a considerare l'elemento $x_2(2)$ perchè il valore di w_1 è pari a 2). Se invece risultava $x_1(0) = 0$ nella riga successiva si andava a considerare la stessa colonna di x_1 in quanto l'oggetto 1 non veniva inserito nello zaino per cui la capacità residua dello stesso rimaneva costante. Considerazioni analoghe vanno ripetute per le successive variabili. Successivamente si ha $x_2(2) = 0$ per cui nella terza riga si considererà l'elemento $x_3(2)$. Per quest'ultimo risulterà $x_3(2) = 1$ per cui nella quarta riga si andrà a controllare il valore di $x_4(4)$ (infatti $w_3 = 2$). Per

questo elemento si avrà $x_4(4) = 1$ il che corrisponde ad analizzare nell'ultima riga il valore di $x_5(11)$ ($w_4 = 7$). Il valore di $x_5(11)$ infine, risulta essere pari a 0. Nelle tabelle sottostanti verranno indicate all'interno di un riquadro le caselle interessate alla costruzione della soluzione ottima. Nel caso specifico di questo problema si avrà:

	0	1	2	3	4	5	6	7	8	9	10	11	12
x_1	$\boxed{1}$	1	1	1	1	1	1	1	1	1	1	0	0
x_2	1	1	$\boxed{0}$	0	0	0	1	0	0	0	0	0	0
x_3	1	1	$\boxed{1}$	1	0	0	1	1	1	1	1	0	0
x_4	1	1	1	1	$\boxed{1}$	1	0	0	0	0	0	0	0
x_5	1	1	1	1	1	0	0	0	0	0	0	$\boxed{0}$	0

La soluzione ottima sarà dunque la seguente:

$$x_1^*=1,\ x_2^* = 0,\ x_3^* = 1,\ x_4^* = 1,\ x_5^* = 0.$$

ESEMPIO 5.14. Risolvere mediante programmazione dinamica il seguente problema dello zaino:

$$\max 9x_1 + 16x_2 + 18x_3 + 8x_4 + 7x_5$$

soggetto a

$$2x_1 + 4x_2 + 5x_3 + 7x_4 + 8x_5 \leq 15$$

$$x_i \in \{0, 1\}\ \forall i$$

Soluzione

$$F_5(E) = \begin{cases} 7 \text{ per } 0 \leq E \leq 7 \\ 0 \text{ per } 8 \leq E \leq 15 \end{cases}$$

$$F_4(E) = \max\{F_5(E), 8 + F_5(E+7)\} = \begin{cases} 15 \text{ per } E = 0 \\ 8 \text{ per } 1 \leq E \leq 8 \\ 0 \text{ per } 9 \leq E \leq 15 \end{cases}$$

$$F_3(E) = \max\{F_4(E), 18 + F_4(E+5)\} = \begin{cases} 26 \text{ per } 0 \leq E \leq 3 \\ 18 \text{ per } 4 \leq E \leq 10 \\ 0 \text{ per } 11 \leq E \leq 15 \end{cases}$$

$$F_2(E) = \max\{F_3(E), 16 + F_3(E+4)\} = \begin{cases} 34 \text{ per } 0 \leq E \leq 6 \\ 18 \text{ per } 7 \leq E \leq 10 \\ 16 \text{ per } E = 11 \\ 0 \text{ per } 12 \leq E \leq 15 \end{cases}$$

$$F_1(E) = \max\{F_2(E), 9 + F_2(E+2)\} = \begin{cases} 43 \text{ per } 0 \le E \le 4 \\ 34 \text{ per } 5 \le E \le 6 \\ 27 \text{ per } 7 \le E \le 8 \\ 25 \text{ per } E = 9 \\ 18 \text{ per } E = 10 \\ 16 \text{ per } E = 11 \\ 9 \text{ per } 12 \le E \le 13 \\ 0 \text{ per } 14 \le E \le 15 \end{cases}$$

Riassumendo tutto sotto forma tabellare si ha:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
F_1	43	43	43	43	43	34	34	27	27	25	18	16	9	9	0	0
F_2	34	34	34	34	34	34	34	18	18	18	18	16	0	0	0	0
F_3	26	26	26	26	18	18	18	18	18	18	18	0	0	0	0	0
F_4	8	8	8	8	8	8	8	8	8	0	0	0	0	0	0	0
F_5	7	7	7	7	7	7	7	7	0	0	0	0	0	0	0	0

Si ottiene quindi come soluzione ottima $F^* = F_1(0) = 43$

Per l'individuazione di x^*si riapplica quanto detto per il primo esercizio e adottando la stessa simbologia (ovvero si indicano all'interno di un riquadro le caselle interessate alla costruzione della soluzione) si ottiene:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1	$\boxed{1}$	1	1	1	1	0	0	1	1	1	0	0	1	1	0	0
x_2	1	1	$\boxed{1}$	1	1	1	1	0	0	0	0	1	0	0	0	0
x_3	1	1	1	1	1	1	$\boxed{1}$	1	1	1	1	0	0	0	0	0
x_4	1	1	1	1	1	1	1	1	1	0	0	$\boxed{0}$	0	0	0	0
x_5	1	1	1	1	1	1	1	1	0	0	0	$\boxed{0}$	0	0	0	0

La soluzione ottima sarà dunque la seguente:

$$x_1^*=1,\ x_2^* = 1,\ x_3^* = 1,\ x_4^* = 0,\ x_5^* = 0.$$

ESEMPIO 5.15. Risolvere mediante programmazione dinamica il seguente problema dello zaino:

$$\max 6x_1 + 4x_2 + 9x_3 + 5x_4 + x_5$$

soggetto a

$$2x_1 + 3x_2 + 10x_3 + 7x_4 + 2x_5 \leq 14$$
$$x_i \in \{0,1\} \quad \forall i$$

Soluzione

$$F_5(E) = \begin{cases} 1 \text{ per } 0 \leq E \leq 12 \\ 0 \text{ per } 13 \leq E \leq 14 \end{cases}$$

$$F_4(E) = \max\{F_5(E), 5 + F_5(E+7)\} = \begin{cases} 6 \text{ per } 0 \leq E \leq 5 \\ 5 \text{ per } 6 \leq E \leq 7 \\ 1 \text{ per } 8 \leq E \leq 12 \\ 0 \text{ per } 13 \leq E \leq 14 \end{cases}$$

$$F_3(E) = \max\{F_4(E), 18 + F_4(E+5)\} = \begin{cases} 10 \text{ per } 0 \leq E \leq 2 \\ 9 \text{ per } 3 \leq E \leq 4 \\ 6 \text{ per } E = 5 \\ 5 \text{ per } 6 \leq E \leq 7 \\ 1 \text{ per } 8 \leq E \leq 12 \\ 0 \text{ per } 13 \leq E \leq 14 \end{cases}$$

$$F_2(E) = \max\{F_3(E), 16 + F_3(E+4)\} = \begin{cases} 13 \text{ per } 0 \leq E \leq 1 \\ 10 \text{ per } E = 2 \\ 9 \text{ per } 3 \leq E \leq 4 \\ 6 \text{ per } E = 6 \\ 5 \text{ per } 6 \leq E \leq 9 \\ 4 \text{ per } 10 \leq E \leq 11 \\ 1 \text{ per } E = 12 \\ 0 \text{ per } 13 \leq E \leq 14 \end{cases}$$

$$F_1(E) = \max\{F_2(E), 9 + F_2(E+2)\} = \begin{cases} 16 \text{ per } E = 0 \\ 15 \text{ per } 1 \leq E \leq 2 \\ 12 \text{ per } E = 3 \\ 11 \text{ per } 4 \leq E \leq 7 \\ 10 \text{ per } 8 \leq E \leq 9 \\ 7 \text{ per } E = 10 \\ 6 \text{ per } 11 \leq E \leq 12 \\ 0 \text{ per } 13 \leq E \leq 14 \end{cases}$$

Riassumendo tutto sotto forma tabellare si ha:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
F_1	16	15	15	12	11	11	11	11	10	10	7	6	6	0	0
F_2	13	13	10	9	9	6	5	5	5	5	4	4	1	0	0
F_3	10	10	10	9	9	6	5	5	1	1	1	1	1	0	0
F_4	6	6	6	6	6	6	5	5	1	1	1	1	1	0	0
F_5	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0

Si ottiene quindi come soluzione ottima $F^* = F_1(0) = 16$
Per l'individuazione di x^*si ha invece:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
x_1	$\boxed{1}$	1	1	1	1	1	1	1	1	1	1	1	1	0	0
x_2	1	1	$\boxed{0}$	0	0	0	0	0	1	1	1	1	0	0	0
x_3	1	1	$\boxed{1}$	1	1	0	0	0	0	0	0	0	0	0	0
x_4	1	1	1	1	1	1	1	1	0	0	0	0	$\boxed{0}$	0	0
x_5	1	1	1	1	1	1	1	1	1	1	1	1	$\boxed{1}$	0	0

La soluzione ottima sarà dunque la seguente:

$$x_1^*=1,\ x_2^* = 0,\ x_3^* = 1,\ x_4^* = 0,\ x_5^* = 1.$$

ESEMPIO 5.16. Risolvere il seguente problema con il metodo della programmazione dinamica:

$$\max 5x_1 + 7x_2 + 9x_3 + 2x_4 + 12x_5$$

soggetto a

$$3x_1 + 2x_2 + 8x_3 + 7x_4 + 3x_5 \leq 12$$

$$x_i \in \{0, 1\}\ \forall i$$

Soluzione

$$F_5(E) = \begin{cases} 12 \text{ per } 0 \leq E \leq 9 \\ 0 \text{ per } 10 \leq E \leq 12 \end{cases}$$

$$F_4(E) = \max\{F_5(E), 2 + F_5(E+7)\} = \begin{cases} 14 \text{ per } 0 \leq E \leq 2 \\ 12 \text{ per } 3 \leq E \leq 9 \\ 0 \text{ per } 10 \leq E \leq 12 \end{cases}$$

$$F_3(E) = \max\{F_4(E), 18 + F_4(E+5)\} = \begin{cases} 21 \text{ per } 0 \leq E \leq 1 \\ 14 \text{ per } E = 2 \\ 12 \text{ per } 3 \leq E \leq 9 \\ 0 \text{ per } 10 \leq E \leq 12 \end{cases}$$

$$F_2(E) = \max\{F_3(E), 16 + F_3(E+4)\} = \begin{cases} 21 \text{ per } 0 \leq E \leq 1 \\ 19 \text{ per } 2 \leq E \leq 7 \\ 12 \text{ per } 8 \leq E \leq 9 \\ 7 \text{ per } E = 10 \\ 0 \text{ per } 11 \leq E \leq 12 \end{cases}$$

$$F_1(E) = \max\{F_2(E), 9 + F_2(E+2)\} = \begin{cases} 24 \text{ per } 0 \le E \le 4 \\ 19 \text{ per } 5 \le E \le 7 \\ 12 \text{ per } 8 \le E \le 9 \\ 7 \text{ per } E = 10 \\ 0 \text{ per } 11 \le E \le 12 \end{cases}$$

Riassumendo tutto sotto forma tabellare si ha:

	0	1	2	3	4	5	6	7	8	9	10	11	12
F_1	24	24	24	24	24	19	19	19	12	12	7	0	0
F_2	21	21	19	19	19	19	19	19	12	12	7	0	0
F_3	21	21	14	12	12	12	12	12	12	12	0	0	0
F_4	14	14	14	12	12	12	12	12	12	12	0	0	0
F_5	12	12	12	12	12	12	12	12	12	12	0	0	0

Si ottiene quindi come soluzione ottima $F^* = F_1(0) = 24$
Per l'individuazione di x^*si ha invece:

	0	1	2	3	4	5	6	7	8	9	10	11	12
x_1	$\boxed{1}$	1	1	1	1	0	0	0	0	0	0	0	0
x_2	1	1	1	$\boxed{1}$	1	1	1	1	0	0	1	0	0
x_3	1	1	0	0	0	$\boxed{0}$	0	0	0	0	0	0	0
x_4	1	1	1	0	0	$\boxed{0}$	0	0	0	0	0	0	0
x_5	1	1	1	1	1	$\boxed{1}$	1	1	1	1	0	0	0

La soluzione ottima sarà dunque la seguente:

$$x_1^*{=}1,\ x_2^* = 1,\ x_3^* = 0,\ x_4^* = 0,\ x_5^* = 1.$$

ESEMPIO 5.17. Risolvere il seguente problema con il metodo della programmazione dinamica:

$$\max 5x_1 + 7x_2 + 9x_3 + 2x_4 + 12x_5$$

soggetto a

$$3x_1 + 2x_2 + 8x_3 + 7x_4 + 3x_5 \le 7$$
$$x_i \in \{0,1\}\ \forall i$$

Soluzione
Si nota che il problema in questione è lo stesso dell'esercizio precedente con capacità dello zaino diminuita di cinque. La soluzione ottima del problema è dunque leggibile nella tabella del problema precedente nella casella $F_1(5)$, ovvero $F_1(5) = 19$ Per l'individuazione di x^* si procede come negli esercizi precedenti, partendo dalla casella $F_1(5)$ invece che da quella $F_1(0)$. Si ha dunque:

	0	1	2	3	4	5	6	7	8	9	10	11	12
x_1	1	1	1	1	1	$\boxed{0}$	0	0	0	0	0	0	0
x_2	1	1	1	1	1	$\boxed{1}$	1	1	0	0	1	0	0
x_3	1	1	0	0	0	0	0	$\boxed{0}$	0	0	0	0	0
x_4	1	1	1	0	0	0	0	$\boxed{0}$	0	0	0	0	0
x_5	1	1	1	1	1	1	1	$\boxed{1}$	1	1	0	0	0

La soluzione ottima sarà dunque la seguente:

$$x_1^*=0,\ x_2^* = 1,\ x_3^* = 0,\ x_4^* = 0,\ x_5^* = 1.$$

5.5 Esercizi proposti

1. Si risolva il seguente problema dello zaino 0/1 mediante i metodi del Branch & Bound e della programmazione dinamica.

$$\begin{array}{ll} \max & 6x_1 + 5x_2 + x_3 + 3x_4 + 5x_5 \\ & 4x_1 + 2x_2 + 2x_3 + 5x_4 + 5x_5 \leq 9 \\ & x_i \in \{0,1\}, \forall i \end{array}$$

2. Si risolva il seguente problema dello zaino 0/1 mediante i metodi del Branch & Bound e della programmazione dinamica.

$$\begin{array}{ll} \max & 2x_1 + 5x_2 + 4x_3 + 1x_4 + 6x_5 \\ & 3x_1 + 4x_2 + 5x_3 + 5x_4 + 2x_5 \leq 9 \\ & x_i \in \{0,1\}, \forall i \end{array}$$

3. Si risolva il seguente problema dello zaino 0/1 mediante i metodi del Branch & Bound e della programmazione dinamica.

$$\begin{array}{ll} \max & 6x_1 + 3x_2 + 5x_3 + 10x_4 + 6x_5 \\ & 5x_1 + 4x_2 + 12x_3 + 10x_4 + 7x_5 \leq 13 \\ & x_i \in \{0,1\}, \forall i \end{array}$$

4. Si risolva il seguente problema dello zaino 0/1 mediante i metodi del Branch & Bound e della programmazione dinamica.

$$\begin{array}{ll} \max & 7x_1 + 3x_2 + 5x_3 + 9x_4 + 6x_5 \\ & 5x_1 + 4x_2 + 12x_3 + 10x_4 + 7x_5 \leq 11 \\ & x_i \in \{0,1\}, \forall i \end{array}$$

Finito di stampare
nel Settembre 2010 dalla
Global Print - Gorgonzola (MI)