$\max \; c'x + d'y$

$\text{sa} \quad Ax + Ey = b$

$x \in \mathbb{Z}_+ \,, \; y \geq 0$

problema di knapsack (zaino) : $\max \sum_{j=1}^{n} p_j x_j$ → ho n oggetti, ognuno ha il proprio "costo" p

$\text{sa} \sum_{j=1}^{n} w_j x_j \leq b$ → ogni oggetto ha il proprio volume w, vincolo da rispettare

$x_j \in \{0,1\} \quad \forall j$ → x variabile decisionale $x \begin{cases} 1 & \text{se lo prendo} \\ 0 & \text{se non lo prendo} \end{cases}$

es:

item	1	2	3	4	5
w	8	7	11	6	15
p	23	15	28	14	44

$\max \; 23x_1 + 15x_2 + 28x_3 + 14x_4 + 44x_5$

$\text{sa} \; 8x_1 + 7x_2 + 11x_3 + 6x_4 + 15x_5 \leq 25$

$x_i \in \{0;1\}$

es: pianificazione multi periodo con lotti minimi

n prodotti, m risorse produttive, T periodi produttivi

c_{jt} = margine del prodotto j nel periodo t

a_{ij} = utilizzo risorsa i per il prodotto j

b_{it} = disponibilità risorsa i nel periodo t

d_{jt} = domanda prodotto j nel periodo t

h_{jt} = prezzo scorta del prodotto j nel periodo t

l_j = lotto minimo del prodotto j

P_{jt} = quantità da produrre di j nel periodo t

I_{jt} = quantità scorta di j nel periodo t

y_{jt} = variabile binaria $\begin{cases} = 1 & \text{j viene prodotto in t} \\ = 0 & \end{cases}$

$$\min \sum_{t=1}^{T} \sum_{j=1}^{n} \left(c_{jt} \cdot \underbrace{P_{jt}}_{\text{quantità}} + h_{jt} \cdot \underbrace{I_{jt}}_{\text{quantità scorta}} \right)$$

$\text{sa} \quad P_{jt} + I_{j,t-1} - I_{jt} = d_{jt} \quad \forall j \; \forall t$ vincolo domanda

$\sum_{j=1}^{n} a_{ij} P_{jt} \leq b_{it} \quad \forall i, \forall t$ vincolo disponibilità

$P_{jt} \geq l_{jt} \cdot y_{jt} \quad \forall j, \forall t$ vincolo lotto minimo

$P_{jt} \leq M \, y_{jt} \quad \forall j, \forall t$ vincolo di consistenza logica

es: domanda

periodo	F	R
1	16	16
2	28	14
3	28	18
lotto minimo	23	18

f. ob.: $\min\ 24(F1+F2+F3) + 18(R1+R2+R3) + 0{,}2(IF1+IF2) + 0{,}03(IR1+IR2)$

vincoli:

- disponibilità
  - $SM1: F1 + R1 \le 40$ — periodo 1
  - $SP1: 4F1 + 2R1 \le 132$
  - $SC1: 2F1 + 4R1 \le 140$
  - $SM2: F2 + R2 \le 40$ — periodo 2
  - $SP2: 4F2 + 2R2 \le 132$
  - $SC2: 2F2 + 4R2 \le 140$
  - $SM3: F3 + R3 \le 40$ — periodo 3
  - $SP3: 4F3 + 2R3 \le 132$
  - $SC3: 2F3 + 4R3 \le 140$

- domanda
  - $DF1: F1 - IF1 = 16$ — periodo 1
  - $DR1: R1 - IR1 = 16$
  - $DF2: F2 + IF1 - IF2 = 28$ — periodo 2
  - $DR2: R2 + IR1 - IR2 = 14$
  - $DF3: F3 + IF2 = 28$ — periodo 3
  - $DR3: R3 + IR3 = 18$

se produco = 1 → $F1 \ge 23$; se non produco = 0 $F1 \ge 0$

- lotti minimi
  - $LF1: F1 - 23\,y_{F1} \ge 0$ — periodo 1
  - $LR1: R1 - 16\,y_{R1} \ge 0$
  - $LF2: F2 - 23\,y_{F2} \ge 0$ — periodo 2
  - $LR2: R2 - 16\,y_{R2} \ge 0$
  - $LF3: F3 - 23\,y_{F3} \ge 0$ — periodo 3
  - $LR3: R3 - 16\,y_{R3} \ge 0$

se non produco = 0 → $F1 \le 0$; se produco = 1 → $F1 \le 500$

- consistenza logica
  - $UF1: F1 - 500\,y_{F1} \le 0$ — periodo 1
  - $UR1: R1 - 500\,y_{R1} \le 0$
  - $UF2: F2 - 500\,y_{F2} \le 0$ — periodo 2
  - $UR2: R2 - 500\,y_{R2} \le 0$
  - $UF3: F3 - 500\,y_{F3} \le 0$ — periodo 3
  - $UR3: R3 - 500\,y_{R3} \le 0$

soluzione (da report) : F : produco 24 (scorta = 8)

