



SAPIENZA
UNIVERSITÀ DI ROMA

Programmazione lineare intera

Esercizi 3 — I solver MIP

Esercizi svolti sui capitoli 9–9, con la soluzione completa. Ogni esercizio cita il problema del catalogo che usa.

Fabio Furini

fabiofurini.github.io

Dipartimento di Ingegneria informatica, automatica e gestionale — Sapienza Università di Roma

Versione online e materiale scaricabile:

fabiofurini.github.io/programmazione-lineare-intera/
github.com/fabiofurini/programmazione-lineare-intera

Anno accademico 2026–2027

Indice

IV	I solver MIP	2
9	I solver MIP: che cosa fanno	3



Parte IV

I solver MIP



I solver MIP: che cosa fanno

Esercizio 9.1 — Presolve: rafforzamento dei bound

Consideriamo il seguente problema PLI con $n = 3$ variabili intere, ciascuna con i bound $0 \leq x_j \leq 10$:

$$\begin{aligned} \max \quad & 3x_1 + 2x_2 + 5x_3 \\ & 4x_1 + 3x_2 + 5x_3 \leq 20 \\ & -x_1 + x_2 \leq -1 \\ & -x_2 - x_3 \leq -2 \\ & x_1, x_2, x_3 \in \mathbb{Z}_{\geq 0} \end{aligned}$$

I vincoli $x_1 - x_2 \geq 1$ e $x_2 + x_3 \geq 2$ sono scritti in forma canonica come secondo e terzo vincolo.

Domande

1. Applicare il rafforzamento dei bound: per ogni vincolo $\sum_j a_{ij} x_j \leq b_i$ e ogni variabile x_k , usare $a_{ik} x_k \leq b_i - \sum_{j \neq k} \min\{a_{ij} x_j\}$ (minimo sui bound correnti), arrotondare agli interi e ripetere finché nessun bound cambia.
2. Verificare i nuovi bound per enumerazione. Sono i più stretti possibili?
3. Confrontare il valore ottimo del rilassamento lineare prima e dopo il rafforzamento dei bound.

Soluzione

1. Il minimo di $a_{ij} x_j$ è a_{ij} per il lower bound se $a_{ij} > 0$, e per l'upper bound se $a_{ij} < 0$. All'interno di ogni round i bound si aggiornano appena vengono trovati.

• Round 1.

- Vincolo 1: $4x_1 \leq 20 - (0) \Rightarrow x_1 \leq 5$, quindi $x_1 \leq 5$ (era 10).
- Vincolo 1: $3x_2 \leq 20 - (0) \Rightarrow x_2 \leq \frac{20}{3}$, quindi $x_2 \leq 6$ (era 10).
- Vincolo 1: $5x_3 \leq 20 - (0) \Rightarrow x_3 \leq 4$, quindi $x_3 \leq 4$ (era 10).

- Vincolo 2: $-x_1 \leq -1 - (0) \Rightarrow x_1 \geq 1$, quindi $x_1 \geq 1$ (era 0).
- Vincolo 2: $x_2 \leq -1 - (-5) \Rightarrow x_2 \leq 4$, quindi $x_2 \leq 4$ (era 6).

• **Round 2.**

- Vincolo 1: $5x_3 \leq 20 - (4) \Rightarrow x_3 \leq \frac{16}{5}$, quindi $x_3 \leq 3$ (era 4).

• **Round 3.** Nessun bound cambia: **STOP**.

I bound rafforzati sono

$$1 \leq x_1 \leq 5, \quad 0 \leq x_2 \leq 4, \quad 0 \leq x_3 \leq 3.$$

2. Il problema PLI ha 6 soluzioni ammissibili, e tutte soddisfano i nuovi bound: nessuna soluzione ammissibile va persa. Il valore più piccolo e quello più grande di ogni variabile sulle soluzioni ammissibili sono

$$1 \leq x_1 \leq 3, \quad 0 \leq x_2 \leq 2, \quad 0 \leq x_3 \leq 3.$$

I bound propagati sono validi, ma non sono tutti i più stretti possibili: per esempio $x_1 = 5$ richiede $x_2 = x_3 = 0$ per il primo vincolo, e questo viola $x_2 + x_3 \geq 2$. Il rafforzamento dei bound considera un vincolo alla volta.

3. Il rilassamento lineare con i bound originali ha valore ottimo 19, e con i bound rafforzati $\frac{75}{4}$. I dati sono interi, quindi l'upper bound passa da 19 a 18.

Il rafforzamento dei bound dà $1 \leq x_1 \leq 5, 0 \leq x_2 \leq 4, 0 \leq x_3 \leq 3$ senza perdere alcuna soluzione ammissibile; l'upper bound passa da 19 a 18, uguale a $z(\text{ILP})$.

Esercizio 9.2 — Presolve: rafforzamento dei coefficienti

Consideriamo il vincolo con 4 variabili binarie

$$6x_1 + 5x_2 + 4x_3 + 3x_4 \leq 10, \quad x_1, \dots, x_4 \in \{0, 1\},$$

e la funzione obiettivo $\max x_1 + x_2 + x_3 + x_4$.

