

9

Modelli di Programmazione Lineare Intera

Come è stato già osservato in precedenza, quando tutte le variabili di un problema di Programmazione Lineare sono vincolate ad assumere valori interi, si parla di Programmazione Lineare Intera. Moltissimi problemi reali possono essere rappresentati da modelli di Programmazione Lineare Intera; tipicamente si tratta di problemi in cui le variabili di decisione rappresentano quantità indivisibili (come il numero di automobili, di persone addette a certe mansioni, etc.) oppure sono problemi caratterizzati dalla necessità di scegliere tra un numero finito di alternative diverse. In quest'ultimo caso, in particolare, si avranno problemi di Programmazione Lineare 0–1, cioè problemi in cui le variabili sono binarie e assumono valore 0 oppure 1.

9.1 VARIABILI INTERE PER RAPPRESENTARE QUANTITÀ INDIVISIBILI

Un numero molto elevato di problemi reali è caratterizzato dalla indivisibilità del bene da produrre o della risorsa da utilizzare. Di qui la necessità di rappresentare tali problemi attraverso modelli di Programmazione Lineare con variabili intere. Questo tipo di problemi riguardano molte applicazioni reali: dai problemi in ambito industriale come la distribuzione dei beni e il sequenziamento delle attività produttive, ai problemi economici come la gestione ottima di un portafoglio titoli; dai problemi di progettazione ottima ai problemi inerenti la biologia e la fisica delle alte energie.

Esempi di modelli di Programmazione Lineare Intera caratterizzati da variabili di decisione associate a quantità indivisibili sono già stati presi in esame all'interno della trattazione dei modelli di Programmazione Lineare. Una situazione tipica è

data dall'Esempio 3.4.2 in cui il bene da produrre è rappresentato da autovetture che sono ovviamente indivisibili; quindi la formulazione di Programmazione Lineare già fornita per questo esempio in realtà, per essere aderente alla situazione reale, deve essere integrata con la presenza del vincolo di interezza sulle variabili che rappresentano i livelli di produzioni delle autovetture. Analogamente l'introduzione del vincolo di interezza sulle variabili è indispensabile quando viene meno una delle ipotesi fondamentali della Programmazione Lineare cioè la continuità delle variabili; in questo caso i modelli di Programmazione Lineare Intera sono uno strumento essenziale per rappresentare situazioni del mondo reale di questo tipo.

9.2 VARIABILI BINARIE PER RAPPRESENTARE SCELTE ALTERNATIVE

Si supponga di dover modellare il fatto che un certo evento possa verificarsi oppure no. La natura binaria del problema suggerisce l'idea di modellare questa dicotomia per mezzo di un variabile binaria $\delta \in \{0, 1\}$; si porrà $\delta = 1$ se l'evento si verifica e $\delta = 0$ altrimenti.

9.2.1 Problemi di assegnamento

Un generico problema di assegnamento consiste nel determinare il modo ottimale di assegnare lavori a persone o, più in generale, di assegnare *mezzi* (persone, macchine, etc.) ad *attività*.

Supponiamo che n persone $\mathbf{P}_1, \dots, \mathbf{P}_n$, debbano svolgere n lavori $\mathbf{L}_1, \dots, \mathbf{L}_n$; ciascun lavoro deve essere svolto esattamente da una persona e ciascuna persona deve svolgere esattamente un lavoro. Naturalmente le persone hanno diversi livelli di esperienza, competenza ed abilità e quindi si può introdurre un costo dell'assegnamento della persona i al lavoro j ; indichiamo tale costo con c_{ij} e supponiamo che sia noto. Questo costo può, ad esempio, essere interpretato come tempo medio impiegato dalla persona i ad eseguire il lavoro j .

Il problema consiste, quindi, nell'assegnare i lavori alle persone minimizzando il costo totale di realizzazione di tutti i lavori.

Questo tipo di problemi sorge in molte situazioni pratiche: esempi tipici sono i problemi di assegnamento del personale all'interno di una azienda e i problemi di assegnare determinati mezzi di trasporto ad alcune particolari linee. Un esempio di problema di assegnamento è stato già considerato nell'Introduzione (pagina 6) quando si è brevemente analizzato il caso dell'assegnamento di 70 dipendenti a 70 mansioni diverse.

Esaminiamo, ora, alcuni esempi.

Esempi

Esempio 9.2.1 Una compagnia finanziaria necessita di ricoprire tre lavori **LAV1**, **LAV2**, **LAV3**, che richiedono differenti abilità ed esperienza. Sono disponibili tre candidati **C1**, **C2**, **C3**, che possono essere assunti con il medesimo salario. A causa delle loro differenti capacità, il costo di assegnazione di ciascun candidato che la compagnia deve sostenere dipende dal tipo di lavoro al quale è assegnato. La stima di tale costo riferito a ciascun candidato se fosse assegnato a ciascuno dei tre lavori è riportato nella tabella seguente

	LAV1	LAV2	LAV3
C1	5	4	7
C2	6	7	3
C3	8	11	2

Si desidera assegnare ogni candidato esattamente ad un lavoro in modo da minimizzare il costo complessivo che la compagnia deve sostenere.

Formulazione.

L'esempio in esame è di piccole dimensioni: infatti ci sono solamente $3! = 6$ possibili assegnazioni.

– *Variabili.* Per ogni lavoro e per ogni persona, introduciamo le variabili binarie

$$x_{ij} = \begin{cases} 1 & \text{se il candidato } i \text{ è assegnato al lavoro } j \\ 0 & \text{altrimenti.} \end{cases}$$

– *Funzione obiettivo.* La funzione obiettivo da minimizzare sarà

$$5x_{11} + 4x_{12} + 7x_{13} + 6x_{21} + 7x_{22} + 3x_{23} + 8x_{31} + 11x_{32} + 2x_{33}.$$

– *Vincoli.* Come già osservato nel caso generale, si devono considerare i seguenti vincoli

$$\sum_{i=1}^3 x_{ij} = 1 \quad j = 1, \dots, 3.$$

$$\sum_{j=1}^3 x_{ij} = 1 \quad i = 1, \dots, 3.$$

La formulazione completa si può scrivere

$$\begin{cases} \min(5x_{11} + 4x_{12} + 7x_{13} + 6x_{21} + 7x_{22} + 3x_{23} + \\ \quad + 8x_{31} + 11x_{32} + 2x_{33}) \\ x_{11} + x_{21} + x_{31} = 1 \\ x_{12} + x_{22} + x_{32} = 1 \\ x_{13} + x_{23} + x_{33} = 1 \\ x_{11} + x_{12} + x_{13} = 1 \\ x_{21} + x_{22} + x_{23} = 1 \\ x_{31} + x_{32} + x_{33} = 1 \\ x_{ij} \in \{0, 1\} \quad i, j = 1, \dots, 3. \end{cases}$$

Formulazione generale di un problema di assegnamento

Esaminiamo, ora, una formulazione in termini di programmazione lineare per un generico problema di assegnamento. Supponiamo che n persone $\mathbf{P}_1, \dots, \mathbf{P}_n$, debbano svolgere n lavori $\mathbf{L}_1, \dots, \mathbf{L}_n$ e che ciascun lavoro deve essere svolto esattamente da una persona e ciascuna persona deve svolgere esattamente un lavoro. Sia c_{ij} il costo dell'assegnamento della persona i al lavoro j ; si devono assegnare i lavori alle persone minimizzando il costo totale di realizzazione di tutti i lavori.

Formulazione.

– *Variabili.* Per ogni lavoro i e per ogni persona j , ($i, j = 1, \dots, n$) introduciamo le seguenti variabili binarie

$$x_{ij} = \begin{cases} 1 & \text{se la persona } i \text{ è assegnata al lavoro } j \\ 0 & \text{altrimenti.} \end{cases}$$

Si tratta di n^2 variabili:

	\mathbf{L}_1	\dots	\mathbf{L}_j	\dots	\mathbf{L}_n
\mathbf{P}_1	x_{11}	\dots	x_{1j}	\dots	x_{1n}
\vdots	\vdots		\vdots		\vdots
\mathbf{P}_i	x_{i1}	\dots	x_{ij}	\dots	x_{in}
\vdots	\vdots		\vdots		\vdots
\mathbf{P}_n	x_{n1}	\dots	x_{nj}	\dots	x_{nn}

– *Funzione obiettivo.* La funzione obiettivo da minimizzare sarà data dal costo totale cioè da

$$\sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}.$$

Naturalmente, se le c_{ij} anziché essere dei costi fossero i valori utili ricavati dall'assegnamento della persona i al lavoro j , allora la funzione obiettivo andrebbe massimizzata.