R : produco 16 (scorta = 0)

F : " 24 (" 4)

R : " 16 (" 2)

F : " 24

R : " 16

es : flow shop scheduling problem

n lavori che devono essere lavorati in m macchine

s_{ji} : tempo di lavorazione del lavoro j nella macchina i

d_j : limite della consegna del lavoro j

da minimizzare il tempo di lavorazione

y_{jhi} = (var. bin.) stabilisce se il lavoro j è processato prima del lavoro h sulla macchina i

t_{ji} : tempo nel quale il lavoro j viene iniziato sulla macchina i

f. ob. : $\min (\max s_{jm} + t_{jm})$

$\min T$

tempo completamento finale

sa : $t_{jm} + s_{jm} \leq T \quad \forall j$

$t_{jm} + s_{jm} \leq d_j \quad \forall j, \forall m$ vincolo della consegna

$t_{ji} + s_{ji} \leq t_{hi} + M(1 - y_{jhi}) \quad \forall j, \forall i, \forall h$ vincolo ordine - tempo iniziale

$t_{hi} + s_{hi} \leq t_{ji} + M\, y_{jhi}$

$t_{ji} + s_{ji} \leq t_{j,i+1} \quad \forall j, \forall i \neq m$ vincolo ordine dello stesso lavoro nelle macchine

altri esempi di problemi :

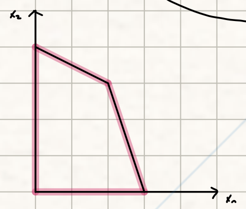
- p. di covering → coprire un'area completamente, anche con eccesso (p. delle telecamere)
- p. di packing → coprire un'area il più possibile senza sovrapposizione (p. della creazione orario treni)
- p. di partitioning → coprire interamente un'area senza sovrapposizione (p. di creazione equipaggio per decolli e atterraggi)

tecnica del rilassamento continuo

$z_I = \max c'x$ $\quad$ $z_C = \max c'x$

s.a. $Ax = b$ $\quad$ s.a. $Ax = b$

$x \in Z^+$ $\quad$ $x \geq 0$



passo da un problema intero a uno continuo, la regione ammissibile aumenta

se ho max trovo un limite superiore

" min " inferiore

ma posso essere molto distante dall'ottimo intero

quindi, altra tecnica: formulazione equivalente → creo un poliedro più piccolo che comprende tutti gli interi presenti nella regione ammissibile originale, e che abbia come punti estremi dei punti interi

quindi:
- se applico la formulazione equivalente ottenendo il poliedro più piccolo possibile (p. estremi = p. interi)
- se poi applico il rilassamento continuo
- ottimo continuo nel 2° poliedro ≡ ottimo intero nel 1° poliedro

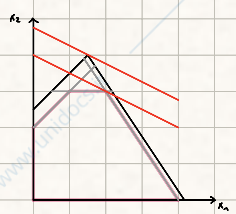
metodo dei piani di taglio

es: $\max x_1 + 2x_2$

s.a. $-2x_1 + 2x_2 \leq 5$

$6x_1 + 4x_2 \leq 25$

$x_1, x_2 \in Z^+$



$x_C = (1{,}5 ; 4)$

$x_I = (2 ; 3)$

nero → poliedro originale

grigio → poliedro dopo la formulazione equivalente

algoritmicamente: aggiungo un nuovo taglio finché la soluzione continua ottima non è intera

$x_C = (1{,}5 ; 4)$

$\max x_1 + 2x_2$

st $-2x_1 + 2x_2 + s_1 = 5$

$6x_1 + 4x_2 + s_2 = 25$

$x_1, x_2 \in Z^+ ; s_1, s_2 \geq 0$

$B = \begin{pmatrix} -2 & 2 \\ 6 & 4 \end{pmatrix} \rightarrow B^{-1} = \frac{1}{-20} \begin{pmatrix} 4 & -2 \\ -6 & -2 \end{pmatrix}$

