



# Strumenti software per la soluzione di problemi di PL e PLI

Fabrizio Marinelli  
[marinelli@di.univaq.it](mailto:marinelli@di.univaq.it)

Università degli Studi dell'Aquila

Pescara, 2 dicembre 2004



[www.oil.di.univaq.it](http://www.oil.di.univaq.it)

## Agenda

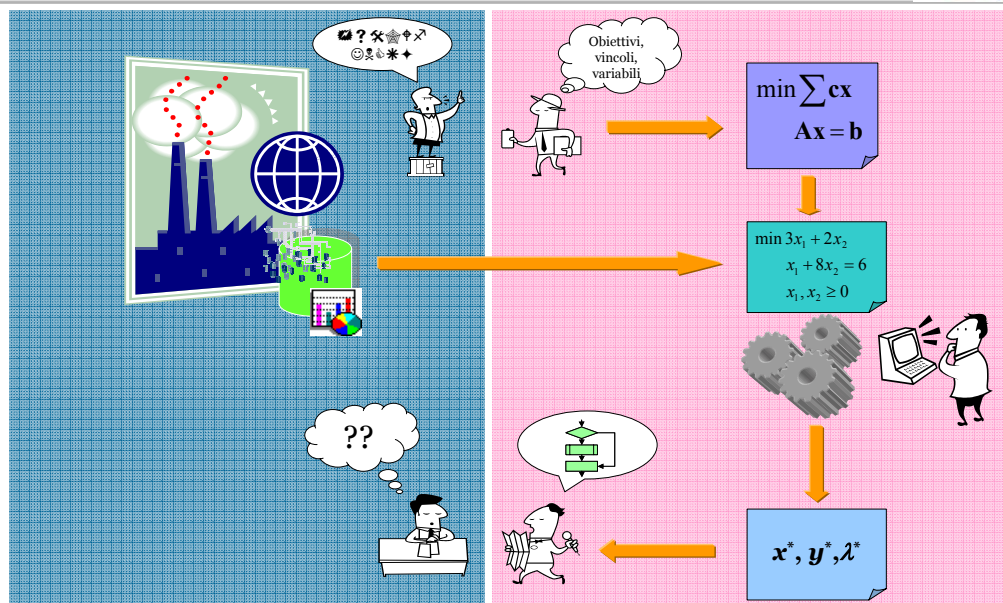


- ☐ **L'OTTIMIZZAZIONE NELLE APPLICAZIONI**
- ☐ **TECNICHE DI SOLUZIONE**
- ☐ **SOFTWARE DI OTTIMIZZAZIONE**
- ☐ **UN ESEMPIO CON EXCEL, CPLEX, LINGO**

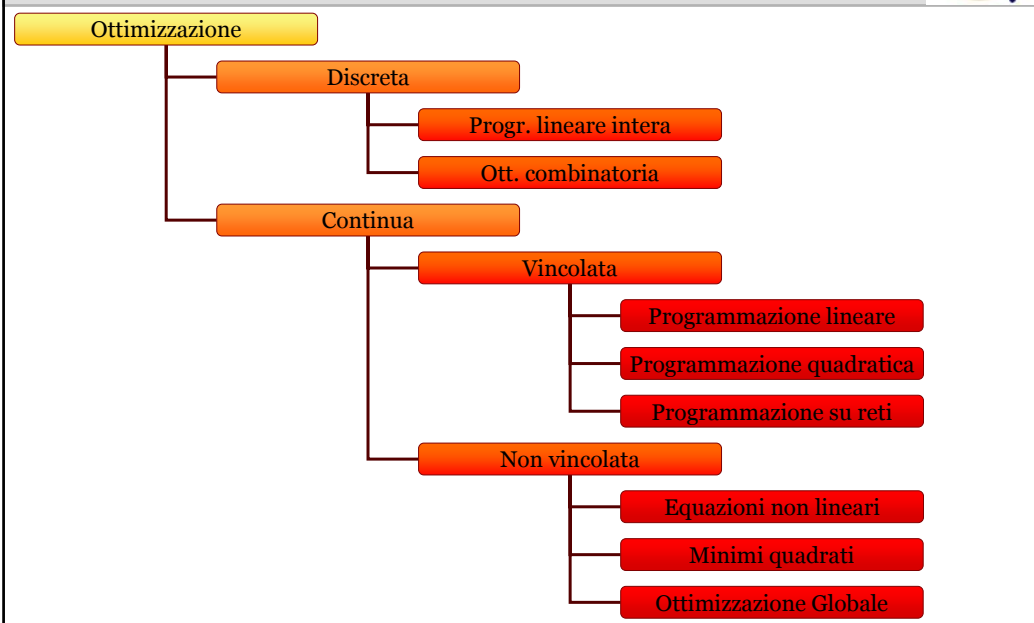


- ☐ L'OTTIMIZZAZIONE NELLE APPLICAZIONI
- ☐ TECNICHE DI SOLUZIONE
- ☐ SOFTWARE DI OTTIMIZZAZIONE
- ☐ UN ESEMPIO CON EXCEL, CPLEX, LINGO