– *Vincoli.* (Vincoli di assegnamento.) Poiché esattamente una persona deve essere assegnata al lavoro j , allora si avranno i seguenti n vincoli

$$\sum_{i=1}^n x_{ij} = 1 \quad j = 1, \dots, n.$$

Inoltre, poiché ciascuna persona deve essere assegnata ad una sola attività, si avranno altri n vincoli

$$\sum_{j=1}^n x_{ij} = 1 \quad i = 1, \dots, n.$$

La formulazione completa sarà, quindi, data da

$$\left\{ \begin{array}{l} \min \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \\ \sum_{i=1}^n x_{ij} = 1 \\ \sum_{j=1}^n x_{ij} = 1 \\ x_{ij} \in \{0, 1\}, \quad i, j = 1, \dots, n. \end{array} \right.$$

Osservazione 9.2.2 Osservando la formulazione ottenuta si può facilmente dedurre che la struttura di un problema di assegnamento è del tutto simile a quella del problema dei trasporti (cfr. Paragrafo 3.4.3) in cui si ha lo stesso numero di origini e di destinazioni. La differenza sostanziale sta nel fatto che in un problema di assegnamento le variabili sono binarie, $x_{ij} \in \{0, 1\}$ mentre in un problema dei trasporti le variabili sono reali non negative $x_{ij} \geq 0$. D'altra parte, per il Teorema 3.4.2, se in un problema dei trasporti i termini noti dei vincoli sono interi, se esiste soluzione ottima allora esiste soluzione ottima intera del problema dei trasporti. Quindi, poiché in un problema di assegnamento tali termini noti sono pari ad 1, i vincoli $x_{ij} \in \{0, 1\}$, $i, j = 1, \dots, n$ possono essere riscritti nella forma $0 \leq x_{ij} \leq 1$, $i, j = 1, \dots, n$. Inoltre, poiché i vincoli $x_{ij} \leq 1$ sono implicati dai vincoli di assegnamento, si possono scrivere semplicemente i vincoli $x_{ij} \geq 0$, $i, j = 1, \dots, n$ e comunque avere la garanzia che se esiste una soluzione ottima allora esiste una soluzione ottima intera 0–1. Quindi un problema di assegnamento

può essere considerato equivalente al problema

$$\left\{ \begin{array}{l} \min \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \\ \sum_{i=1}^n x_{ij} = 1 \\ \sum_{j=1}^n x_{ij} = 1 \\ x_{ij} \geq 0, \quad i, j = 1, \dots, n \end{array} \right.$$

cioè può essere riscritto come un particolare problema di Programmazione Lineare avente la struttura medesima del problema dei trasporti.

È possibile effettuare una generalizzazione del problema dell'assegnamento per categorie di lavori. Infatti, frequentemente ci sono molti lavori identici che richiedono la stessa qualifica; tali lavori possono essere raggruppati in categorie di attività. Assumiamo quindi che esistano n categorie di attività e denotiamo con b_j il numero di lavori raggruppati nella j -esima categoria. Anche le persone possono essere raggruppate in categorie di persone aventi lo stesso valore; assumiamo che esistano m di queste categorie di persone e sia a_i il numero di persone poste nella i -esima categoria. Denotiamo con c_{ij} il valore utile ottenuto assegnando una persona della categoria i ad un lavoro della categoria j . Assumiamo che $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$. Introducendo le variabili x_{ij} rappresentanti il numero di persone della stessa categoria i assegnate ad un lavoro della categoria j , questo generale problema di assegnamento può essere formulato in termini di un problema di programmazione lineare nel seguente modo:

$$\left\{ \begin{array}{l} \min \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \\ \sum_{j=1}^n x_{ij} = a_i \quad i = 1, \dots, m \\ \sum_{i=1}^m x_{ij} = b_j \quad j = 1, \dots, n \\ x_{ij} \geq 0 \quad i = 1, \dots, m \quad j = 1, \dots, n. \end{array} \right.$$

Esempio 9.2.3 Una compagnia aerea cerca di pianificare le proprie linee aeree creando un aeroporto centrale e cercando di avere un elevato numero di voli in arrivo in questo aeroporto in una certa fascia oraria ed un elevato numero di partenze nella fascia oraria immediatamente successiva. Questo permette ai passeggeri di avere un elevato numero di combinazioni tra città di partenza e città di destinazione con una sola coincidenza e al più un cambio di aereo nell'aeroporto centrale. Il fine è quello di creare una tale struttura in modo da minimizzare i cambi di aerei e quindi il movimento di bagagli nell'aeroporto centrale. Supponiamo che la compagnia aerea abbia cinque voli che arrivano tra le 8 e le 8.30 nell'aeroporto centrale e che poi gli stessi aerei partono per altre diverse destinazioni tra le 8.40 e le 9.20. La tabella che segue riporta il numero medio di passeggeri che arrivano con uno dei voli in arrivo **A1**, **A2**, **A3**, **A4**, **A5** e che ripartono con i voli in partenza **P1**, **P2**, **P3**, **P4**, **P5**, ovvero passeggeri che non cambiano aereo

	P1	P2	P3	P4	P5
A1	15	20	8	16	12
A2	17	9	15	25	12
A3	12	32	16	9	20
A4	-	15	9	7	30
A5	-	-	35	10	18

Il volo **A4** arriva troppo tardi e non permette di prendere il volo in partenza **P1**; analogamente il volo **A5** non permette coincidenze con i voli in partenza **P1** e **P2**. Supponendo che tutti gli aerei sono identici, il problema consiste nell'assegnare ciascun aereo in arrivo ad uno dei voli in partenza in modo da massimizzare il numero delle persone che non devono cambiare aereo.

Formulazione.

Il problema in analisi può essere formulato come problema di assegnamento.

– *Variabili.* Introduciamo le variabili di decisione x_{ij} definite come segue

$$x_{ij} = \begin{cases} 1 & \text{se l'aereo del volo } \mathbf{Ai} \text{ è assegnato al volo } \mathbf{Pj} \\ 0 & \text{altrimenti.} \end{cases}$$

– *Funzione obiettivo.* Definiamo come funzione obiettivo il numero di passeggeri che non devono cambiare aereo:

$$\begin{aligned} & 15x_{11} + 20x_{12} + 8x_{13} + 16x_{14} + 12x_{15} + \\ & + 17x_{21} + 9x_{22} + 15x_{23} + 25x_{24} + 12x_{25} + \\ & + 12x_{31} + 32x_{32} + 16x_{33} + 9x_{34} + 20x_{35} + \\ & + 15x_{42} + 9x_{43} + 7x_{44} + 30x_{45} + \\ & + 35x_{53} + 10x_{54} + 18x_{55}. \end{aligned}$$

Tale funzione deve naturalmente essere massimizzata.

– *Vincoli.* I vincoli saranno

$$\begin{aligned}
 x_{11} + x_{12} + x_{13} + x_{14} + x_{15} &= 1 \\
 x_{21} + x_{22} + x_{23} + x_{24} + x_{25} &= 1 \\
 x_{31} + x_{32} + x_{33} + x_{34} + x_{35} &= 1 \\
 x_{42} + x_{43} + x_{44} + x_{45} &= 1 \\
 x_{53} + x_{54} + x_{55} &= 1 \\
 x_{11} + x_{21} + x_{31} &= 1 \\
 x_{12} + x_{22} + x_{32} + x_{42} &= 1 \\
 x_{13} + x_{23} + x_{33} + x_{43} + x_{53} &= 1 \\
 x_{14} + x_{24} + x_{34} + x_{44} + x_{54} &= 1 \\
 x_{15} + x_{25} + x_{35} + x_{45} + x_{55} &= 1.
 \end{aligned}$$

Quindi la formulazione completa sarà

$$\left\{ \begin{array}{l}
 \max \left(15x_{21} + 20x_{22} + 8x_{23} + 16x_{24} + 12x_{25} + \right. \\
 \quad + 17x_{11} + 9x_{12} + 15x_{13} + 25x_{14} + 12x_{15} + \\
 \quad + 12x_{31} + 32x_{32} + 16x_{33} + 9x_{34} + 20x_{35} + \\
 \quad + 15x_{42} + 9x_{43} + 7x_{44} + 30x_{45} + \\
 \quad \left. + 35x_{53} + 10x_{54} + 18x_{55} \right) \\
 x_{11} + x_{12} + x_{13} + x_{14} + x_{15} = 1 \\
 x_{21} + x_{22} + x_{23} + x_{24} + x_{25} = 1 \\
 x_{31} + x_{32} + x_{33} + x_{34} + x_{35} = 1 \\
 x_{42} + x_{43} + x_{44} + x_{45} = 1 \\
 x_{53} + x_{54} + x_{55} = 1 \\
 x_{11} + x_{21} + x_{31} = 1 \\
 x_{12} + x_{22} + x_{32} + x_{42} = 1 \\
 x_{13} + x_{23} + x_{33} + x_{43} + x_{53} = 1 \\
 x_{14} + x_{24} + x_{34} + x_{44} + x_{54} = 1 \\
 x_{15} + x_{25} + x_{35} + x_{45} + x_{55} = 1 \\
 x_{ij} \geq 0 \quad i, j = 1, \dots, 5.
 \end{array} \right.$$

Riportiamo di seguito il file `aeroporto.mod` e `aeroporto.dat` che realizzano un'implementazione in AMPL del modello ora formulato.

```

aeroporto.mod
set ARRIVI;
set PARTENZE;

param passeggeri{ARRIVI,PARTENZE}>=0;

```

```

var x{ARRIVI,PARTENZE} binary;
#var x{ARRIVI,PARTENZE} >=0, <= 1;

maximize passeggeri_che_non_cambiano_aereo :
    sum{i in ARRIVI, j in PARTENZE} passeggeri[i,j]*x[i,j];

s.t. vincoliA{i in ARRIVI} : sum{j in PARTENZE} x[i,j] = 1;
s.t. vincoliP{j in PARTENZE} : sum{i in ARRIVI} x[i,j] = 1;

```

aeroporto.dat

```

set ARRIVI := A1, A2, A3, A4, A5;
set PARTENZE := P1, P2, P3, P4, P5;

param passeggeri : P1    P2    P3    P4    P5 :=
      A1  15    20    8    16    12
      A2  17     9    15    25    12
      A3  12    32    16     9    20
      A4   0    15     9     7    30
      A5   0     0    35    10    18;

# fissare le 3 variabili a zero

fix x["A4","P1"]:=0;
fix x["A5","P1"]:=0;
fix x["A5","P2"]:=0;

```

Una soluzione ottima di questo problema è la seguente:

$$x_{11} = 1, x_{24} = 1, x_{32} = 1, x_{45} = 1, x_{53} = 1$$

e le altre variabili nulle; il valore ottimo della funzione obiettivo è 137.