$-8-12$

$$x_B + B^{-1} D x_D = B^{-1} b$$

$$x_B - \frac{1}{20}\begin{pmatrix} 4 & -2 \\ -6 & -2 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} x_D = -\frac{1}{20}\begin{pmatrix} 4 & -2 \\ -6 & -2 \end{pmatrix}\begin{pmatrix} 5 \\ 25 \end{pmatrix}$$

$$x_B - \frac{1}{20}\begin{pmatrix} 4 & -2 \\ -6 & -2 \end{pmatrix}\begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = -\frac{1}{20}\begin{pmatrix} -30 \\ -80 \end{pmatrix}$$

$$\begin{cases} x_1 - \frac{4}{20} s_1 + \frac{2}{20} s_2 = \frac{30}{20} \\ x_2 + \frac{6}{20} s_1 + \frac{2}{20} s_2 = \frac{80}{20} \end{cases}$$

$$\begin{cases} x_1 - \frac{1}{5} s_1 + \frac{1}{10} s_2 = \frac{3}{2} \\ x_2 + \frac{3}{10} s_1 + \frac{1}{10} s_2 = 4 \end{cases}$$

utilizzo x_1 perché è quello non intero

$$\left(-\frac{1}{5} - \left\lfloor -\frac{1}{5} \right\rfloor\right) s_1 + \left(\frac{1}{10} - \left\lfloor \frac{1}{10} \right\rfloor\right) s_2 \geq \frac{3}{2} - \left\lfloor \frac{3}{2} \right\rfloor \rightarrow \frac{4}{5} s_1 + \frac{1}{10} s_2 \geq \frac{1}{2}$$

$$8 s_1 + s_2 \geq 5$$

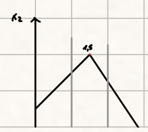
sapendo che: $\begin{cases} s_1 = 5 + 2x_1 - 2x_2 \\ s_2 = 25 - 6x_1 - 4x_2 \end{cases}$

ottengo nuovo vincolo: $-x_1 + 2x_2 \leq 6$

calcolo la nuova coppia (x_1, x_2) → se non è intera, rifaccio

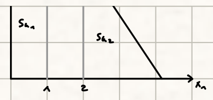
algoritmo branch and bound

→ divido il poliedro in più sotto-regioni



→ elimino la striscia centrale, ottengo 2 regioni ammissibili:

- $S_{k1} = \{x : x \in S_k,\ x_i \leq \lfloor f \rfloor\}$
- $S_{k2} = \{x : x \in S_k,\ x_i \geq \lfloor f \rfloor + 1\}$



P_0 : max $x_1 + 2x_2$ → sol: (1,5 ; 4) z = 9,5

s.a. $-2x_1 + 2x_2 \leq 5$

$6x_1 + 4x_2 \leq 25$

$x_1, x_2 \geq 0$

→ P_1 : max $x_1 + 2x_2$ → sol: (1; 3,5) z = 8 → ma più accettabile: ho già una sol intera con z = 8

s.a. $-2x_1 + 2x_2 \leq 5$

$6x_1 + 4x_2 \leq 25$

$x_1 \leq 1$

$x_1, x_2 \geq 0$

→ P_2 : max $x_1 + 2x_2$ → sol: $(2, \overset{\approx 3,..}{13/4})$ z = 8

applico ancora l'algoritmo su x_2

s.a. $-2x_1 + 2x_2 \leq 5$

$6x_1 + 4x_2 \leq 25$

$x_1 \geq 2$

$x_1, x_2 \geq 0$

→ P_3 : max $x_1 + 2x_2$ → sol: (13/6 ; 3) z = $8,1\bar{6}$

s.a. $-2x_1 + 2x_2 \leq 5$

$6x_1 + 4x_2 \leq 25$

$x_1 \geq 2$

$x_2 \leq 3$

$x_1, x_2 \geq 0$

→ P_5 : max $x_1 + 2x_2$ → sol: (2 , 3) z = 8 → stop (candidato)

s.a. $-2x_1 + 2x_2 \leq 5$

$6x_1 + 4x_2 \leq 25$

$x_1 = 2$

$x_2 \leq 3$

$x_1, x_2 \geq 0$

→ P_6 : max $x_1 + 2x_2$ → sol: (3, 7/4) z = 6,5 → stop (z < 8)

s.a. $-2x_1 + 2x_2 \leq 5$

$6x_1 + 4x_2 \leq 25$

$x_1 \geq 3$

$x_2 \leq 3$

$x_1, x_2 \geq 0$

→ P_4 : max $x_1 + 2x_2$ → inammissibile → stop

s.a. $-2x_1 + 2x_2 \leq 5$

$6x_1 + 4x_2 \leq 25$

$x_1 \geq 2$

$x_2 \geq 4$

$x_1, x_2 \geq 0$

NB: regole di Fathoming ока quando non devo fare branching

- se S_h é vuoto (= inammissibile)
- se $z \leq LB$ con $LB = z$ più alto con sol intera
- se x é intera

NB:
- inizializzazione : $LB = -\infty$, $UB = +\infty$
  $L = \{P_0\}$ da rilassamento del problema $\rightarrow$ L lista problemi da risolvere
- test di arresto : $L = \emptyset$
- scelta sotto problema : scelgo quello con $z >$

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari
www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari