Università degli studi di Padova

Facoltà di Ingegneria

Dipartimento di tecnica e gestione dei sistemi industriali

Tesi di Laurea in Ingegneria Gestionale

PRODUZIONE SU MACCHINE PARALLELE NON CORRELATE
ALGORITMI ESATTI PER LA MINIMIZZAZIONE DEI COSTI

Relatore: prof Giorgio Romanin-Jacur

Laureando: Luca Cortese

Matricola: 610462-IG

ANNO ACCADEMICO 2011/2012

INDICE

CAPITOLO 1
Introduzione

Negli ultimi anni l'inasprimento della concorrenza a livello globale obbliga sempre più le imprese ad ottimizzare l'uso delle proprie risorse aziendali al fine di massimizzare i profitti ed essere competitive nel mercato. Le risorse aziendali sono limitate perciò diventa necessario allocarle in modo tale che l'obbiettivo, solitamente il profitto, venga raggiunto ed ottimizzato. Grazie ad una buona schedulazione dei lavori è possibile, infatti, ridurre il tempo di lavorazione e, di conseguenza, i costi di produzione. In questo elaborato viene affrontato il seguente problema: la schedulazione di più lavori su macchine parallele non correlate (macchine che hanno tempi di processo varianti in modo del tutto arbitrario a seconda dell'articolo prodotto), ovvero nell'allocare sia le risorse (le macchine) che nello schedulare i lotti da produrre.

In altre parole si vuole individuare una sequenza di lavoro sulle singole risorse aziendali al fine di minimizzare il costo di lavorazione di un insieme di determinati articoli rispettando i tempi di consegna di ciascuno di essi. Il problema affrontato è un problema di allocazione di risorse con delle differenze rispetto a quelli considerati usualmente. Vi sono presenti, infatti, vincoli dovuti alle finestre di tempo, numeri variabili di articoli e di macchine, tempi e costi di set-up dipendenti dai 2 articoli e dalla macchina.

Partendo dal modello di Dondo e Cerdà (EJOR 2007, pp. 4-7) sui problemi di allocazione delle risorse con più magazzini, finestre di tempo e veicoli eterogenei, ho elaborato io stesso degli algoritmi in grado di calcolare la sequenza di produzione e fornire un dettagliato piano di lavoro. Questi permettono di calcolare l'istante in cui la lavorazione di un articolo deve iniziare ed individuano la macchina, tra quelle disponibili, che dovrà eseguirne la lavorazione. Rispettando tali indicazioni sarà possibile minimizzare il costo per concludere il progetto. Successivamente questi algoritmi sono stati implementati nel linguaggio di programmazione gams e testati su delle istanze di prova. Da tali test si sono potute valutare anche le prestazioni che gli algoritmi sviluppati garantiscono.

Nell'elaborato ho considerato i seguenti elementi che caratterizzano il problema affrontato:

-m macchine parallele non correlate, rappresentate con un cerchio

-n articoli da produrre, rappresentati con un rettangolo

-relativi tempi e costi di set-up e di produzione

CAPITOLO 2
Dati del Problema

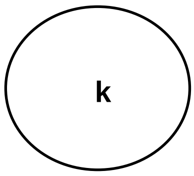
2.1 Dati relativi alle macchine

Ogni macchina k ha una propria finestra di lavoro [smk,fmk] da rispettare, cioè il momento in cui può essere avviata e il momento entro il quale deve essere spenta. Vi è inoltre un dato costo di accensione ck, che potrebbe essere, per esempio, il costo di un operaio addetto al controllo della macchina.

La macchina, inoltre, ha un proprio costo orario (ctk), che si applicherà al tempo totale in cui la macchina sarà accesa, e un certo costo sull'eventuale ritardo d'accensione (crk).

Possiamo riassumere i dati come segue:

[smk,fmk]



ck

2.2 Dati relativi agli articoli

Ogni articolo i, che presenta una quantità positiva p(i) da produrre, deve essere processato interamente da un'unica macchina k.

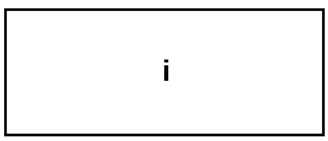
Tempo di lavorazione per articolo in min/pz (tp) e costo di lavorazione per articolo in €/pz (dp) sono noti per ogni coppia macchina-articolo.

Ogni articolo, inoltre, ha una propria finestra di lavorazione (si,fi), cioè il momento in cui si può iniziarne la produzione e il momento entro il quale deve essere terminata.

Possiamo riassumere i dati come segue:

[si,fi]

tp(i,k)



dp(i,k)

p(i)

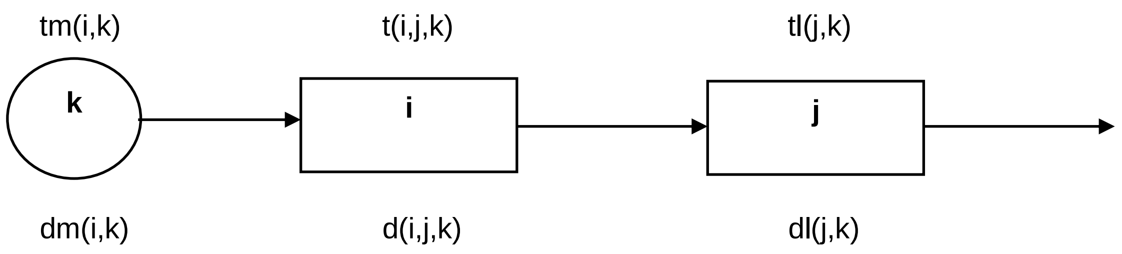
2.3 Dati relativi al set-up

Ogni articolo per essere processato su una macchina ha bisogno di un proprio set-up. Quando si passa dalla produzione di un dato articolo ad un altro, dunque, bisogna reimpostare la macchina in modo che sia configurata correttamente per la produzione del nuovo articolo. Queste operazioni comportano perdita di tempo e, di conseguenza, costi aggiuntivi.

Nel nostro problema tempi e costi di set-up sono dati nei seguenti casi:

- tm(i,k) e dm(i,k): tempo e costo di set-up per ogni coppia articolo-macchina nel caso in cui l'articolo i sia il primo ad essere lavorato dalla macchina k dopo l'accensione.
- t(i,j,k) e d(i,j,k): tempo e costo di set-up per ogni macchina k per passare dalla produzione dell'articolo i alla produzione dell'articolo j.
- tl(j,k) e dl(j,k): tempo e costo per resettare il set-up di j dalla macchina k nel caso in cui l'articolo j sia l'ultimo ad essere lavorato dalla macchina k prima dello spegnimento.

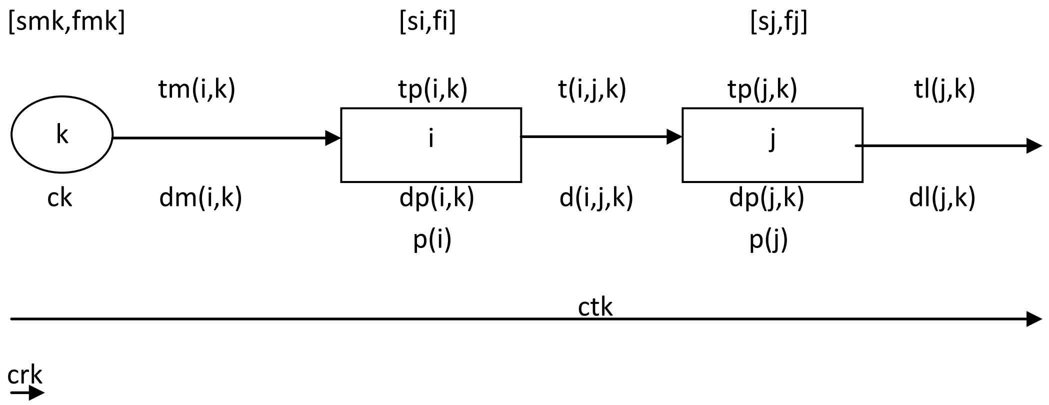
I dati possono essere riassunti come segue:



CAPITOLO 3
Modello, vincoli ed obiettivo del problema

3.1 Modello del problema

Rielaborando i dati di cui al cap.2, possiamo rappresentare il modello del problema qui affrontato come segue:
Per ogni macchina k



3.2 Obiettivo del problema

L’obiettivo del problema qui affrontato è quello di trovare degli algoritmi esatti tali da minimizzare il costo di produzione totale rispettando i vincoli imposti dalle finestre di tempo di ciascun articolo.
Con il modello adottato l’obiettivo del problema si può dunque riassumere come segue:
MINIMIZZARE
∑ { costi di accensione + costi totali di produzione + costi totali di set-up + costi del tempo lavorato + costi di ritardo d’accensione }

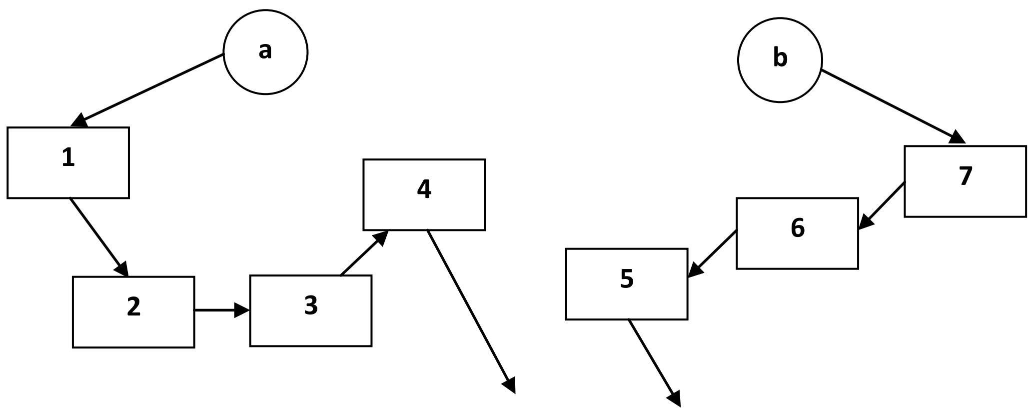
CAPITOLO 4
Variabili del problema

4.1 Variabili binarie

Nella soluzione del modello presentato, adattato da quello pubblicato da Dondo e Cerdà nell' European Journal of Operational Research (EJOR 2007, pp. 4-7), ripreso da Giorgio Romanin-Jacur e Carlo Filippi (Fresh Bovine Skin Transportation from Slaughters to Tannery with special time windows and capacity contraints, Euro 2012, Vilnius, p.3) in riferimento al trasporto di pellame, ho considerato le seguenti variabili binarie:

- ym(k), y(i,k) = 1 se, rispettivamente, la macchina k viene accesa e produce l'articolo i;
- ym(k), y(i,k) = 0 altrimenti;
- x(i,j) = 1 se l'articolo i precede l'articolo j nella produzione, con i < j;
- x(i,j) = 0 se l'articolo j precede l'articolo i nella produzione, con i < j;
- x(i,j) senza significato altrimenti;

In riferimento ad x e ad y, per esempio:



y(a) = y(b) = 1;

y(1,a) = y(2,a) = y(3,a) = y(4,a) = 1;

y(5,a) = y(6,a) = y(7,a) = 0;

y(5,b) = y(6,b) = y(7,b) = 1;

y(1,b) = y(2,b) = y(3,b) = y(4,b) = 0;

x(1,2) = x(1,3) = x(1,4) = x(2,3) = x(2,4) = x(3,4) = 1;

x(5,6) = x(5,7) = x(6,7) = 0;

In alternativa, come citato anche da Dondo e Cerdà, si potrebbero usare le seguenti variabili binarie:

- xm(i,k), x(i,j,k), xl(j,k), xml(k) = 1 se, rispettivamente, la macchina k viene accesa e produce l'articolo i, se dall'articolo i passa all'articolo j, se dopo la produzione dell'articolo j viene spenta o se non viene accesa;
- xm(i,k), x(i,j,k), xl(j,k), xml(k) = 0 altrimenti;

Si sconsiglia però l'uso di quest'ultima possibilità perché, oltre che a comportare un numero maggiore di variabili, opera correttamente solamente con piccole dimensioni (nel caso di Dondo e Cerdà con 8 macelli e 2 camion, contro i 13 macelli e 3 camion della prima possibilità).

4.2 Variabili non negative

Il modello utilizzato considera, inoltre, le seguenti variabili non negative:

- zm(k) e zl(k) rispettivamente tempo d'avvio della macchina k e fine della produzione (spegnimento di k);
- z(i) tempo fine produzione articolo i;
- cost(i) costo di produzione totale alla fine della lavorazione dell'articolo i;
- costl(k) costo totale di produzione della macchina k;

CAPITOLO 5

Equazioni del problema

Per risolvere il problema si è dovuto pensare a delle equazioni tali da riuscire a descrivere i vincoli in maniera corretta e tali da far “capire” al calcolatore il significato delle variabili x ed y.

Qui di seguito sono elencate e spiegate tutte le equazioni scritte nel programma Gams usato per la risoluzione del problema.

- MIN $\sum (costl(k) \mid k = 1,m)$ questa è l’equazione obiettivo del problema, descritto anche nel cap. 3.
- $\sum (((fm(k)-sm(k)-tm(i,k)-tl(i,k))/tp(i,k))*y(i,k) \mid k= 1,m) \geq p(i)$

  per ogni i

  in quest’equazione dico che la quantità di articoli prodotti non deve essere maggiore della quantità di articoli che la macchina k riuscirebbe a produrre se nel tempo a sua disposizione producesse solo quell’articolo
- $\sum (y(i,k) \mid k = 1,m) \leq 1$ per ogni i

  quest’equazione esprime il fatto che ciascun articolo i deve essere prodotto da un’unica macchina k
- $\sum (p(i)*tp(i,k)*y(i,k) \mid i = 1,n) \leq (fm(k)-sm(k))*ym(k)$ per ogni k

  in questa equazione dico, invece, che il tempo totale di lavorazione della macchina k, se avviata, non deve superare il tempo che ha a disposizione
- $z(i) \geq zm(k)+(tm(i,k)+tp(i,k)*p(i))*(ym(k)+y(i,k)-1)$ per ogni i
- $z(j) \geq z(i)+tp(j,k)*p(j)+t(i,j,k)-M*(1-x(i,j))-M*(2-y(i,k)-y(j,k))$ per ogni $i<j$
- $z(i) \geq z(j)+tp(i,k)*p(i)+t(j,i,k)-M*x(i,j)-M*(2-y(i,k)-y(j,k))$per ogni $i<j$

  queste ultime tre equazioni sono, in pratica, la definizione di z(i) e z(j)
- $zl(k) \geq z(i)+tl(i,k)-M*(1-y(i,k))$ per ogni k

  questa è la definizione di zl
- $zm(k) \geq sm(k)$ per ogni k
- $zm(k) \leq fm(k)$ per ogni k

  queste ultime due equazioni dicono che la macchina k deve essere accesa all’interno dell’intervallo di tempo a sua disposizione
- $z(i) \geq s(i)+tp(i,k)*p(i)-M*(1-y(i,k))$ per ogni i,k
- $z(i) \leq f(i)+M*(1-y(i,k))$ per ogni i,k

queste due equazioni dicono che il tempo di fine lavorazione di i deve essere compreso tra il momento in cui si può iniziarne la lavorazione più il suo tempo di produzione e il momento entro il quale deve essere ultimata la lavorazione dell'articolo

- zl(k) >= sm(k) per ogni k
- zl(k) <= fm(k) per ogni k

  queste due equazioni, invece, dicono che il tempo di spegnimento della macchina k deve essere all'interno della sua finestra di spegnimento

- cost(i) >= (c(k)+dm(i,k)+dp(i,k)*p(i))*(ym(k)+y(i,k)-1)   per ogni i,k
- cost(j) >= cost(i)+d(i,j,k)+dp(j,k)*p(j)-M*(1-x(i,j))-M*(2-y(i,k)-y(j,k)) per ogni i<j, per ogni k
- cost(i) >= cost(j)+d(j,i,k)+dp(i,k)*p(i)-M*x(i,j)-M*(2-y(i,k)-y(j,k)per ogni i<j, per ogni k

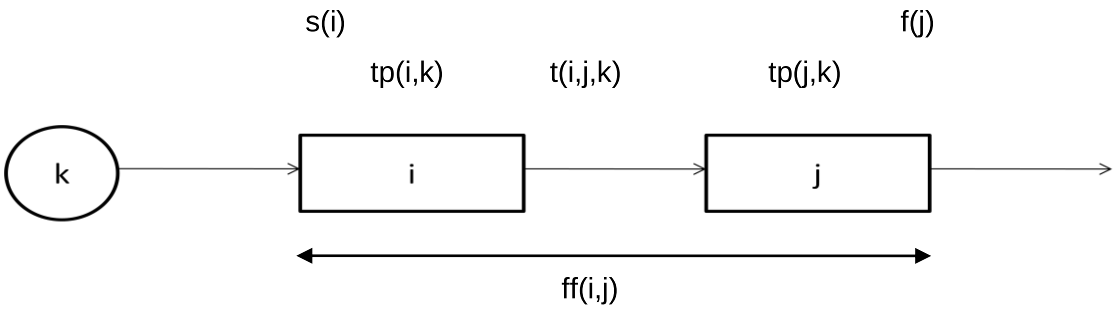
  queste tre ultime equazioni sono, in pratica, la definizione di cost(i) e cost(j)

- costl(k) >= cost(i)+dl(i,k)+ct(k)*(zl(k)-zm(k))+cr(k)*(zm(k)-sm(k))-M*(2-y(i,k)-ym(k))   per ogni i,k

  questa è la definizione di costl(k)

- ff(i,j) >= f(j)-s(i)   per ogni i,j
- y(i,k)+y(j,k)+x(i,j) <= 2+(ff(i,j)/(tp(i,k)*p(i)+tp(j,k)*p(j)+t(i,j,k)))per ogni i<j, per ogni k
- y(i,k)+y(j,k)-x(i,j) =l= 1+(ff(j,i)/(tp(i,k)*p(i)+tp(j,k)*p(j)+t(j,i,k)))per ogni i<j, per ogni k

  queste ultime tre equazioni, in pratica, affermano che la produzione degli articoli i e j (con j prodotto subito dopo i) deve avvenire all'interno della finestra di tempo [s(i),f(j)]. Si possono rappresentare come segue:



M nelle equazioni è un numero sufficientemente grande, tale da rendere negative alcune equazioni nel caso in cui x e y siano = 0.

CAPITOLO 6

Programma gams

In questo capitolo viene riportato il programma scritto nel linguaggio gams usato per la risoluzione del problema:

```
*prodotti, macchine
SETS i/i1*i11/,k/k1*k3/;ALIAS (i,j);
*finestra scarico in magazzino, big M
SCALAR M/5000/;
PARAMETERS
*finestra (start,finish) accensione,spegnimento macchinari
sm(k)/k1 10,k2 30,k3 20/,fm(k)/k1 1000,k2 1130,k3 1020/,
*finestra (start) possibile inizio produzione per prodotto
s(i)/i1 120,i2 380,i3 180,i4 450,i5 210,i6 500,i7 880,i8 590,i9 110,i10 720,i11 700/,
*finestra (finish) max fine lavorazione prodotto
f(i)/i1 540,i2 960,i3 490,i4 840,i5 900,i6 1500,i7 1400,i8 1000,i9 710,i10 1060,i11 930/,
*quantita' prodotto
p(i)/i1 25,i2 53,i3 80,i4 30,i5 39,i6 34,i7 42,i8 39,i9 48,i10 62,i11 44/,
*costo di accensione,
c(k)/k1 200,k2 220,k3 180/,
*costo del tempo impiegato e del ritardo per macchina
ct(k)/k1 0.1,k2 0.1,k3 0.1/,cr(k)/k1 0.05,k2 0.05,k3 0.05/;

*tempo di produzione per ogni macch. min/pz
TABLE tp(i,k)
      k1    k2    k3
i1    1     0.8   1.1
i2    0.9   1     1
i3    1.3   1.1   0.9
…
```

*costo di lavorazione per ogni macch. euro/pz

TABLE dp(i,k)

	k1	k2	k3
i1	2	2	2
i2	3	3	3
i3	3	4	4

…

*tempo "primo" set-up per ogni prodotto su ogni macchina

TABLE tm(i,k)

	k1	k2	k3
i1	125	130	132
i2	137	140	143
i3	151	157	160

…

*costo "primo" set-up per ogni prodotto su ogni macchina

TABLE dm(i,k)

	k1	k2	k3
i1	25	30	27
i2	37	40	38
i3	51	57	53

…

*tempo per togliere set-up di ogni prodotto da ogni macchina

TABLE tl(i,k)

	k1	k2	k3
i1	128	131	133
i2	139	142	144
i3	153	157	160

…

*costo per togliere set-up di ogni prodotto da ogni macchina

TABLE dl(i,k)

	k1	k2	k3
i1	28	31	29
i2	39	42	40
i3	53	57	55

…

```
*tempo di set-up da ogni prodotto ad ogni altro prodotto per ogni macchina
TABLE t(i,j,k)
        k1    k2    k3
i1.i2   145   147   146
i1.i3   161   166   163
i1.i4   156   159   157
…
*costo di set-up da ogni prodotto ad ogni altro prodotto per ogni macchina
TABLE d(i,j,k)
        k1    k2    k3
i1.i2   45    47    46
i1.i3   61    66    63
i1.i4   56    59    57
…

VARIABLES y(i,k),ym(k),x(i,j);BINARY VARIABLES y,ym,x;
VARIABLES z(i),zm(k),zl(k),costot,cost(i),costl(k),ff(i,j);
POSITIVE VARIABLES z,zm,zl,cost,costl,ff;
EQUATIONS costo,vmac1(i),vmac2(i),port(k),
temini(i,k),tem1(i,j,k),tem2(i,j,k),temfin(i,k),
finini1(k),finini2(k),fin1(i,k),fin2(i,k),finl1(k),finl2(k),
cini(i,k),c1(i,j,k),c2(i,j,k),cfin(i,k),
diff(i,j),imp1(i,j,k),imp2(i,j,k);
costo..costot =e= sum(k,costl(k));
vmac1(i)..sum(k,((fm(k)-sm(k)-tm(i,k)-tl(i,k))/tp(i,k))*y(i,k)) =g= p(i);
vmac2(i)..sum(k,y(i,k)) =l= 1;
port(k)..sum(i,p(i)*tp(i,k)*y(i,k)) =l= (fm(k)-sm(k))*ym(k);
temini(i,k)..z(i) =g= zm(k)+(tm(i,k)+tp(i,k)*p(i))*(ym(k)+y(i,k)-1);
tem1(i,j,k)$(ord(i)<ord(j))..z(j) =g= z(i)+tp(j,k)*p(j)+t(i,j,k)-M*(1-x(i,j))-
M*(2-y(i,k)-y(j,k));
tem2(i,j,k)$(ord(i)<ord(j))..z(i) =g= z(j)+tp(i,k)*p(i)+t(j,i,k)-M*x(i,j)-
M*(2-y(i,k)-y(j,k));
temfin(i,k)..zl(k) =g= z(i)+tl(i,k)-M*(1-y(i,k));
finini1(k)..zm(k) =g= sm(k);
finini2(k)..zm(k) =l= fm(k);
fin1(i,k)..z(i) =g= s(i)+tp(i,k)*p(i)-M*(1-y(i,k));
fin2(i,k)..z(i) =l= f(i)+M*(1-y(i,k));
```

```
finl1(k)..zl(k) =g= sm(k);
finl2(k)..zl(k) =l= fm(k);
cini(i,k)..cost(i) =g= (c(k)+dm(i,k)+dp(i,k)*p(i))*(ym(k)+y(i,k)-1);
c1(i,j,k)$(ord(i)<ord(j))..cost(j) =g= cost(i)+
        d(i,j,k)+dp(j,k)*p(j)-M*(1-x(i,j))-M*(2-y(i,k)-y(j,k));
c2(i,j,k)$(ord(i)<ord(j))..cost(i) =g= cost(j)+d(j,i,k)+
dp(i,k)*p(i)-M*x(i,j)-M*(2-y(i,k)-y(j,k));
cfin(i,k)..costl(k) =g= cost(i)+dl(i,k)+ct(k)*(zl(k)-zm(k))+
cr(k)*(zm(k)-sm(k))-M*(2-y(i,k)-ym(k));
diff(i,j)..ff(i,j) =g= f(j)-s(i);
imp1(i,j,k)$(ord(i)<ord(j))..
        y(i,k)+y(j,k)+x(i,j) =l= 2+(ff(i,j)/(tp(i,k)*p(i)+tp(j,k)*p(j)+t(i,j,k)));
imp2(i,j,k)$(ord(i)<ord(j))..
        y(i,k)+y(j,k)-x(i,j) =l= 1+(ff(j,i)/(tp(i,k)*p(i)+tp(j,k)*p(j)+t(j,i,k)));
MODEL prodotticosto/all/;OPTIONS mip=cplex,optcr=0.0;
SOLVE prodotticosto USING mipMINIMIZINGcostot;
DISPLAYy.l,ym.l,x.l,z.l,zm.l,zl.l,costot.l,cost.l,costl.l;
```

CAPITOLO 7
Risultati

Dopo l'implementazione del programma gams sono state effettuate varie prove variando il numero di articoli, variando M e variando i dati.
Il modello utilizzato opera correttamente con tre macchine e fino ad un massimo di undici articoli, senza sforare il limite massimo di risorse predefinito di gams. Il numero di articoli massimo dipende molto sia dai dati che dal valore di M.
Sono state fatte delle prove con M uguale a cinque mila e a cento mila.
Con dati ravvicinati tra loro si sono ottenuti i seguenti risultati:

Numero di iterazioni del calcolatore

Numero articoli	M = 5.000	M = 100.000
10 art.	1.000.000	3.000.000
11 art.	7.000.000	3.000.000

Con dati differenti tra loro si sono ottenuti i seguenti risultati:

Numero di iterazioni del calcolatore

Numero articoli	M = 5.000	M = 100.000
10 art.	500.000	300.000
11 art.	1.500.000	1.000.000

Per quanto riguarda i dati ravvicinati tra loro si può notare che, all'aumentare del numero di articoli, aumenta il numero di iterazioni solo nel caso di M uguale a cinque mila, invece nel caso di M uguale a cento mila il numero di iterazioni non varia.
Per quanto riguarda i dati differenti tra loro, invece, risulta evidente che con M grande il calcolatore risolve il problema più velocemente che con M piccolo, anche se di poco.

CAPITOLO 8
Conclusioni

I test effettuati, ricapitolando, hanno permesso di concludere che l'algoritmo sviluppato è in grado di ottenere soddisfacenti risultati per quanto riguarda la minimizzazione dei costi.

E' giusto far notare che il numero di iterazioni ottenute con questo algoritmo è più grande rispetto al modello Dondo & Cerdà, il che è dovuto al fatto che c'è un minore numero di vincoli da rispettare e quindi il calcolatore impiega più tempo a trovare la soluzione ottimale.

Per quanto riguarda il variare del numero di iterazioni al variare di M, si è notato che il tempo di elaborazione dei dati non ha un andamento fisso ma dipende molto dal numero di articoli e dai dati, perciò è difficile stabilire a priori quale sia il valore di M che più si adatta ad un certo tipo di problema. Nella maggior parte delle prove effettuate, però, l'utilizzo di un M grande comporta un minor numero di iterazioni e quindi un minor tempo di risoluzione del problema.

CAPITOLO 9

Bibliografia

[1] Dondo e Cerdà, EJOR 2007, pp. 4-7

[2] Giorgio Romanin-Jacur, Carlo Filippi, “Fresh Bovine Skin Transportation from Slaughters to Tannery with special time windows and capacity constraints”, Euro 2012 Vilnius, p.3

[3] Anthony Brook et al., GAMS- A USER’ S GUIDE, 1998.

[4] Gams Development Corporation, GAMS-The Solver Manuals, 1999.

CAPITOLO 10

Allegati

In questo capitolo sono allegate le informazioni più importanti riportate nei file .lst di gams, ossia le soluzioni dei problemi ed il numero di iterazioni effettuate dal calcolatore.

10.1 Esempio di SOLVE SUMMARY

```
S O L V E      S U M M A R Y

     MODEL   tenprodotti         OBJECTIVE  costot
     TYPE    MIP                 DIRECTION  MINIMIZE
     SOLVER  CPLEX               FROM LINE  375

**** SOLVER STATUS     1 NORMAL COMPLETION
**** MODEL STATUS      1 OPTIMAL
**** OBJECTIVE VALUE             2757.5200

 RESOURCE USAGE, LIMIT        680.880      1000.000
 ITERATION COUNT, LIMIT   7332637         10000

GAMS/Cplex    Jun 14, 2002 LNX.CP.CL 20.7 022.024.040.LNX For Cplex 7.5
Using environment variable ILOG_LICENSE_FILE to look for a Cplex license.
Cplex 7.5.0, GAMS Link 22

Proven optimal solution.

MIP Solution:         2757.520000    (7332601 iterations, 1701252 nodes)
Final LP:             2757.520000    (36 iterations)

Best integer solution possible:        2757.520000
Absolute gap:            0.000000
Relative gap:            0.000000
```

10.2 Risultati con 10 articoli e dati ravvicinati

```
----    332 VARIABLE y.L

             k1          k2
i1        1.000
i2                    1.000
i3                    1.000
i4        1.000
i5                    1.000
i6        1.000
i7                    1.000
i8                    1.000
i9        1.000
i10       1.000

----    332 VARIABLE ym.L

k1 1.000,    k2 1.000,    k3 1.000
```

```
----    332 VARIABLE x.L

            i4          i5          i6          i7          i8          i9         i10

i1       1.000                   1.000                   1.000       1.000       1.000
i2       1.000       1.000       1.000       1.000       1.000
i3       1.000       1.000       1.000       1.000       1.000
i4                                                                               1.000
i5                                           1.000
i6                                                                               1.000
i9                                                                               1.000


----    332 VARIABLE z.L

i1  206.400,    i2  479.000,    i3  275.000,    i4  698.800,    i5  814.700,    i6
540.800,    i7  957.700,    i8  646.900
i9  348.000,    i10 864.400

----    332 VARIABLE zm.L

k1 56.400,    k2 30.000,    k3 20.000

----    332 VARIABLE zl.L

k1  988.400,    k2 1119.700,    k3   20.000

        332 VARIABLE costot.L             =     2563.490

----    332 VARIABLE cost.L

i1   275.000,    i2   807.000,    i3   597.000,    i4   650.000,    i5  1202.000,    i6
529.000,    i7  1349.000,    i8  1019.000
i9   407.000,    i10  941.000

----    332 VARIABLE costl.L

k1 1062.520,    k2 1500.970
```

10.3 Risultati con 10 articoli e dati distanti

```
----    332 VARIABLE y.L

             k1          k2          k3
i1                    1.000
i2        1.000
i3        1.000
i4                                1.000
i5                                1.000
i6        1.000
i7                    1.000
i8                    1.000
i9        1.000
i10                               1.000

----    332 VARIABLE ym.L

k1 1.000,    k2 1.000,    k3 1.000

----    332 VARIABLE x.L
            i6          i7          i8         i10

i1                   1.000       1.000       1.000
i2       1.000       1.000       1.000
i3       1.000
i4                   1.000                   1.000
i5                                           1.000


----    332 VARIABLE z.L

i1  533.200,    i2  665.000,    i3  468.300,    i4  603.000,    i5  434.000,    i6
872.000,    i7  926.200,    i8  719.000
i9  219.300,    i10 775.800
```

```
----    332 VARIABLE zm.L

k1  83.700,    k2 219.300,    k3 219.300

----    332 VARIABLE zl.L

k1 1000.000,    k2 1088.200,    k3  901.800

        332 VARIABLE costot.L             =     2584.664

----    332 VARIABLE cost.L

i1  286.000,    i2  613.000,    i3  511.000,    i4  613.000,    i5  496.000,    i6
741.000,    i7  666.000,    i8  523.000
i9  229.000,    i10 717.000

----    332 VARIABLE costl.L

k1 865.263,    k2 842.586,    k3 876.815
```

10.4 Risultati con 11 articoli e dati ravvicinati

```
----    376 VARIABLE y.L

             k1          k2          k3
i1                    1.000
i2                                1.000
i3        1.000
i4                                1.000
i5                    1.000
i6        1.000
i7                    1.000
i8                    1.000
i9        1.000
i10                               1.000
i11                   1.000


----    376 VARIABLE ym.L

k1 1.000,    k2 1.000,    k3 1.000

----    376 VARIABLE x.L

             i4          i6          i7          i8          i9         i10         i11

i1                                1.000       1.000       1.000                   1.000
i2        1.000                               1.000                   1.000       1.000
i3                    1.000
i4                                                                    1.000
i5                                1.000       1.000                               1.000
i8                                                                                1.000

----    376 VARIABLE z.L

i1  451.000,    i2  433.000,    i3  398.600,    i4  630.000,    i5  274.000,    i6
580.400,    i7  962.900,    i8  632.900
i9  149.600,    i10 802.800,    i11 808.900

----    376 VARIABLE zm.L

k1  10.000,    k2  94.200,    k3 149.600

----    376 VARIABLE zl.L

k1  708.400,    k2 1124.900,    k3  928.800

        376 VARIABLE costot.L             =     2757.520

----    376 VARIABLE cost.L

i1  475.000,    i2  377.000,    i3  611.000,    i4  498.000,    i5  380.000,    i6
730.000,    i7  935.000,    i8  673.000
i9  329.000,    i10 726.000,    i11 812.000
```

```
----    376 VARIABLE costl.L

k1  835.840,    k2 1084.280,    k3  837.400
```

10.5 Risultati con 11 articoli e dati distanti

```
----    376 VARIABLE y.L

             k1          k2          k3
i1                    1.000
i2          1.000
i3          1.000
i4                                1.000
i5                                1.000
i6                    1.000
i7                    1.000
i8          1.000
i9          1.000
i10                               1.000
i11                   1.000

----    376 VARIABLE ym.L

k1 1.000,    k2 1.000,    k3 1.000

----    376 VARIABLE x.L

            i2          i4          i5          i6          i7          i8         i10        i11

i1       1.000       1.000                   1.000       1.000                               1.000
i2                   1.000       1.000                                           1.000
i3                                                                   1.000
i4                                                                               1.000
i5                                                                   1.000       1.000
i6                                                       1.000                               1.000

----    376 VARIABLE z.L

i1  421.400,    i2  861.000,    i3  453.300,    i4  603.000,    i5  434.000,    i6
579.000,    i7  926.200,    i8  673.300
i9  204.300,    i10 775.800,    i11 768.000

----    376 VARIABLE zm.L

k1  68.700,    k2 204.300,    k3 204.300

----    376 VARIABLE zl.L

k1 1000.000,    k2 1088.200,    k3  901.800

        376 VARIABLE costot.L              =     2714.964
----    376 VARIABLE cost.L

i1  286.000,    i2  807.000,    i3  511.000,    i4  613.000,    i5  496.000,    i6
438.000,    i7  714.000,    i8  709.000
i9  229.000,    i10 717.000,    i11 591.000

----    376 VARIABLE costl.L

k1 944.413,    k2 891.786,    k3 878.765
```