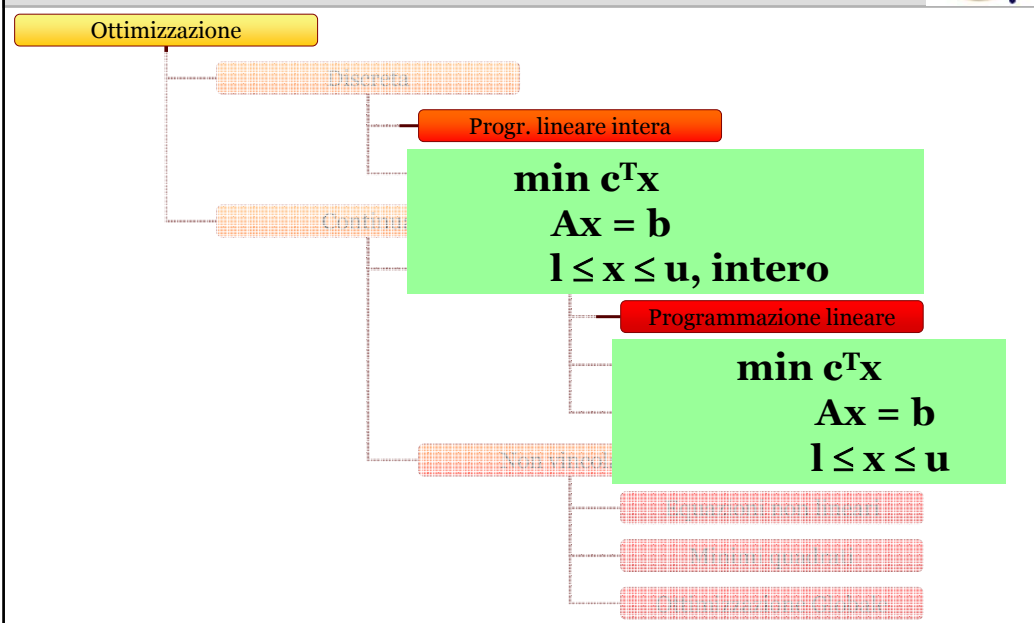
## L'ottimizzazione nelle applicazioni



# L'albero dell'ottimizzazione



# L'albero dell'ottimizzazione





☐ L'OTTIMIZZAZIONE NELLE APPLICAZIONI

☒ TECNICHE DI SOLUZIONE

☐ SOFTWARE DI OTTIMIZZAZIONE

☐ UN ESEMPIO CON EXCEL, CPLEX, LINGO



### **Algoritmi per la PL**

- Metodo del simplesso (**Dantzig, 1947**)
  - Simpleso duale, ridotto
  - Simpleso su reti
- Algoritmi interior-point (**Karmarkar, 1984**)

### **Algoritmi per la PLI**

- Cutting plane (**Gomory, 1957**)
- Branch-and-Bound (**Doig & Land, 1960**)
  - Branch-and-cut, Branch-and-Price



## Problemi di PL

- 48 vin. e 72 var. (1951)
- 1024 vin. (1963)
- 32.000 vin. (1973)

### *Attualmente*

- < 1.000 vin. prob. piccolo
- < 100.000 vin. prob. medio
- < 1.000.000 vin. prob. grande
- > 1.000.000 vin. stato dell'arte

## Problemi di PLI

### TSP (Travel Salesman Prob.)

- 1954 49 città
- 1971 64 città
- 1977 120 città
- 1987 2.392 città
- 1994 7.397 città
- 1998 13.509 città
- 2001 15.112 città
- 2004 24.978 città

## La PLI è veramente difficile



### **Modello applicativo:**

Problema in forma di massimo

44 vincoli, 51 variabili intere, 167 coefficienti non zero

### **Branch-and-Cut:**

Soluzione iniziale intera: -2586.0

Upper bound iniziale: -1379.4

...dopo 120,000 seconds, 370,000,000 nodi di B&C  
e 45 GB di memoria

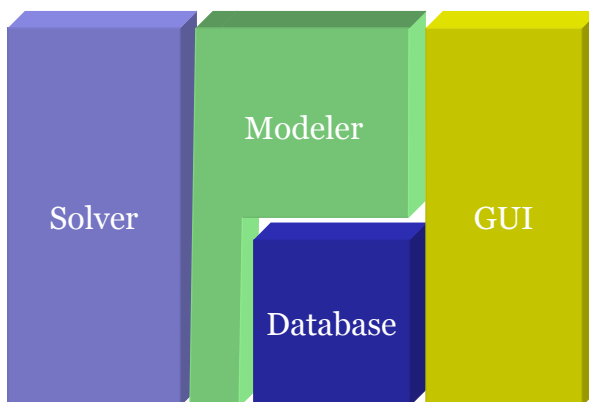
**Soluzione intera e bound INVARIATI**



- ☐ **L'OTTIMIZZAZIONE NELLE APPLICAZIONI**
- ☐ **TECNICHE DI SOLUZIONE**
- ☒ **SOFTWARE DI OTTIMIZZAZIONE**
- ☐ **UN ESEMPIO CON EXCEL, CPLEX, LINGO**

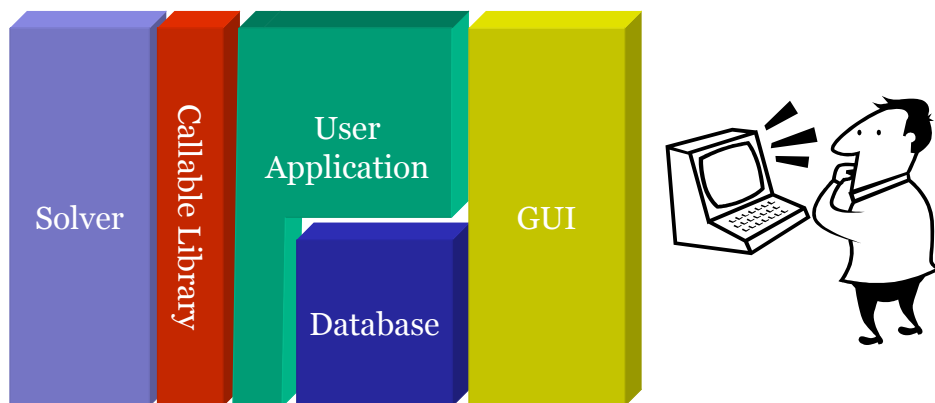


### **Ambienti integrati General Purpose** (OPL Studio, LINGO)

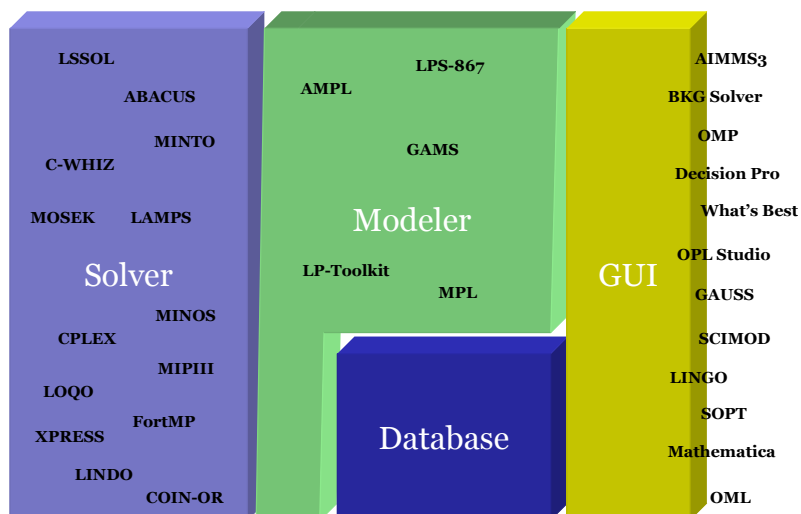




## Applicazioni verticalizzate



## Cosa offre il mercato



... ci sono più di 200 pacchetti software!



### **NEOS Guide to Optimization Software**

<http://www.mcs.anl.gov/home/otc/Guide/guide.html>

### **Software, Benchmark, Books, Tutorials**

<http://plato.la.asu.edu/guide.html>

### **Software for Optimization: A Buyer's Guide**

<http://www.ampl.com/solvers.html>

### **2003 linear programming software survey**

<http://lionhrtpub.com/orms/surveys/LP/LP-survey.html>

## Agenda



- ☐ L'OTTIMIZZAZIONE NELLE APPLICAZIONI
- ☐ TECNICHE DI SOLUZIONE
- ☐ SOFTWARE DI OTTIMIZZAZIONE
- ☐ UN ESEMPIO CON EXCEL, CPLEX, LINGO



# EXCEL, CPLEX e LINGO



		EXCEL	CPLEX	LINGO
<b>Problemi</b>	Variabili	continue, intere, binarie	continue, intere, binarie	continue, intere, binarie
	Vincoli e obiettivi	lineari, non lineari	lineari, quadratici	lineari, non lineari
<b>Algoritmi</b>	PL	Gradiente generalizzato	simpleso primale, simpleso duale, simpleso su reti, sifting, interior point	simpleso primale, simpleso duale, interior point
	PLI	Branch-and-bound	branch-and-cut	branch-and-cut
	Dimensione problemi	200 variabili	Spazio Disco	Spazio Disco
	piattaforme	Windows	Windows, Linus, Solaris, AIX, HP Alpha/Tru64, HP RISC/HP UX, IBM/RS6000, SGI/Irix,	Windows, Linus, Solaris, AIX
<b>Interfacce</b>	Read/write SpreadSheet	✓		✓
	Read/write Database	✓		✓
	Read/write File testo	✓	✓	✓
	Callable Library		✓	✓
	Object/Class Library	✓	✓	
	Add-in			Excel, Access
	Altri software		ILOG OPL Studio, AIMMS, AMPL, GAMS, MPL	
	Costo (Educational)		circa 1000 €	circa 1200 €

## Un esempio: il problema della dieta



- $n$  alimenti e  $m$  sostanze nutrienti
- $c_i$ : costo unitario dell'alimento  $i$
- $u_i$ : disponibilità dell'alimento  $i$
- $b_j$ : fabbisogno minimo della sostanza nutriente  $j$
- $a_{ij}$ : q.tà di sostanza nutriente  $j$  in una unità di alimento  $i$

### Problema:

determinare la dieta (il mix di alimenti) di costo minimo che garantisca il fabbisogno nutrizionale e non superi la disponibilità degli alimenti

## il problema della dieta: il modello



### Modello:

$x_i$  = quantità di alimento  $i$  inserito nella dieta

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

$$\sum_{i=1}^n a_{ij} x_i \geq b_j \quad \forall j = 1, \dots, m$$

$$0 \leq x_i \leq u_i \quad \forall i = 1, \dots, n$$

## Un esempio: il problema della dieta



### costi e disponibilità degli alimenti per unità di misura

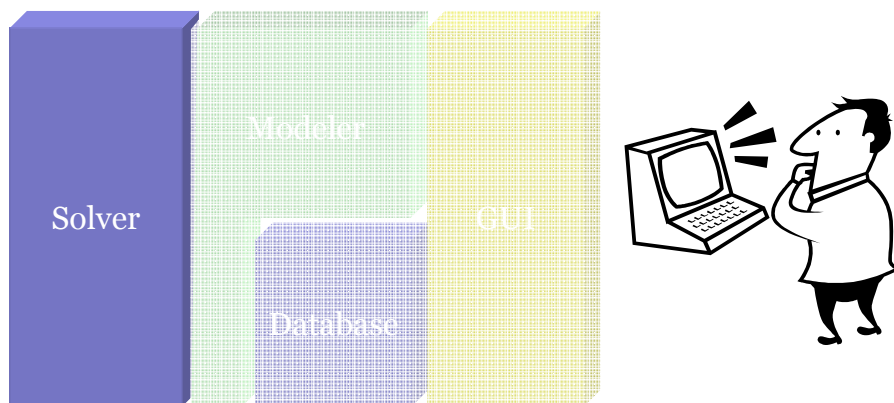
	Pane	Latte	Uova	Carne	Dolce
Costo	2	3	4	19	20
disponibilità	4	8	3	2	2

### sostanze nutritive per unità di alimento

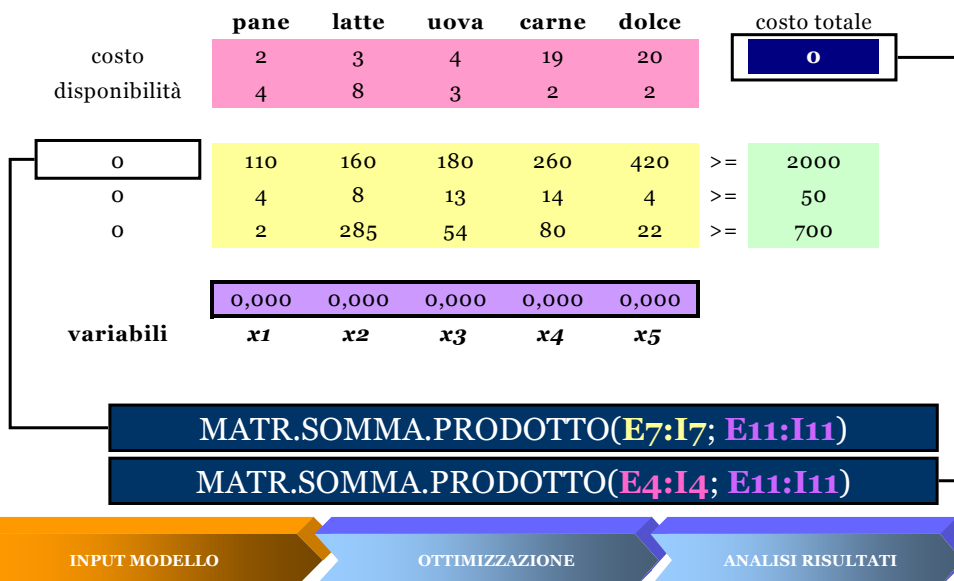
	Pane	Latte	Uova	Carne	Dolce
Calorie (cal.)	110	160	180	260	420
Proteine (g)	4	8	13	14	4
Calcio (mg)	2	285	54	80	22

### requisiti nutrizionali minimi

Calorie (cal.)	Proteine (g)	Calcio (mg)
2000	50	700



## Soluzione con Excel



## Soluzione con Excel



	pane	latte	uova	carne	dolce	costo totale
costo	2	3	4	19	20	0
disponibilità	4	8	3	2	2	

0	110	160	180	260	420	>=	2000
0	4	8	13	14	4	>=	50
0	2	285	54	80	22	>=	700

variabili	0,000	0,000	0,000	0,000	0,000
	x1	x2	x3	x4	x5

**MATR.SOMMA.PRODOTTO(E8:I8; E11:I11)**

INPUT MODELLO

OTTIMIZZAZIONE

ANALISI RISULTATI

## Soluzione con Excel



	pane	latte	uova	carne	dolce	costo totale
costo	2	3	4	19	20	0
disponibilità	4	8	3	2	2	

0	110	160	180	260	420	>=	2000
0	4	8	13	14	4	>=	50
0	2	285	54	80	22	>=	700

variabili	0,000	0,000	0,000	0,000	0,000
	x1	x2	x3	x4	x5

**MATR.SOMMA.PRODOTTO(E9:I9; E11:I11)**

INPUT MODELLO

OTTIMIZZAZIONE

ANALISI RISULTATI

## Soluzione con Excel



**Parametri del Risolutore**

Imposta cella obiettivo:

Uguale a: ☐ Max ☒ Min ☐ Valore di:

Cambiando le celle:

**Vincoli:**

- \$C\$7 >= \$K\$7
- \$C\$8 >= \$K\$8
- \$C\$9 >= \$K\$9
- \$E\$11 <= \$E\$5
- \$F\$11 <= \$F\$5
- \$G\$11 <= \$G\$5

Buttons: Risolvi, Chiudi, Ipotizza, Aggiungi, Cambia, Elimina, Opzioni, Reimposta, ?

INPUT MODELLO

OTTIMIZZAZIONE

ANALISI RISULTATI

## Soluzione con Excel



**Opzioni del Risolutore**

Tempo massimo:  secondi

Iterazioni:

Approssimazione:

Tolleranza:  %

Convergenza:

Buttons: OK, Annulla, Carica modello..., Salva modello..., ?

☒ Presupponi modello lineare ☐ Usa scala automatica

☒ Presupponi non negativo ☐ Mostra il risultato delle iterazioni

**Stima**

- ☒ Tangente
- ☐ Quadratica

**Derivate**

- ☒ Diretta
- ☐ Centrale

**Cerca**

- ☒ Newton
- ☐ Gradienti coniugati

INPUT MODELLO

OTTIMIZZAZIONE

ANALISI RISULTATI

## Soluzione con Excel



Cella obiettivo (Min)

Cella	Nome	Valori originali	Valore finale
\$K\$4	costo costo totale	0	38,2222222

Celle variabili

Cella	Nome	Valori originali	Valore finale
\$E\$11	pane	0,000	4,000
\$F\$11	latte	0,000	8,000
\$G\$11	uova	0,000	1,556
\$H\$11	carne	0,000	0,000
\$I\$11	dolce	0,000	0,000

Vincoli

Cella	Nome	Valore della cella	Formula	Stato	Tolleranza
\$C\$7	disponibilità	2000	\$C\$7>=\$K\$7	Vincolante	0
\$C\$8	disponibilità	100,2222222	\$C\$8>=\$K\$8	Non vincolante	50,2222222
\$C\$9	disponibilità	2372	\$C\$9>=\$K\$9	Non vincolante	1672
\$E\$11	pane	4,000	\$E\$11<=\$E\$5	Vincolante	0
\$F\$11	latte	8,000	\$F\$11<=\$F\$5	Vincolante	0
\$G\$11	uova	1,556	\$G\$11<=\$G\$5	Non vincolante	1,444444444
\$H\$11	carne	0,000	\$H\$11<=\$H\$5	Non vincolante	2
\$I\$11	dolce	0,000	\$I\$11<=\$I\$5	Non vincolante	2

INPUT MODELLO

OTTIMIZZAZIONE

ANALISI RISULTATI

## Soluzione con Excel



Cella	Obiettivo Nome	Valore
\$K\$4	costo costo totale	38,2222222

Cella	Variabile Nome	Valore
\$E\$11	pane	4,000
\$F\$11	latte	8,000
\$G\$11	uova	1,556
\$H\$11	carne	0,000
\$I\$11	dolce	0,000

Limite inferiore	Risultato obiettivo
4,000	38,222
8,000	38,222
1,556	38,222
0,000	38,222
0,000	38,222

Limite superiore	Risultato obiettivo
4,000	38,222
8,000	38,222
3,000	44,000
2,000	76,222
2,000	78,222

INPUT MODELLO

OTTIMIZZAZIONE

ANALISI RISULTATI

## Soluzione con Excel

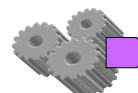


Celle variabili

Cella	Nome	Valore finale	ridotto Costo	oggettivo Coefficiente	consentito Incremento	consentito Decremento
\$E\$11	pane	4,000	-0,444	2	0,444444444	1E+30
\$F\$11	latte	8,000	-0,556	3	0,555555556	1E+30
\$G\$11	uova	1,556	0,000	4	4,571428571	0,625
\$H\$11	carne	0,000	13,222	19	1E+30	13,22222222
\$I\$11	dolce	0,000	10,667	20	1E+30	10,66666667

Vincoli

Cella	Nome	Valore finale	ombra Prezzo	Vincolo a destra	consentito Incremento	consentito Decremento
\$C\$7	disponibilità	2000	0,022222222	2000	260	280
\$C\$8	disponibilità	100,2222222	0	50	50,22222222	1E+30
\$C\$9	disponibilità	2372	0	700	1672	1E+30



INPUT MODELLO

OTTIMIZZAZIONE

ANALISI RISULTATI

## Soluzione con CPLEX



### Il formato LP

```

minimize           \oppure maximize
Obj:    2x1      +3x2    +4x3    +19x4    +20x5

subject to
c1:    110x1     +160x2  +180x3  +260x4  +420x5  >= 2000
c2:    4x1       +8x2    +13x3  +14x4    +4x5   >= 50
c3:    2x1       +285x2  +54x3  +80x4    +22x5  >= 700

bounds           \facoltativo: >= 0 di default
0 <= x1 <= 4
0 <= x2 <= 8
0 <= x3 <= 3
0 <= x4 <= 2
0 <= x5 <= 2

Generals         \oppure binary; facoltativi
x3

end
    
```

DOS

```

CPLEX>
CPLEX> read dieta.lp
    
```

INPUT MODELLO

OTTIMIZZAZIONE

ANALISI RISULTATI

# Soluzione con CPLEX



## DOS

```
CPLEX>  
CPLEX> mipopt
```

INPUT MODELLO

OTTIMIZZAZIONE

ANALISI RISULTATI

# Soluzione con CPLEX



## DOS

```
CPLEX>  
CPLEX> mipopt  
CPLEX> Tried aggregator 1 time.  
Reduced MIP has 3 rows, 5 columns, and 15 nonzeros.  
Presolve time = 0.00 sec.  
Root relaxation solution time = 0.02 sec.
```

Nodes		Objective	IInf	Best Integer	Cuts/	ItCnt	Gap
Node	Left				Best Node		
0	0	38.2222	1		38.2222	1	
*	0+	0	38.5000	0	38.5000	38.2222	1
							0.72%

```
Integer optimal solution: Objective = 3.85000000000e+001  
Solution time = 0.09 sec. Iterations = 1 Nodes = 0  
CPLEX>
```

INPUT MODELLO

OTTIMIZZAZIONE

ANALISI RISULTATI



## Soluzione con CPLEX



### DOS

```
CPLEX>
CPLEX> mipopt
CPLEX> Tried aggregator 1 time.
Reduced MIP has 3 rows, 5 columns, and 15 nonzeros.
Presolve time = 0.00 sec.
Root relaxation solution time = 0.02 sec.

      Nodes
      Node  Left   Objective  IInf  Best Integer    Cuts/
                                Best Node   ItCnt    Gap
*      0      0      38.2222    1      38.5000      38.2222     1
      0+     0      38.5000    0      38.5000      38.2222     1   0.72%

Integer optimal solution: Objective = 3.85000000000e+001
Solution time = 0.09 sec. Iterations = 1 Nodes = 0
CPLEX>

CPLEX> primopt
CPLEX> tranopt
CPLEX> netopt
```

INPUT MODELLO

OTTIMIZZAZIONE

ANALISI RISULTATI

## Soluzione con CPLEX



### DOS

```
CPLEX>
CPLEX> display solution var -
CPLEX> Variable Name      Solution Value
x1                        4.000000
x2                        8.000000
x3                        1.555556
All other variables in the range 1-5 are zero.
CPLEX>
```

INPUT MODELLO

OTTIMIZZAZIONE

ANALISI RISULTATI

## Soluzione con CPLEX



### DOS

```
CPLEX>
CPLEX> display solution var -
CPLEX> Variable Name      Solution Value
x1                          4.000000
x2                          8.000000
x3                          1.555556
All other variables in the range 1-5 are zero.
CPLEX> display solution dual -
Constraint Name            Dual Price
c1                          0.022222
All other dual prices in the range 1-3 are zero.
CPLEX>
```

INPUT MODELLO

OTTIMIZZAZIONE

ANALISI RISULTATI

## Soluzione con CPLEX



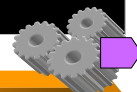
### DOS

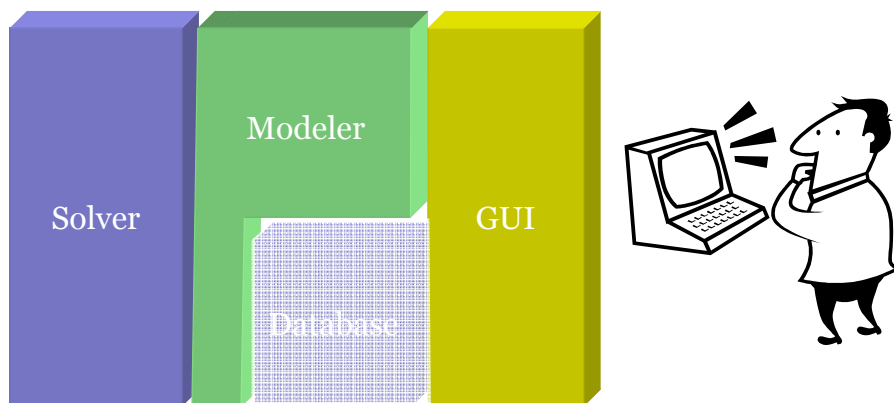
```
CPLEX>
CPLEX> display solution var -
CPLEX> Variable Name      Solution Value
x1                          4.000000
x2                          8.000000
x3                          1.555556
All other variables in the range 1-5 are zero.
CPLEX> display solution dual -
Constraint Name            Dual Price
c1                          0.022222
All other dual prices in the range 1-3 are zero.
CPLEX> display solution reduced -
Variable Name              Reduced Cost
x1                         -0.444444
x2                         -0.555556
x4                         13.222222
x5                         10.666667
All other reduced costs in the range 1-5 are zero.
CPLEX>
```

INPUT MODELLO

OTTIMIZZAZIONE

ANALISI RISULTATI





## Soluzione con LINGO



**LINGO** è un software dotato di un **linguaggio algebrico** per descrivere modelli complessi

- ad *alto livello* (comprensibile all'uomo)
- formalmente *strutturato* (interpretabile da un software)
- che realizza la *separazione* tra dati e modello in modo da:
  - poter cambiare i dati senza cambiare modello
  - importare i dati da diverse sorgenti (file di testo, spreadsheet, database, ...)

## Soluzione con LINGO



```
model:
title   Dieta;

min =   2*x1   +3*x2   +4*x3   +19*x4   +20*x5;

C1]    110*x1  +160*x2 +180*x3 +260*x4 +420*x5 >= 2000;
C2]    4*x1   +8*x2   +13*x3  +14*x4   +4*x5   >= 50;
C3]    2*x1   +285*x2 +54*x3   +80*x4   +22*x5  >= 700;

@bnd(0, x1, 4);           !bound sulle variabili;
@bnd(0, x2, 8);
@bnd(0, x3, 3);
@bnd(0, x4, 2);
@bnd(0, x5, 2);

@gin(x3);                 !vincolo di interezza;

end
```

INPUT MODELLO

OTTIMIZZAZIONE

ANALISI RISULTATI

## Soluzione con LINGO



LINGO Options

Nonlinear Solver | Integer Pre-Solver | Integer Solver  
Interface | General Solver | Linear Solver

Method: Solver Decides

Initial Linear Feasibility Tol: 3e-006

Final Linear Feasibility Tol: 1e-007

Model Reduction: Solver Decides

Pricing Strategies:

Primal Solver: Solver Decides

Dual Solver: Solver Decides

☐ Matrix Decomposition  
☒ Scale Model

Help Cancel Default Save Apply OK

INPUT MODELLO

OTTIMIZZAZIONE

ANALISI RISULTATI

## Soluzione con LINGO



**LINGO Solver Status [dieta]**

<b>Solver Status:</b> Model Class: LP State: Global Optimum Objective: 38.2222 Infeasibility: 0 Iterations: 2		<b>Variables:</b> Total: 5 Nonlinear: 0 Integers: 0
<b>Extended Solver Status:</b> Solver Type: . . . Best Obj: . . . Obj Bound: . . . Steps: . . . Active: . . .		<b>Constraints:</b> Total: 4 Nonlinear: 0  <b>Nonzeros:</b> Total: 20 Nonlinear: 0  <b>Generator Memory Used (K):</b> 6  <b>Elapsed Runtime (hh:mm:ss):</b> 00:00:00

Update Interval: 2    Interrupt Solver    Close

INPUT MODELLO

OTTIMIZZAZIONE

ANALISI RISULTATI

## Soluzione con LINGO



**LINGO - [Solution Report - dieta]**

File Edit LINGO Window Help

Global optimal solution found at iteration: 0  
Objective value: 38.2222

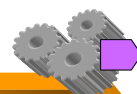
Variable	Value	Reduced Cost
X1	4.000000	-0.4444444
X2	8.000000	-0.5555556
X3	1.555556	0.000000
X4	0.000000	13.22222
X5	0.000000	10.66667

Row	Slack or Surplus	Dual Price
1	38.22222	-1.000000
C1	0.000000	-0.2222222E-01
C2	50.22222	0.000000
C3	1672.000	0.000000

INPUT MODELLO

OTTIMIZZAZIONE

ANALISI RISULTATI



## LINGO: il linguaggio di modellazione



### **MODEL:**

[**TITLE** <nome modello>;]

### **SETS:**

<definizione degli insieme di indici, delle variabili e delle costanti>

### **ENDSETS**

### **DATA:**

<assegnazioni valori alle costanti>

### **ENDDATA**

<definizione della funzione obiettivo e dei vincoli>

### **END**

## Esempio: il problema della dieta



### **MODEL:**

**TITLE** dieta;

### **SETS:**

ALIMENTI /PANE, LATTE, UOVA, CARNE, DOLCE/: COSTO, QTA, DISPONIBILITA;

SOSTANZE /CALORIE, PROTEINE, CALCIO/: MINIMO;

SOSTANZE\_ALIMENTI(SOSTANZE, ALIMENTI): COEFF;

### **ENDSETS**

## Esempio: il problema della dieta



**ALIMENTI /PANE, LATTE, UOVA, CARNE, DOLCE/: COSTO, QTA, DISPONIBILITA;**

	COSTO	QTA	DISPONIBILITA
PANE			
LATTE			
UOVA			
CARNE			
DOLCE			

**SOSTANZE /CALORIE, PROTEINE, CALCIO/: MINIMO;**

	MINIMO
CALORIE	
PROTEINE	
CALCIO	

**SOSTANZE\_ALIMENTI(SOSTANZE, ALIMENTI): COEFF;**

COEFF	PANE	LATTE	UOVA	CARNE	DOLCE
CALORIE					
PROTEINE					
CALCIO					

## Esempio: il problema della dieta



**MODEL:**

**TITLE** dieta;

**SETS:**

ALIMENTI /PANE, LATTE, UOVA, CARNE, DOLCE/: COSTO, QTA, DISPONIBILITA;

SOSTANZE /CALORIE, PROTEINE, CALCIO/: MINIMO;

SOSTANZE\_ALIMENTI(SOSTANZE, ALIMENTI): COEFF;

**ENDSETS**

**DATA:**

COSTO = 2, 3, 4, 19, 20;

DISPONIBILITA = 4, 8, 3, 2, 2;

MINIMO = 2000, 50, 700;

COEFF = 110, 160, 180, 260, 420,

4, 8, 13, 14, 4,

2, 285, 54, 80, 22;

**ENDDATA**

## Esempio: il problema della dieta



**COSTO** = 2, 3, 4, 19, 20;  
**DISPONIBILITA** = 4, 8, 3, 2, 2;

	COSTO	QTA	DISPONIBILITA
PANE	2		4
LATTE	3		8
UOVA	4		3
CARNE	19		2
DOLCE	20		2

**MINIMO** = 2000, 50, 700;

	MINIMO
CALORIE	2000
PROTEINE	50
CALCIO	700

**COEFF** = 110, 160, 180, 260, 420,  
 4, 8, 13, 14, 4,  
 2, 285, 54, 80, 22;

COEFF	PANE	LATTE	UOVA	CARNE	DOLCE
CALORIE	110	160	180	260	420
PROTEINE	4	8	13	14	4
CALCIO	2	285	54	80	22

## Esempio: il problema della dieta



**MODEL:**  
**TITLE** dieta;

**SETS:**  
 ALIMENTI /PANE, LATTE, UOVA, CARNE, DOLCE/: COSTO, QTA, DISPONIBILITA;  
 SOSTANZE /CALORIE, PROTEINE, CALCIO/: MINIMO;  
 SOSTANZE\_ALIMENTI(SOSTANZE, ALIMENTI): COEFF;

**ENDSETS**

**DATA:**

COSTO	= 2, 3, 4, 19, 20;
DISPONIBILITA	= 4, 8, 3, 2, 2;
MINIMO	= 2000, 50, 700;
COEFF	= 110, 160, 180, 260, 420, 4, 8, 13, 14, 4, 2, 285, 54, 80, 22;

**ENDDATA**

[OBJ] **min** = @sum(ALIMENTI(i): COSTO(i) \* QTA(i));

**@for**(SOSTANZE(j):  
 [RICHIESTA] @sum(ALIMENTI(i): COEFF(j,i) \* QTA(i)) >= MINIMO(j));

**@for**(ALIMENTI(i):  
 [BOUND] @bnd(o,QTA(i),DISPONIBILITA(i));

**END**



## Esempio: il problema della dieta



**[OBJ] min = @sum(ALIMENTI(i): COSTO(i) \* QTA(i));**

OBJ] MIN= 2 QTA( PANE) + 3 QTA( LATTE) + 4 QTA( UOVA) + 19 QTA( CARNE) + 20 QTA( DOLCE);

**@for(SOSTANZE(j):**

**[RICHIESTA] @sum(ALIMENTI(i): COEFF(j,i) \* QTA(i)) >= MINIMO(j));**

RICHIESTA( CALORIE)]

110 QTA( PANE) + 160 QTA( LATTE) + 180 QTA( UOVA) + 260 QTA( CARNE) + 420 QTA( DOLCE) >= 2000;

RICHIESTA( PROTEINE)]

4 QTA( PANE) + 8 QTA( LATTE) + 13 QTA( UOVA) + 14 QTA( CARNE) + 4 QTA( DOLCE) >= 50;

RICHIESTA( CALCIO)]

2 QTA( PANE) + 285 QTA( LATTE) + 54 QTA( UOVA) + 80 QTA( CARNE) + 22 QTA( DOLCE) >= 700;

**@for(ALIMENTI(i):**

**[BOUND] @bnd(o,QTA(i),DISPONIBILITA(i));**

o <= QTA( PANE) <= 4.000;

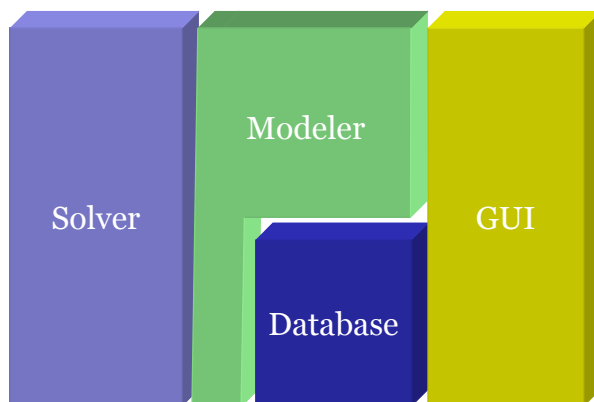
o <= QTA( LATTE) <= 8.000;

o <= QTA( UOVA) <= 3.000;

o <= QTA( CARNE) <= 2.000;

o <= QTA( DOLCE) <= 2.000;

## Interazione con i database



## Esempio: dati da excel



### MODEL:

**TITLE** dieta;

### SETS:

ALIMENTI /PANE, LATTE, UOVA, CARNE, DOLCE/: COSTO, QTA, DISPONIBILITA;  
SOSTANZE /CALORIE, PROTEINE, CALCIO/: MINIMO;  
SOSTANZE\_ALIMENTI(SOSTANZE, ALIMENTI): COEFF;

### ENDSETS

### DATA:

COSTO = @ole('dieta.xls','E4:I4');  
DISPONIBILITA = @ole('dieta.xls','E5:I5');  
MINIMO = @ole('dieta.xls','K7:K9');  
COEFF = @ole('dieta.xls','E7:I9');

@ole('dieta.xls','E11:I11') = QTA;

### ENDDATA

[OBJ] **min** = @sum(ALIMENTI(i): COSTO(i) \* QTA(i));

@for(SOSTANZE(j):

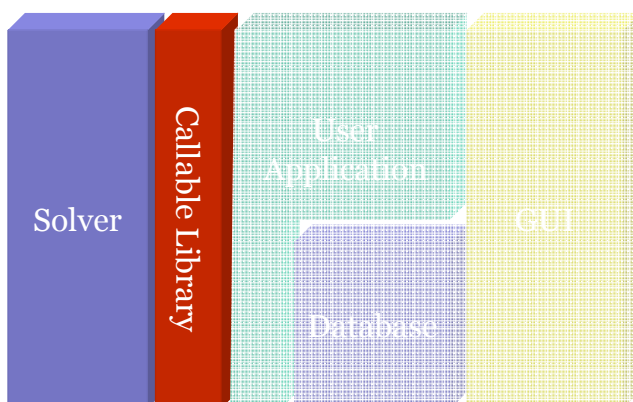
[RICHIESTA] @sum(ALIMENTI(i): COEFF(j,i) \* QTA(i)) >= MINIMO(j));

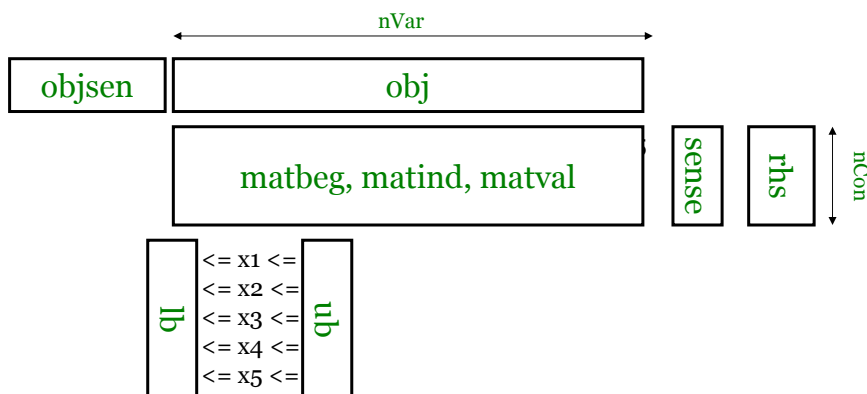
@for(ALIMENTI(i):

[BOUND] @bnd(0,QTA(i),DISPONIBILITA(i));

**END**

## La Callable Library





- **Inizializzazione CPLEX**

```
CPXENVptr env; CPXLPptr lp; int status;
env = CPXopenCPLEX(&status);
lp = CPXcreateprob(env, &status, probname);
```

- **Popolamento delle strutture dati** (obj, lb, ub, rhs, sense, rmatbeg, rmatind, rmatval)

- **Costruzione del problema per righe**

```
CPXchgobjsen(env, lp, CPX_MIN);
status = CPXnewcols(env, lp, nVar, obj, lb, ub, NULL, NULL);
status = CPXaddrows(env, lp, 0, nCon, nzcnt, rhs, sense,
    rmatbeg, rmatind, rmatval, NULL, NULL);
```

- **Soluzione del problema**

```
status = CPXlpopt(env, lp);
```

- **Acquisizione risultati**

```
status = CPXsolution(env, lp, &lpstat, &objval, x, pi, NULL, NULL);
```

- **Chiusura CPLEX**

```
CPXfreeprob(env, &lp);
CPXcloseCPLEX(&env);
```



**Micheal Trick's Operations Research Page**

<http://mat.gsia.cmu.edu/>

**Linear Programming Frequently Asked Questions**

<http://www-unix.mcs.anl.gov/otc/Guide/faq/linear-programming-faq.html>

**Mathematical Programming Glossary**

<http://carbon.cudenver.edu/~hgreenbe/glossary/index.php>

**Strumenti software per la soluzione di  
problemi di PL e PLI**

Fabrizio Marinelli  
[marinelli@di.univaq.it](mailto:marinelli@di.univaq.it)

Università degli Studi dell'Aquila

---

Pescara, 2 dicembre 2004



[www.oil.di.univaq.it](http://www.oil.di.univaq.it)