Osservando questa soluzione si può notare come ciascun aereo in arrivo sia stato assegnato al volo in partenza che permette di mantenere sullo stesso aereo il maggior numero di passeggeri eccetto che per il volo in arrivo **A1**: infatti l'aereo in arrivo con il volo **A1** è stato assegnato al volo in partenza **P1** e quindi il numero dei passeggeri che non devono cambiare aereo è 15 contro ad esempio un numero di 20 o 16 passeggeri che sarebbero rimasti sullo stesso aereo se questo fosse stato assegnato rispettivamente al volo in partenza **P2** o **P4**.

9.2.2 Problemi di Knapsack binario

Il problema del “knapsack”, nella sua versione originaria, può essere descritto come segue: dato un insieme di n oggetti di dimensioni diverse e differenti valori, si vuole determinare un sottoinsieme di questi oggetti da inserire in una “bisaccia” (knapsack) di capacità limitata in modo da massimizzare il valore trasportato. In questo caso l’evento da modellare è l’inserimento dell’oggetto nella “bisaccia”; è quindi intuitivo introdurre una variabile binaria $x_i \in \{0, 1\}$, $i = 1, \dots, n$ che assuma valore 1 se l’ i -esimo oggetto è inserito nella “bisaccia”, 0 se invece non è inserito.

Più in generale, supponiamo di avere n progetti e un budget disponibile per la loro realizzazione. Il problema consiste nello scegliere un sottoinsieme dei progetti in modo da massimizzare la somma dei valori senza superare il limite imposto dal budget nell’ipotesi che ciascun progetto scelto deve essere realizzato completamente e non è accettata una realizzazione parziale del progetto.

Esempio 9.2.4 *Si supponga di disporre di un capitale di 18 mila euro e di poterle investire in 4 progetti diversi. Nel primo progetto si debbono investire 8 euro per ricavarne 40, nel secondo si debbono investire 6 euro per ricavarne 24, nel terzo progetto si debbono investire 5 euro per ricavarne 15, infine nel quarto progetto si debbono investire 4 euro per ricavarne 8. Formulare il problema di PLI che consente di scegliere l’insieme di progetti che massimizza il profitto rispettando i vincoli di disponibilità di capitale.*

Formulazione.

– *Variabili.* Le variabili di decisione sono definite, per $i = 1, 2, 3, 4$ come segue

$$x_i = \begin{cases} 1 & \text{se si sceglie il progetto } i \\ 0 & \text{altrimenti.} \end{cases}$$

– *Funzione obiettivo.* La funzione obiettivo da massimizzare è

$$40x_1 + 24x_2 + 15x_3 + 8x_4.$$

– *Vincoli.* I vincoli esprimono il fatto che il costo degli investimenti non può superare il budget disponibile, cioè

$$8x_1 + 6x_2 + 5x_3 + 4x_4 \leq 18.$$

Complessivamente il problema si scrive:

$$\begin{cases} \max 40x_1 + 24x_2 + 15x_3 + 8x_4 \\ 8x_1 + 6x_2 + 5x_3 + 4x_4 \leq 18 \\ x_i \in \{0, 1\} \quad i = 1, \dots, 4 \end{cases}$$

□

Formulazione generale di un problema di knapsack binario.

In generale, supponiamo di avere n progetti tali che l' i -esimo progetto ha costo di realizzazione a_i ed un valore pari c_i , $i = 1, \dots, n$; supponiamo inoltre che esista un budget b disponibile per la realizzazione dei progetti. Il problema consiste nello scegliere un sottoinsieme dei progetti in modo da massimizzare la somma dei valori senza superare il limite imposto dal budget.

L'evento da modellare, in questo caso, è la realizzazione del singolo progetto. Ciò può essere effettuato introducendo n variabili binarie nel seguente modo.

– *Variabili.* Introduciamo le variabili $x_i \in \{0, 1\}$, $i = 1, \dots, n$, tali che

$$x_i = \begin{cases} 1 & \text{se l}'i\text{-esimo progetto è realizzato} \\ 0 & \text{se l}'i\text{-esimo progetto non è realizzato.} \end{cases}$$

– *Funzione obiettivo.* È data dal valore complessivo cioè da

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

– *Vincoli.* Non si deve superare il budget disponibile e quindi si deve imporre

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

La formulazione complessiva può essere quindi scritta

$$\begin{cases} \max \sum_{i=1}^n c_i x_i \\ \sum_{i=1}^n a_i x_i \leq b \\ x \in \{0, 1\}^n. \end{cases}$$

Tuttavia, in generale, questi problemi di scelta tra sottoprogetti possono avere più vincoli: si pensi al caso in cui il budget sia mensile e si voglia pianificare per più di un mese; in questo caso il problema è detto *knapsack multidimensionale*.

9.2.3 Problemi di “Capital Budgeting” (pianificazione degli investimenti)

I problemi di pianificazione degli investimenti rappresentano una delle problematiche di maggiore importanza all'interno delle problematiche finanziarie. Anche all'interno delle realtà aziendali, la politica degli investimenti è strettamente legata alla pianificazione finanziaria e ai processi di spesa. Di solito gli investimenti sono valutati attraverso il cosiddetto *indice di redditività* di ciascun investimento

e una strategia di scelta degli investimenti dettata dal buon senso e quella di ordinare gli investimenti in base a tali indici e scegliendo gli investimenti nell'ordine stabilito cercano di non violare il vincolo sul budget disponibile ed eventuali altri vincoli. Ovviamente una formulazione di un modello di Programmazione Lineare 0–1 che rappresenti il problema permette invece di ottenere una soluzione ottima del problema

Il modello di Programmazione Lineare Intera che descrive il problema della pianificazione degli investimenti viene denominato *modello di "Capital Budgeting"* ed è stato proposto alla fine degli anni '50 dagli economisti Manne e Markowitz; quest'ultimo fu poi insignito del premio Nobel per l'Economia.

In sintesi il problema della pianificazione degli investimenti può essere così descritto: siano dati n progetti di investimento da realizzare o meno. Si fissa un orizzonte temporale T entro il quale si vuole effettuare l'analisi (ad esempio, $T = 1$ anno). T si suddivide in t periodi $T = \{1, \dots, t\}$ (ad esempio, un anno può essere diviso in quattro trimestri, ovvero $t = 4$). Ciascun progetto di investimento i -esimo, $i = 1, \dots, n$ è caratterizzato da un vettore $a_i = (a_{i1}, \dots, a_{it})^T$ del *flusso di cassa*, ovvero a_{ij} rappresenta il flusso di cassa (positivo o negativo) generato dall' i -esimo progetto nel periodo j -esimo. Si assume che il flusso di cassa positivo corrisponda ad una spesa, mentre un flusso di cassa negativo corrisponda ad un guadagno. Quindi se, ad esempio, il flusso di cassa relativo ad un certo progetto su un orizzonte temporale di quattro periodi è pari a $(4, 3, -2, -7)$, allora la realizzazione del progetto richiede spese di 4 e 3 nei primi due periodi e poi fornisce un guadagno di 2 e 7 rispettivamente nel terzo e quarto periodo. Fra l'altro, questa struttura è tipica dei flussi di cassa. La spesa totale sarebbe $4 + 3 - 2 - 7 = -2$, ovvero un guadagno complessivo di 2. Ora, per ogni $i = 1, \dots, n$ denotiamo con

$$c_i = - \sum_{j=1}^t a_{ij}$$

l'indice di redditività del progetto i -esimo, dove il segno meno è necessario per far corrispondere c_i al valore del guadagno. Inoltre, per ogni periodo $j = 1, \dots, t$ c'è un budget limitato denotato con b_j . Il problema consiste nel determinare un sottoinsieme di progetti da realizzare in modo da avere guadagno massimo. Si assume inoltre che i progetti non sono frazionabili, cioè non possono essere realizzati parzialmente.

Formulazione.

– *Variabili.* Introduciamo le variabili $x_i \in \{0, 1\}$ $j = 1, \dots, n$, così definite:

$$x_i = \begin{cases} 1 & \text{se l}'i\text{-esimo progetto è realizzato} \\ 0 & \text{se il } i\text{-esimo progetto non è realizzato.} \end{cases}$$

– *Funzione obiettivo.* È data dal valore complessivo cioè da

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

– *Vincoli.* Per ogni periodo $j \in T$ non si deve superare il budget b_j disponibile e quindi per ogni $j = 1, \dots, t$ si deve imporre

$$\sum_{i=1}^n a_{ij} x_i \leq b_j.$$

Si ha quindi un vincolo di “knapsack” per ogni periodo $j = 1, \dots, t$.

La formulazione complessiva può essere quindi scritta

$$\begin{cases} \max \sum_{i=1}^n c_i x_i \\ \sum_{i=1}^n a_{ij} x_i \leq b_j & j = 1, \dots, t \\ x \in \{0, 1\}^n. \end{cases}$$

Osservazione 9.2.5 Se fossero possibili realizzazioni parziali di alcuni progetti, le variabili corrispondenti non sarebbero vincolate ad appartenere all’insieme $\{0, 1\}$, ma sarebbero appartenenti all’intervallo $[0, 1]$ e rappresenterebbero il livello di realizzazione del progetto; in questo caso si avrebbe un problema di Programmazione Lineare Mista

Osservazione 9.2.6 Altri vincoli che realizzano delle condizioni logiche sono facilmente introducibili nella formulazione. Ad esempio:

- la realizzazione di un particolare progetto (ad esempio il 5°)

$$x_5 = 1;$$

- la realizzazione di *esattamente* uno dei progetti 1°, 3° e 11°:

$$x_1 + x_3 + x_{11} = 1;$$

- la realizzazione di *almeno* due dei progetti 1°, 3° e 11°:

$$x_1 + x_3 + x_{11} \geq 2;$$

- la realizzazione di *al più* due dei progetti 1°, 3° e 11°:

$$x_1 + x_3 + x_{11} \leq 2.$$

9.3 VARIABILI BINARIE COME VARIABILI INDICATRICI

Un altro classico uso di variabili 0 – 1, consiste nell'indicare le relazioni di dipendenza tra alcune grandezze di un problema; cioè, in questo caso, le variabili binarie vengono utilizzate come *variabili indicatrici*.

Supponiamo che la variabile $x_i \geq 0$ rappresenti una grandezza del problema e di conoscere un limite superiore di tale variabile, cioè un valore M positivo maggiore del più grande valore che può assumere la x_i . Allora, può essere necessario imporre la condizione:

$$x_i > 0 \Rightarrow \delta = 1 \quad (9.3.1)$$

oppure la condizione equivalente $\delta = 0 \Rightarrow x_i = 0$ (si ricordi che era assunto che $x_i \geq 0$). L'implicazione (9.3.1) può essere modellata con il vincolo

$$x_i - M\delta \leq 0.$$

Tuttavia, in altri casi, può essere necessario imporre la condizione

$$\delta = 1 \Rightarrow x_i > 0 \quad (9.3.2)$$

(che è equivalente a $x_i = 0 \Rightarrow \delta = 0$, poiché, per ipotesi, $x_i \geq 0$). La condizione logica (9.3.2) non si può facilmente rappresentare con un vincolo. Supponiamo, ad esempio, che in un problema di miscelazione una variabile x_i rappresenti la quantità di un ingrediente da includere nella miscela e quindi si ha $x_i \geq 0$; si può usare una variabile indicatrice $\delta \in \{0, 1\}$ per distinguere tra il caso in cui $x_i = 0$ e $x_i > 0$. La condizione logica (9.3.2) afferma che se $\delta = 1$ allora l'ingrediente rappresentato da x deve apparire nella miscela, ma non fornisce nessuna indicazione sulla quantità dell'ingrediente. In realtà, è più verosimile imporre una condizione logica del tipo

$$\delta = 1 \Rightarrow x_i \geq \varepsilon > 0 \quad (9.3.3)$$

cioè se $\delta = 1$ allora la variabile x_i assume un valore almeno pari ad ε .

La (9.3.3) è rappresentabile dal vincolo

$$x_i - \varepsilon\delta \geq 0. \quad (9.3.4)$$

Riepilogando possiamo considerare il seguente schema: se x_i è una variabile non negativa e $\delta \in \{0, 1\}$ ed inoltre $x_i < M$ e $\varepsilon > 0$, allora

$$\begin{aligned} x_i - M\delta \leq 0 & \Leftrightarrow \begin{cases} x_i > 0 \Rightarrow \delta = 1 \\ \delta = 0 \Rightarrow x_i = 0 \end{cases} \\ x_i - \varepsilon\delta \geq 0 & \Leftrightarrow \begin{cases} \delta = 1 \Rightarrow x_i \geq \varepsilon \\ x_i = 0 \Rightarrow \delta = 0. \end{cases} \end{aligned}$$

Analizziamo, ora, un esempio di miscelazione in cui applichiamo quanto appena esposto.

Esempio 9.3.1 *Sia data la seguente tavola di valori nutrizionali che riporta il tipo di alimento, il costo unitario, le unità di sostanze (proteine, carboidrati, grassi, vitamine, calcio) per unità di alimento*

	costo	prot.	carb.	grassi	vitam.	calcio
1	0.15	0	7	1	1	0
2	0.23	1	0	3	1	4
3	0.79	5	0	4	0	1
4	0.47	2	2	1	3	0
5	0.52	0	3	0	2	1

Formulare un problema di PLI che permetta di trovare una dieta di costo minimo sapendo che si devono assumere almeno 3 unità di proteine, 10 unità di carboidrati, 2 unità di grasso, 3 unità di vitamine e 2 unità di calcio e sapendo che se è presente l'alimento 1 la dieta non può contenere l'alimento 5.

Formulazione.

È un classico problema di miscelazione; le quantità di alimenti presenti nella dieta si suppongono frazionabili. A causa della presenza di una condizione logica, è necessario utilizzare, in aggiunta alle variabili del problema, una variabile 0 – 1 per modellarla cioè per esprimere con un vincolo il legame tra la presenza nella dieta dell'alimento 1 e dell'alimento 5.

– *Variabili di decisione.* Introduciamo come variabili del problema le unità di alimenti presenti nella dieta, x_i con $i = 1, \dots, 5$. Inoltre, introduciamo la variabile booleana $\delta \in \{0, 1\}$.

– *Vincoli.* Si hanno i seguenti vincoli:

- Vincoli di qualità: la dieta deve contenere alcuni valori minimi di sostanze nutrizionali; dalla tabella si ottiene che deve essere

$$x_2 + 5x_3 + 2x_4 \geq 3$$

$$7x_1 + 2x_4 + 3x_5 \geq 10$$

$$x_1 + 3x_2 + 4x_3 + x_4 \geq 2$$

$$x_1 + x_2 + 3x_4 + 2x_5 \geq 3$$

$$4x_2 + x_3 + x_5 \geq 2$$

- Vincolo logico: se nella dieta è presente l'alimento 1 allora non deve esserci l'alimento 5. Vogliamo quindi definire dei vincoli che consentano di esprimere le seguenti condizioni logiche

$$\begin{aligned} x_1 > 0 &\Rightarrow \delta = 1 \\ \delta = 1 &\Rightarrow x_5 = 0 \end{aligned}$$

Secondo quanto descritto, ciò può essere modellato introducendo i vincoli

$$x_1 - M\delta \leq 0$$

$$x_5 - M(1 - \delta) \leq 0$$

dove M è un numero positivo maggiore del più grande valore che possono assumere le variabili.

- Vincoli di non negatività: Si tratta di quantità di alimenti, e quindi deve essere

$$x_i \geq 0 \quad i = 1, \dots, 5.$$

– *Funzione obiettivo.* È il costo da minimizzare ed è data da

$$0.15x_1 + 0.23x_2 + 0.79x_3 + 0.47x_4 + 0.52x_5.$$

Complessivamente la formulazione di PLI per questo problema può essere scritta

$$\left\{ \begin{array}{l} \min (0.15x_1 + 0.23x_2 + 0.79x_3 + 0.47x_4 + 0.52x_5) \\ x_2 + 5x_3 + 2x_4 \geq 3 \\ 7x_1 + 2x_4 + 3x_5 \geq 10 \\ x_1 + 3x_2 + 4x_3 + x_4 \geq 2 \\ x_1 + x_2 + 3x_4 + 2x_5 \geq 3 \\ 4x_2 + x_3 + x_5 \geq 2 \\ x_1 - M\delta \leq 0 \\ x_5 - M(1 - \delta) \leq 0 \\ x_i \geq 0 \quad i = 1, \dots, 5 \\ \delta \in \{0, 1\}. \end{array} \right.$$

□

Si riportano di seguito i file `dieta.mod` e `dieta.dat` che realizzano un'implementazione AMPL di questo problema.

dieta.mod

```

set ALIMENTI;
set SOSTANZE;

param contenuti{ALIMENTI,SOSTANZE}>=0;
param costi{ALIMENTI}>=0;
param contenuti_min{SOSTANZE};
param BigM>0;

var x{ALIMENTI}>=0;
var d binary;

minimize costo : sum{i in ALIMENTI} costi[i]*x[i];

s.t. vincoli_qualita{j in SOSTANZE} :
    sum{i in ALIMENTI} contenuti[i,j]*x[i] >= contenuti_min[j];

s.t. vincolo_logico1 : x["A1"] - BigM * d <=0;
s.t. vincolo_logico2 : x["A4"] - (1-d)*BigM <=0;

```

dieta.dat

```

set ALIMENTI:= A1 A2 A3 A4 A5;
set SOSTANZE:= prot carb grassi vitam calcio;

param BigM := 10000;

param costi :=
A1  0.15
A2  0.23
A3  0.79
A4  0.47
A5  0.52;

param contenuti_min :=
prot 3
carb 10
grassi 2
vitam 3
calcio 2;

```

```

param contenuti : prot carb grassi vitam calcio :=
A1  0 7 1 1 0
A2  1 0 3 1 4
A3  5 0 4 0 1
A4  2 2 1 3 0
A5  0 3 0 2 1;

```

9.3.1 Problema del costo fisso

Esaminiamo un altro esempio di applicazione di variabili indicatrici: *il problema del costo fisso*. Nei modelli di PL la funzione obiettivo è una funzione lineare nelle variabili di decisione che, di solito, rappresentano livelli di attività. Questa ipotesi, in molti problemi pratici, non è verosimile: può infatti accadere che il costo di un'attività abbia un costo iniziale (set-up), ad esempio l'acquisto di un macchinario, che esiste solo se quell'attività è svolta a livello non nullo.

In riferimento ad un'applicazione industriale, indichiamo con c il costo della manifattura per unità di prodotto, con $f \geq 0$ il costo di set-up (costo fisso) e con $x \geq 0$ la quantità di prodotto da fabbricare.

Quindi se $x = 0$ il costo totale è ovviamente nullo; se $x > 0$ allora il costo totale è dato da $cx + f$. Quindi la funzione obiettivo è data dall'espressione

$$f(x) = \begin{cases} cx + f & \text{se } x > 0 \\ 0 & \text{se } x = 0. \end{cases}$$

Tale funzione ha una discontinuità nell'origine e quindi non è lineare (Figura 9.3.1).

Per formulare questo problema in termini di programmazione lineare, introduciamo una variabile indicatrice $\delta \in \{0, 1\}$ tale che, se il prodotto rappresentato dalla x è fabbricato in una qualsiasi quantità allora $\delta = 1$; se il prodotto non è fabbricato allora $\delta = 0$. Dobbiamo, quindi modellare con un vincolo le condizioni logiche

$$x > 0 \Rightarrow \delta = 1 \tag{9.3.5}$$

$$x = 0 \Rightarrow \delta = 0. \tag{9.3.6}$$

L'implicazione (9.3.5) si realizza introducendo il vincolo

$$x - M\delta \leq 0$$

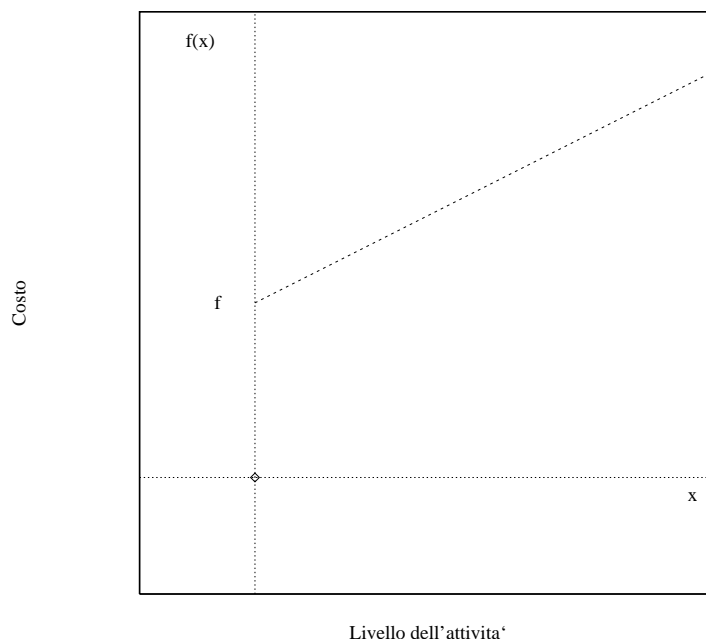


Figura 9.3.1 Problema del costo fisso.

dove M è un numero positivo maggiore del più grande valore che può assumere la x . Per realizzare l'implicazione (9.3.6), si dovrebbe introdurre un vincolo del tipo $x - \varepsilon\delta \geq 0$ con $\varepsilon > 0$; in realtà, ciò non è necessario perché, come vedremo, la condizione (9.3.6) discende direttamente dal fatto che ci troviamo in un problema di minimizzazione. Infatti, il problema può essere formulato come

$$\min (cx + f\delta)$$

con vincolo aggiuntivo

$$x - M\delta \leq 0$$

con $x \geq 0$ e $\delta \in \{0, 1\}$.

Dalla struttura della funzione discende immediatamente che se $x = 0$ allora, poiché si tratta di un problema di minimo, all'ottimo deve essere $\delta = 0$, essendo $f \geq 0$. Quindi non è necessario introdurre nella formulazione la condizione logica (9.3.6).

Si può facilmente generalizzare il problema del costo fisso al caso di n attività. Supponiamo che x_i , $i = 1, \dots, n$ rappresenti il livello al quale viene svolta ciascuna attività. Supponiamo che il costo della i -esima attività sia dato da

$$\begin{cases} c_i x_i + f_i & \text{se } x_i > 0 \\ 0 & \text{se } x_i = 0 \end{cases} \quad i = 1, \dots, n$$

dove $f_i \geq 0$ è il costo fisso dell'attività i -esima e deve essere pagato solo se l'attività i viene svolta ad un livello non nullo.

Il corrispondente problema di ottimizzazione è:

$$\min z(x) = \sum_{i=1}^n c_i x_i + \sum_{i \in I(x)} f_i$$

dove $I(x) = \{i \in \{1, \dots, n\} : x_i > 0\}$ e quindi è una funzione discontinua nell'origine, non lineare. Per formularlo come problema di PLI, si introduce per ogni $i = 1, \dots, n$ una variabile $\delta_i \in \{0, 1\}$ tale che

$$\delta_i = \begin{cases} 1 & \text{se l'attività } i \text{ è svolta a livello non nullo} \\ 0 & \text{se l'}i\text{-esima attività non è svolta.} \end{cases}$$

Si vuole quindi che siano verificate le seguenti condizioni logiche

$$x_i > 0 \quad \Rightarrow \quad \delta_i = 1, \quad x_i = 0 \quad \Rightarrow \quad \delta_i = 0.$$

Analogamente al caso precedente, il problema può essere formulato

$$\min \left(\sum_{i=1}^n c_i x_i + \sum_{i=1}^n \delta_i f_i \right)$$

con vincoli aggiuntivi

$$x_i - M\delta_i \leq 0 \quad i = 1, \dots, n$$

e con

$$x_i \geq 0, \quad \delta_i \in \{0, 1\} \quad i = 1, \dots, n.$$

È chiaro che se $x_i = 0$, allora all'ottimo $\delta_i = 0$ perché $f_i \geq 0$ e quindi la condizione logica $x_i = 0 \Rightarrow \delta_i = 0$ è automaticamente verificata. Inoltre, se $x_i > 0$ allora $\delta_i = 1$ e quindi il suo costo fisso si aggiungerà al valore della funzione costo nella funzione obiettivo. È quindi evidente che una soluzione ottima del problema iniziale è anche ottima per il problema trasformato.

Esempio 9.3.2 *In una centrale elettrica sono a disposizione tre generatori e ogni giorno si deve decidere quali usare di giorno e quali di notte per assicurare una produzione di almeno 4000 megawatts di giorno e di almeno 2800 megawatts di notte. L'uso di un generatore comporta la presenza di personale tecnico che sorvegli il suo funzionamento; tale personale viene retribuito in maniera diversa tra il giorno e la notte e a seconda del tipo di generatore; tali costi di attivazione sono riportati nella tabella che segue (in euro) insieme al costo (in euro) per ogni megawatt prodotta e alla massima capacità di produzione in megawatts per ogni singolo periodo (giorno/notte).*

	Costo attivazione		Costo per	Capacità
	giorno	notte	megawatt	max
Generatore A	750	1000	3	2000
Generatore B	600	900	5	1700
Generatore C	800	1100	6	2500

Formulare un modello di PLI che permetta di rappresentare il problema in analisi.

Formulazione.

È un problema di costo fisso e può essere formulato in termini di Programmazione Lineare Interpolata come appena descritto in generale. Per brevità di notazione, chiameremo 1° periodo il giorno e 2° periodo la notte.

– *Variabili.* Indichiamo con x_{A_i} , x_{B_i} e x_{C_i} , $i = 1, 2$, i megawatts generati rispettivamente dai generatori A, B e C nel periodo i . Inoltre, per ottenere una formulazione lineare, è necessario introdurre sei variabili 0 – 1, δ_{A_i} , δ_{B_i} e δ_{C_i} , $i = 1, 2$, definite come segue :

$$\delta_{A_i} = \begin{cases} 1 & \text{se il generatore A è attivato nell}'i\text{-esimo periodo} \\ 0 & \text{se nell}'i\text{-esimo periodo il generatore A non è attivato} \end{cases} \quad i = 1, 2.$$

Analoga è la definizione per le altre variabili δ_{B_i} e δ_{C_i} , $i = 1, 2$.

– *Funzione obiettivo.* La funzione obiettivo da minimizzare può esser scritta

$$3x_{A_1} + 3x_{A_2} + 5x_{B_1} + 5x_{B_2} + 6x_{C_1} + 6x_{C_2} + 750\delta_{A_1} + \\ + 1000\delta_{A_2} + 600\delta_{B_1} + 900\delta_{B_2} + 800\delta_{C_1} + 1100\delta_{C_2}.$$

– *Vincoli.* Si devono considerare i vincoli sulla richiesta cioè

$$x_{A_1} + x_{B_1} + x_{C_1} \geq 4000$$

$$x_{A_2} + x_{B_2} + x_{C_2} \geq 2800.$$

Inoltre, per quanto esposto nel caso generale si devono considerare i vincoli

$$x_{A_i} - 2000\delta_{A_i} \leq 0 \quad i = 1, 2$$

$$x_{B_i} - 1700\delta_{B_i} \leq 0 \quad i = 1, 2$$

$$x_{C_i} - 2500\delta_{C_i} \leq 0 \quad i = 1, 2.$$

Quindi la formulazione complessiva può essere scritta

$$\left\{ \begin{array}{l} \min \left(3x_{A_1} + 3x_{A_2} + 5x_{B_1} + 5x_{B_2} + 6x_{C_1} + 6x_{C_2} + \right. \\ \quad \left. + 750\delta_{A_1} + 1000\delta_{A_2} + 600\delta_{B_1} + 900\delta_{B_2} + 800\delta_{C_1} + 1100\delta_{C_2} \right) \\ x_{A_1} + x_{B_1} + x_{C_1} \geq 4000 \\ x_{A_2} + x_{B_2} + x_{C_2} \geq 2800 \\ x_{A_1} - 2000\delta_{A_1} \leq 0 \\ x_{B_1} - 1700\delta_{B_1} \leq 0 \\ x_{C_1} - 2500\delta_{C_1} \leq 0 \\ x_{A_2} - 2000\delta_{A_2} \leq 0 \\ x_{B_2} - 1700\delta_{B_2} \leq 0 \\ x_{C_2} - 2500\delta_{C_2} \leq 0 \\ x_{A_i} \geq 0, x_{B_i} \geq 0, x_{C_i} \geq 0, \quad i = 1, 2 \\ \delta_{A_i} \in \{0, 1\}, \delta_{B_i} \in \{0, 1\}, \delta_{C_i} \in \{0, 1\} \quad i = 1, 2. \end{array} \right.$$

□

Si riportano di seguito i file `centrale.mod` e `centrale.dat` che rappresentano una implementazione AMPL del problema in esame.

```

centrale.mod
set GENERATORI;
set PERIODI;

param costo_fisso{GENERATORI,PERIODI}>=0;
param costo_unitario{GENERATORI}>=0;
param capacita_max{GENERATORI};
param produzione_min{PERIODI};
param BigM = 10^6;

var x{GENERATORI,PERIODI} >=0;
var d{GENERATORI,PERIODI} binary;

minimize costo_totale : sum{i in GENERATORI, j in PERIODI}
    (costo_unitario[i]*x[i,j]+costo_fisso[i,j]*d[i,j]);

s.t. produzione_minima{j in PERIODI} : sum{i in GENERATORI}
    x[i,j]>= produzione_min[j];

s.t. vincoli_logici{i in GENERATORI, j in PERIODI} :
    x[i,j] - capacita_max[i]*d[i,j] <=0;

```

centrale.dat

```

set GENERATORI := A, B, C;
set PERIODI := giorno notte;

param : costo_unitario      capacita_max :=
  A      3                  2000
  B      5                  1700
  C      6                  2500;

param produzione_min :=
  giorno    4000
  notte     2800;

param costo_fisso : giorno    notte :=
                A      750    1000
                B      600    900
                C      800    1100;

```

9.3.2 Problemi di “lot sizing” (gestione della scorte)

I modelli multiperiodo esaminati nel paragrafo 3.4.1 rientrano in una classe di modelli per la programmazione della produzione che va sotto il nome di *Modelli per la gestione della scorte* (“*lot sizing*”) che anche da un punto di vista storico costituiscono un argomento centrale della Ricerca Operativa

Attualmente negli USA alcune indagini hanno evidenziato che il 50% delle aziende americane di produzione utilizzano strumenti matematici per la gestione ottima delle scorte. C'è la necessità di integrare la fase produttiva con quella della gestione delle scorte. L'utilizzazione di scorte nei processi produzione ha numerosi vantaggi:

- *economia di scala* che si possono conseguire aumentando i volumi produttivi minimizzando l'incidenza dei costi fissi;
- *flessibilità della produzione*: si riesce a far fronte con le scorte all'eventuale andata fuori servizio di qualche linea di produzione;
- *equipartizione dei carichi di lavori* sull'intero orizzonte produttivo.

Un problema di “lot sizing” si può formalizzare nel seguente modo: si tratta di pianificare la fabbricazione di un bene in assegnato un orizzonte temporale costituito da un insieme finito di periodi di controllo $T = \{1, \dots, t\}$. Per ogni periodo $i \in \{1, \dots, t\}$ è nota la richiesta di questo bene (che deve essere soddisfatta

esattamente) che indichiamo con d_i . Sono noti i costi unitari c_i , $i = 1, \dots, t$ di produzione del bene in ciascun periodo ed inoltre in ogni periodo, ad eccezione dell'ultimo, è possibile immagazzinare quantità di questo bene che andrà a fare parte della quantità di bene disponibile nel periodo successivo. Anche il costo di stockaggio unitario è assegnato ed è pari a b_i . La novità rispetto ai modelli multiperiodo consiste nella presenza di costi di setup corrispondenti all'avviamento della produzione in ciascun periodo; si tratta di costi fissi che non dipendono dalle quantità prodotte e vengono sostenuti solamente se si produce qualcosa nel periodo; indichiamo con f_i questi costi fissi.

Il problema consiste nel determinare le quantità di bene da produrre in ciascun periodo e le quantità da immagazzinare in modo da soddisfare le richieste minimizzando il costo complessivo dato dalla somma dei costi di produzione e di stockaggio tenendo conto che all'inizio del primo periodo non c'è nessuna scorta disponibile e che nell'ultimo periodo non si può effettuare alcuno stockaggio.

Formulazione.

– *Variabili.* Indichiamo con x_i , $i = 1, \dots, t$ il livello di produzione nel periodo i -esimo, cioè le quantità del bene da produrre in quel periodo. Indichiamo inoltre con s_i , $i = 1, \dots, t-1$ le quantità di bene che vengono immagazzinate nel periodo i . Inoltre, per $i = 1, \dots, t$ introduciamo le seguenti variabili 0 – 1:

$$\delta_i = \begin{cases} 1 & \text{se nell}'i\text{-esimo periodo c'è produzione} \\ 0 & \text{altrimenti;} \end{cases}$$

Il problema può essere efficacemente rappresentato come in Figura 9.3.2

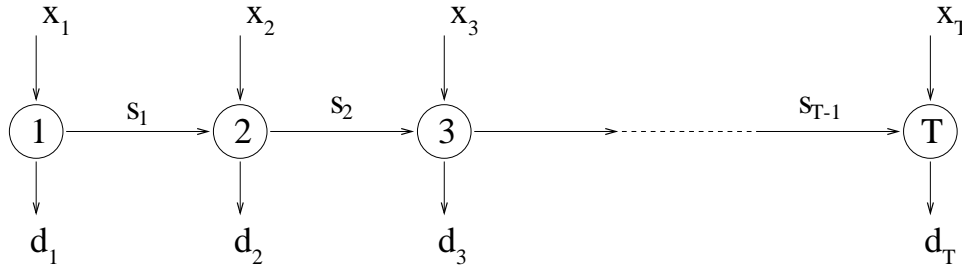


Figura 9.3.2 Un problema di “Lot sizing”

– *Funzione obiettivo.* La funzione obiettivo sarà data dalla somma dei costi di produzione e dei costi di stockaggio e quindi può essere scritta nella forma

$$\sum_{i=1}^t c_i x_i + \sum_{i=1}^{t-1} b_i s_i + \sum_{i=1}^t f_i \delta_i$$

– *Vincoli.* I vincoli del problema sono i seguenti già esaminati nel caso di modelli multiperiodo:

$$x_1 = d_1 + s_1$$

$$\begin{aligned}
s_{i-1} + x_i &= d_i + s_i, & i = 2, \dots, t-1 \\
s_{t-1} + x_t &= d_t \\
x_1 \geq 0, x_2 \geq 0, &\dots, x_t \geq 0, \\
s_1 \geq 0, s_2 \geq 0, &\dots, s_{t-1} \geq 0
\end{aligned}$$

Inoltre si devono considerare i vincoli relativi alla presenza dei costi fissi, ovvero i vincoli

$$x_i - M\delta_i \leq 0 \quad i = 1, \dots, t$$

dove M , ad esempio, può essere scelta pari a $\sum_{i=1}^t d_i$, cioè pari a quanto viene richiesto durante l'intero orizzonte temporale.

Quindi la formulazione complessiva di un problema di “lot sizing” si può scrivere come

$$\left\{ \begin{array}{l}
\min \left(\sum_{i=1}^t c_i x_i + \sum_{i=1}^{i-1} b_i s_i + \sum_{i=1}^t f_i \delta_i \right) \\
x_1 - s_1 = d_1 \\
s_{i-1} + x_i - s_i = d_i \quad i = 2, \dots, t-1 \\
s_{t-1} + x_t = d_t, \\
x_i - M\delta_i \leq 0 \quad i = 1, \dots, t \\
x_1 \geq 0, \dots, x_t \geq 0, \\
s_1 \geq 0, \dots, s_{t-1} \geq 0 \\
\delta_1 \in \{0, 1\}, \dots, \delta_t \in \{0, 1\}.
\end{array} \right.$$

□

9.3.3 Problemi di localizzazione di impianti

Si tratta di problemi che nascono nell'ambito della pianificazione industriale che possono essere schematizzati nel seguente modo: sono date n aree $\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_n$, distribuite in un territorio. In ciascuna di esse è possibile costruire una fabbrica che produce merce. Per ciascuna area \mathbf{A}_i , $i = 1, \dots, n$ è nota la massima capacità produttiva p_i , $i = 1, \dots, n$ che una fabbrica avrebbe se fosse localizzata in \mathbf{A}_i . Sia inoltre f_i il costo fisso di costruzione della fabbrica nell'area \mathbf{A}_i . Sono inoltre dati m siti $\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_m$, ove risiedono clienti ai quali deve essere trasportate la merce prodotta. Per ciascun sito \mathbf{C}_j è assegnato un quantitativo r_j , $j = 1, \dots, m$, di una data merce richiesta presso il sito \mathbf{C}_j . Tale richiesta deve essere soddisfatta esattamente. Per soddisfare questa richiesta possono essere costruite $q \leq n$ fabbriche che producono la merce. Esistono altri costi fissi dovuti alla eventuale costruzione di una strada dall'area \mathbf{A}_i al sito \mathbf{C}_j , per ogni $i = 1, \dots, n$

e $j = 1, \dots, m$; indicheremo questi costi fissi con f_{ij} . Siano inoltre c_{ij} il costo necessario per trasportare una unità di merce dalla fabbrica costruita nell'area \mathbf{A}_i al sito \mathbf{C}_j e M_{ij} il quantitativo massimo di merce trasportabile. Il problema consiste nel determinare quante fabbriche e su quali aree costruirle, insieme a quali strade di collegamento costruire, in modo da soddisfare le richieste di i siti minimizzando i costi di costruzione delle fabbriche, delle strade di collegamento e il costo del trasporto della merce una volta che le costruzioni delle fabbriche sono state ultimate determinando al tempo stesso il piano per il trasporto della merce per soddisfare tutte le richieste.

Questo problema può essere formulato come problema di Programmazione Lineare Intera nel seguente modo: si introducono le seguenti variabili

$$\delta_i = \begin{cases} 1 & \text{se una fabbrica è costruita sull'area } \mathbf{A}_i \\ 0 & \text{altrimenti;} \end{cases}$$

$$y_{ij} = \begin{cases} 1 & \text{se una strada è costruita da } \mathbf{A}_i \text{ a } \mathbf{C}_j \\ 0 & \text{altrimenti.} \end{cases}$$

Si introducono inoltre le variabili x_{ij} che rappresentano la quantità di merce trasportata dalla fabbrica costruita nell'area \mathbf{A}_i al sito \mathbf{C}_j .

I vincoli sono innanzitutto i vincoli di richiesta

$$\sum_{i=1}^n x_{ij} = r_j \quad \text{per ogni } j = 1, \dots, m.$$

Inoltre per ogni $i = 1, \dots, m$, si vuole che se $\sum_{j=1}^m x_{ij} > 0$ allora $\delta_i = 1$. Questa implicazione si realizza con i vincoli

$$\sum_{j=1}^m x_{ij} - p_i \delta_i \leq 0 \quad i = 1, \dots, n.$$

Ragionando analogamente si ottengono i vincoli

$$x_{ij} - M_{ij} y_{ij} \leq 0 \quad i = 1, \dots, n, \quad j = 1, \dots, m.$$

Infine dovrà essere

$$\sum_{i=1}^n \delta_i \leq q.$$

Si devono poi esplicitare i vincoli $x_{ij} \geq 0$ e $\delta_i \in \{0, 1\}$, $y_{ij} \in \{0, 1\}$, $i = 1, \dots, n$, $j = 1, \dots, m$.

La funzione obiettivo si può quindi scrivere

$$\sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} + \sum_{i=1}^n f_i \delta_i + \sum_{i=1}^n \sum_{j=1}^m f_{ij} y_{ij}.$$

Esaminiamo, ora, un esempio molto semplice di problema di localizzazione di impianti.

Esempio 9.3.3 Una compagnia di distribuzione deve rifornire i suoi clienti C_1 , C_2 , C_3 , C_4 e C_5 che sono dislocati in località diverse di una regione. Per ottimizzare il rifornimento la compagnia vuole costruire un numero di depositi non superiore a due disponendo di tre possibili zone dove costruirli. A seconda della zona in cui vengono costruiti, i tre possibili depositi hanno un costo di costruzione e una capacità massima diversi. La tabella che segue riporta questi costi in migliaia di euro e queste capacità in tonnellate.

	Costo costruzione	Capacità massima
Deposito 1	10000	180
Deposito 2	15000	230
Deposito 3	13000	500

Il quantitativo di merce (in tonnellate) richiesto da ciascun cliente è riportato nella tabella che segue insieme ai costi (in migliaia di euro) del trasporto di una unità di merce da ciascuno dei possibili depositi a ciascun cliente.

	C_1	C_2	C_3	C_4	C_5
Richiesta	91	170	135	153	110
Deposito 1	15	13	27	9	7
Deposito 2	12	21	34	21	3
Deposito 3	7	10	2	17	12

Costruire un modello lineare che rappresenti il problema in analisi per soddisfare esattamente la richiesta minimizzando il costo complessivo trascurando la possibilità di costruire ulteriori collegamenti rispetto a quelli esistenti e supponendo che non ci siano limitazioni sulle quantità massime di merci trasportabili.

Formulazione.

È un problema che rientra nello schema generale di un problema di localizzazione di impianti e quindi può essere formulato in termini di Programmazione Lineare Intera come appena descritto nel caso generale.

– *Variabili.* È sufficiente introdurre le variabili binarie

$$\delta_i = \begin{cases} 1 & \text{se è costruito l'}i\text{-esimo deposito} \\ 0 & \text{altrimenti} \end{cases}$$

e le variabili x_{ij} che rappresentano la quantità di merce da trasportare dal deposito i -esimo alla zona j -esima.

– *Funzione obiettivo.* La funzione obiettivo da minimizzare sarà

$$15x_{11} + 13x_{12} + 27x_{13} + 9x_{14} + 7x_{15} + 12x_{21} + 21x_{22} + 34x_{23} + 21x_{24} + 3x_{25} \\ + 7x_{31} + 10x_{32} + 2x_{33} + 17x_{34} + 12x_{35} + 10000\delta_1 + 15000\delta_2 + 13000\delta_3.$$

– *Vincoli.* I vincoli da considerare sono innanzitutto i vincoli di richiesta

$$\sum_{i=1}^3 x_{i1} = 91, \quad \sum_{i=1}^3 x_{i2} = 170, \quad \sum_{i=1}^3 x_{i3} = 135, \quad \sum_{i=1}^3 x_{i4} = 153, \quad \sum_{i=1}^3 x_{i5} = 110.$$

Inoltre

$$\sum_{j=1}^5 x_{1j} - 180\delta_1 \leq 0,$$

$$\sum_{j=1}^5 x_{2j} - 230\delta_2 \leq 0,$$

$$\sum_{j=1}^5 x_{3j} - 500\delta_3 \leq 0.$$

Poiché non si possono costruire più di due depositi, si deve poi imporre che

$$\delta_1 + \delta_2 + \delta_3 \leq 2.$$

Naturalmente devono essere anche esplicitati i vincoli

$$x_{ij} \geq 0 \quad \delta_i \in \{0, 1\} \quad i = 1, 2, 3 \quad j = 1, 2, 3, 4, 5.$$

Quindi la formulazione complessiva è:

$$\left\{ \begin{array}{l} \min \left(15x_{11} + 13x_{12} + 27x_{13} + 9x_{14} + 7x_{15} + 12x_{21} + 21x_{22} + 34x_{23} \right. \\ \quad \left. + 21x_{24} + 3x_{25} + 7x_{31} + 10x_{32} + 2x_{33} + 17x_{34} + 12x_{35} \right. \\ \quad \left. + 10000\delta_1 + 15000\delta_2 + 13000\delta_3 \right) \\ \sum_{i=1}^3 x_{i1} = 91 \\ \sum_{i=1}^3 x_{i2} = 170 \\ \sum_{i=1}^3 x_{i3} = 135 \\ \sum_{i=1}^3 x_{i4} = 153 \\ \sum_{i=1}^3 x_{i5} = 110 \\ \sum_{j=1}^5 x_{1j} - 180\delta_1 \leq 0 \\ \sum_{j=1}^5 x_{2j} - 230\delta_2 \leq 0 \\ \sum_{j=1}^5 x_{3j} - 500\delta_3 \leq 0 \\ \delta_1 + \delta_2 + \delta_3 \leq 2 \\ x_{ij} \geq 0 \quad \delta_i \in \{0, 1\} \quad i = 1, 2, 3 \quad j = 1, 2, 3, 4, 5 \end{array} \right.$$

□

Anche in questo caso riportiamo di seguito i file del modello e dei dati che realizzano una implementazione AMPL del problema in analisi.

```
localizzazione.mod
```

```
set CLIENTI;
```

```

set ZONE;

param costo_costruzione{ZONE}>=0;
param capacita_max{ZONE}>=0;
param richieste{CLIENTI}>=0;
param costi_trasp{ZONE,CLIENTI}>=0;

var x{ZONE,CLIENTI}>=0;
var d{ZONE} binary;

minimize costo_totale : sum{i in ZONE, j in CLIENTI}
    costi_trasp[i,j]*x[i,j]+sum{i in ZONE}costo_costruzione[i]*d[i];

s.t. vincoli_domanda{j in CLIENTI} :
    sum{i in ZONE} x[i,j]=richieste[j];
s.t. vincoli_logici{i in ZONE} : sum{j in CLIENTI}
    x[i,j] - capacita_max[i]*d[i]<=0;
s.t. max_depositi : sum{i in ZONE} d[i] <= 2;

```

localizzazione.dat

```

set CLIENTI:= C1 C2 C3 C4 C5;
set ZONE := dep1 dep2 dep3;

param : costo_costruzione capacita_max :=
dep1 10000 180
dep2 15000 230
dep3 13000 500;

param richieste :=
C1 91
C2 170
C3 135
C4 153
C5 110;

param costi_trasp : C1 C2 C3 C4 C5 :=
dep1 15 13 27 9 7
dep2 12 21 34 21 3
dep3 7 10 2 17 12;

```

9.4 VARIABILI BINARIE PER INDICARE IL SODDISFACIMENTO DI VINCOLI DISGIUNTIVI

Nell'usuale definizione di problemi di ottimizzazione si assume che tutti i vincoli debbano essere soddisfatti simultaneamente da una soluzione ammissibile. Tuttavia in molte applicazioni può accadere che solo un sottoinsieme dei vincoli debba essere soddisfatto e che tale sottoinsieme sia specificato dal valore che assume un'opportuna variabile di decisione. In questo caso si dice che i vincoli sono *disgiuntivi*.

Come esempio di questo uso delle variabili binarie, analizziamo una importante classe di problemi.

9.4.1 Problemi di “scheduling” (sequenziamento)

Si tratta di problemi di produzione in cui si deve decidere l'ordine di processamento di una sequenza di lavori su una macchina in grado di eseguire un lavoro alla volta (capacità unitaria). Si deve quindi esprimere la condizione disgiuntiva

$$\begin{aligned} &\text{“il lavoro } i\text{-esimo precede il lavoro } j\text{-esimo”} \\ &\quad \text{oppure} \\ &\text{“il lavoro } j\text{-esimo precede il lavoro } i\text{-esimo”}. \end{aligned}$$

Questo tipo di problema si presenta spesso in ambito industriale e nei sistemi di elaborazione.

Formalmente si ha la seguente situazione: siano dati n lavori *indipendenti* (il tempo di esecuzione di ciascun lavoro non dipende da quando viene eseguito rispetto agli altri lavori) e *indivisibili* (ciascun lavoro deve essere completato prima di poter eseguire il successivo).

Supponiamo inoltre che ciascun lavoro sia presente nel sistema fin dall'inizio, cioè che la macchina possa iniziare la lavorazione di un qualunque lavoro in qualsiasi istante.

Sia noto p_i , $i = 1, \dots, n$ il tempo di processamento di ciascun lavoro sulla macchina.

Il problema consiste nel determinare la sequenza di lavorazione dei lavori sulla macchina, cioè gli istanti t_i , $i = 1, \dots, n$ in cui la macchina inizia la lavorazione del lavoro i -esimo, in modo da ottimizzare un opportuno criterio.

Avendo introdotto le variabili t_i indicanti gli istanti di tempo in cui la macchina inizia a processare l' i -esimo lavoro, formulare un problema di scheduling significa determinare i vincoli sulle variabili t_i in modo che esse rappresentino sequenze effettivamente realizzabili sulla macchina.

Formulazione.

– *Variabili.* Introduciamo formalmente le seguenti variabili: per indicare se il lavoro i precede il lavoro j o viceversa, per ogni $1 \leq i < j \leq n$, si introducono le

variabili 0 – 1 così definite

$$y_{ij} = \begin{cases} 1 & \text{se il lavoro } i \text{ precede il lavoro } j \\ 0 & \text{se il lavoro } j \text{ precede il lavoro } i. \end{cases}$$

Si introducono, inoltre, le variabili temporali t_i , $i = 1, \dots, n$ indicanti gli istanti di tempo di inizio dei lavori.

– *Vincoli.* Come già osservato, la macchina ha capacità unitaria e deve completare un lavoro prima di iniziarne un altro. Quindi uno solo dei due vincoli “il lavoro i -esimo precede il lavoro j -esimo”, oppure “il lavoro j -esimo precede il lavoro i -esimo” deve essere soddisfatto.

Se il lavoro i è iniziato sulla macchina prima del lavoro j , si deve avere

$$t_j \geq t_i + p_i.$$

Se invece il lavoro j inizia prima del lavoro i , allora si deve avere

$$t_i \geq t_j + p_j.$$

Si devono, quindi, esplicitare le seguenti condizioni logiche:

$$y_{ij} = 1 \quad \Rightarrow \quad t_i - t_j \leq -p_i \quad (9.4.1)$$

$$y_{ij} = 0 \quad \Rightarrow \quad t_j - t_i \leq -p_j. \quad (9.4.2)$$

Se M è un limite superiore sia per $t_i - t_j + p_i$ sia per $t_j - t_i + p_j$, allora usando la (9.3.1), le condizioni (9.4.1) e (9.4.2) possono essere rispettivamente modellate dai vincoli

$$t_i - t_j + p_i \leq M(1 - y_{ij}) \quad 1 \leq i < j \leq n \quad (9.4.3)$$

$$t_j - t_i + p_j \leq M y_{ij} \quad 1 \leq i < j \leq n. \quad (9.4.4)$$

Infatti se $y_{ij} = 1$ la (9.4.3) esprime la condizione che la lavorazione del lavoro j può iniziare solo dopo il completamento del lavoro i mentre la (9.4.4) è sempre soddisfatta (per la scelta di M) e quindi non introduce ulteriori restrizioni. Se $y_{ij} = 0$, allora la (9.4.4) esprime la condizione che la lavorazione del lavoro i può iniziare solo dopo il completamento del lavoro j , mentre la (9.4.3) è sempre soddisfatta e quindi non introduce alcuna ulteriore restrizione. La (9.4.3) e la (9.4.4) sono di solito chiamati *vincoli di sequenziamento*.

Si devono inoltre esplicitare i vincoli di non negatività sulle variabili t_i , cioè

$$t_i \geq 0 \quad i = 1, \dots, n.$$

Si può riassumere quanto fino ad ora esposto nel seguente risultato:

Teorema 9.4.1 *Se un vettore $(t, y)^T$ con $t \in \mathbb{R}^n$ ed $y \in \{0, 1\}^{n \times n}$ soddisfa il sistema*

$$\begin{cases} t_i - t_j + p_i \leq M(1 - y_{ij}) \\ t_j - t_i + p_j \leq My_{ij} \end{cases} \quad 1 \leq i < j \leq n$$

allora ciascuna componente del vettore t rappresenta un istante ammissibile di inizio processamento per il corrispondente lavoro. Viceversa, per ogni vettore ammissibile t esiste sicuramente un vettore y (che rappresenta l'ordine di processamento dei lavori sulla macchina) tale che il vettore (t, y) è ammissibile per il precedente sistema di vincoli.

Naturalmente possono essere facilmente inseriti nel modello vincoli di precedenza o altre restrizioni temporali aggiungendo vincoli lineari sulle variabili t ed y .

– *Funzione obiettivo.* Nei problemi di scheduling la funzione obiettivo è di solito costruita in modo da ottimizzare un opportuno criterio. Analizziamo, ora, due dei criteri più diffusi:

a) *Tempo medio di permanenza nel sistema.*

Ogni istante t_i può essere, infatti, anche interpretato come tempo di attesa nel sistema del lavoro i prima di essere processato. Quindi, il tempo medio di permanenza nel sistema può essere scritto

$$\frac{\sum_{i=1}^n (t_i + p_i)}{n}.$$

b) *Tempo complessivo di utilizzazione della macchina.*

Questo criterio è significativo nel caso dell'uso di più macchine, perché nel caso di una sola macchina questo tempo complessivo è noto; infatti esso è dato da $\sum_{i=1}^n p_i$. Tuttavia anche in questo caso esso è esprimibile come nel caso generale cioè nella forma

$$\max_{1 \leq i < j \leq n} (t_i + p_i).$$

Si osservi che questa funzione obiettivo da minimizzare è di tipo “max” e quindi non è lineare.

Analizziamo, ora, un semplice esempio di problema di scheduling.

Esempio 9.4.1 *Sia data una macchina a capacità unitaria che deve effettuare tre lavori aventi tempo di processamento $p_1 = 2$, $p_2 = 3$, $p_3 = 4$. Formulare il problema di scheduling che consenta di determinare la sequenza che minimizza il tempo medio di permanenza nel sistema, tenendo conto che, se il primo lavoro precede il secondo, l'inizio del terzo lavoro deve aspettare un tempo $\Delta_3 = 2$ dopo*

il termine del secondo lavoro, mentre, se il terzo lavoro precede il primo, l'inizio del secondo deve attendere un tempo $\Delta_2 = 3$ dopo il termine del primo lavoro.

Formulazione.

Formuliamo questo problema come appena esposto nel caso generale.

– *Variabili.* Introduciamo tre variabili continue t_1, t_2, t_3 , indicanti gli istanti di inizio dei lavori sulla macchina e tre variabili 0 – 1 per esprimere i vincoli di sequenziamento così definite:

$$y_{ij} = \begin{cases} 1 & \text{se il lavoro } i \text{ precede il lavoro } j \\ 0 & \text{se il lavoro } j \text{ precede il lavoro } i \end{cases} \quad 1 \leq i < j \leq 3.$$

– *Vincoli di sequenziamento.* Introducendo una costante positiva M che sia una limitazione superiore per $t_i - t_j + p_i$ e per $t_j - t_i + p_j$, i vincoli di sequenziamento possono essere scritti

$$\begin{aligned} t_1 - t_2 + 2 &\leq M(1 - y_{12}) \\ t_2 - t_1 + 3 &\leq My_{12} \\ t_1 - t_3 + 2 &\leq M(1 - y_{13}) \\ t_3 - t_1 + 4 &\leq My_{13} \\ t_2 - t_3 + 3 &\leq M(1 - y_{23}) \\ t_3 - t_2 + 4 &\leq My_{23} \end{aligned}$$

– *Altri vincoli.* Gli altri vincoli di attese reciproche possono essere rappresentati utilizzando le variabili binarie precedentemente introdotte e la costante positiva M .

$$\begin{aligned} t_2 + 3 + 2 - t_3 &\leq M(1 - y_{12}) \\ t_1 + 2 + 3 - t_2 &\leq My_{13} \end{aligned}$$

Inoltre, si devono esplicitare i vincoli di non negatività

$$t_1 \geq 0 \quad t_2 \geq 0 \quad t_3 \geq 0.$$

– *Funzione obiettivo.* La funzione obiettivo da minimizzare è data dal tempo medio di permanenza nel sistema e quindi può essere scritta

$$\frac{1}{3}(t_1 + 2 + t_2 + 3 + t_3 + 4).$$

La formulazione finale sarà quindi

$$\left\{ \begin{array}{l} \min \frac{1}{3}(t_1 + t_2 + t_3 + 9) \\ t_1 - t_2 + 2 \leq M(1 - y_{12}) \\ t_2 - t_1 + 3 \leq My_{12} \\ t_1 - t_3 + 2 \leq M(1 - y_{13}) \\ t_3 - t_1 + 4 \leq My_{13} \\ t_2 - t_3 + 3 \leq M(1 - y_{23}) \\ t_3 - t_2 + 4 \leq My_{23} \\ t_2 + 3 + 2 - t_3 \leq M(1 - y_{12}) \\ t_1 + 2 + 3 - t_2 \leq My_{13} \\ t_1 \geq 0 \quad t_2 \geq 0 \quad t_3 \geq 0 \\ y_{12} \in \{0, 1\}, \quad y_{13} \in \{0, 1\}, \quad y_{23} \in \{0, 1\}. \end{array} \right.$$

□