Domande

1. Applicare il rafforzamento dei coefficienti ai coefficienti $a_1, \dots, a_4$, uno alla volta: il coefficiente a_j si può alzare a $b - M_j$, dove M_j è il valore più grande di $\sum_{k \neq j} a_k x_k$ sui punti binari con $x_j = 1$ che soddisfano il vincolo (corrente).
2. Verificare che i due vincoli abbiano le stesse soluzioni binarie.
3. Confrontare i valori ottimi dei rilassamenti lineari ($0 \leq x_j \leq 1$) dei due vincoli con il valore ottimo del problema binario.

Soluzione

1. Alzare a_j a $b - M_j$ conserva ogni soluzione binaria con $x_j = 1$ (il suo primo membro è al più $b - M_j + M_j = b$) e non cambia i punti con $x_j = 0$.

- a_1 : con $x_1 = 1$ il punto migliore è $(1, 0, 1, 0)$, quindi $M_1 = 4$ e $b - M_1 = 6$: $a_1 = 6$ non cambia.
- a_2 : con $x_2 = 1$ il punto migliore è $(0, 1, 1, 0)$, quindi $M_2 = 4$ e a_2 sale da 5 a $10 - 4 = 6$.
- a_3 : con $x_3 = 1$ il punto migliore è $(1, 0, 1, 0)$, quindi $M_3 = 6$ e $b - M_3 = 4$: $a_3 = 4$ non cambia.
- a_4 : con $x_4 = 1$ il punto migliore è $(1, 0, 0, 1)$, quindi $M_4 = 6$ e a_4 sale da 3 a $10 - 6 = 4$.

Il vincolo rafforzato è

$$6x_1 + 6x_2 + 4x_3 + 4x_4 \leq 10.$$

2. Entrambi i vincoli sono soddisfatti dagli stessi 10 punti binari (sui $2^4 = 16$), cioè $(0, 0, 0, 0)$, $(0, 0, 0, 1)$, $(0, 0, 1, 0)$, $(0, 0, 1, 1)$, $(0, 1, 0, 0)$, $(0, 1, 0, 1)$, $(0, 1, 1, 0)$, $(1, 0, 0, 0)$, $(1, 0, 0, 1)$, $(1, 0, 1, 0)$.
3. Il rilassamento lineare del vincolo originale ha soluzione ottima $(0, \frac{3}{5}, 1, 1)$ con valore $\frac{13}{5}$; il rilassamento lineare del vincolo rafforzato ha soluzione ottima $(\frac{1}{3}, 0, 1, 1)$ con valore $\frac{7}{3}$. Il problema binario ha valore ottimo 2.

Il rafforzamento dei coefficienti trasforma $6x_1 + 5x_2 + 4x_3 + 3x_4 \leq 10$ in $6x_1 + 6x_2 + 4x_3 + 4x_4 \leq 10$: stesse soluzioni binarie, stessa dimensione, e il bound del rilassamento lineare migliora da $\frac{13}{5}$ a $\frac{7}{3}$ (il valore ottimo è 2).

Esercizio 9.3 — Lettura del log di un solver

Gurobi risolve un problema PLI generico con $n = 10$ variabili intere $x_j \in \{0, \dots, 3\}$ e $m = 3$ vincoli, $\max \mathbf{c}'\mathbf{x}$ soggetto a $\mathbf{A}\mathbf{x} \leq \mathbf{b}$, con dati interi pseudo-casuali (Python `random.Random(4)`): a_{ij} uniforme in $\{1, \dots, 20\}$, c_j uniforme in $\{5, \dots, 25\}$, e $b_i = \lfloor 3 \sum_j a_{ij} / 3 \rfloor$, con `Presolve = 0`, `Cuts = 0` e `Heuristics = 0` (`Threads = 1`, `Seed = 0`). Il log è:

Log di Gurobi: problema PLI generico, presolve, tagli ed euristiche disattivati

```
01| Gurobi Optimizer version 13.0.3 build v13.0.3rc0 (linux64 - "Ubuntu 26.04 LTS")
02|
03| CPU model: Intel(R) Core(TM) i5-14500T, instruction set [SSE2|AVX|AVX2]
04| Thread count: 20 physical cores, 20 logical processors, using up to 1 threads
05|
06| Non-default parameters:
07| Heuristics 0
08| Cuts 0
09| LogToConsole 0
10| Presolve 0
11| Threads 1
12|
13| Optimize a model with 3 rows, 10 columns and 30 nonzeros (Max)
14| Model fingerprint: 0x92726b84
15| Model has 10 linear objective coefficients
16| Variable types: 0 continuous, 10 integer (0 binary)
17| Coefficient statistics:
18|   Matrix range    [1e+00, 2e+01]
19|   Objective range [7e+00, 2e+01]
20|   Bounds range    [3e+00, 3e+00]
21|   RHS range       [8e+01, 1e+02]
22|
23| Variable types: 0 continuous, 10 integer (0 binary)
24|
25| Root relaxation: objective 2.296103e+02, 4 iterations, 0.00 seconds (0.00 work units)
26|
27|      Nodes |      Current Node |      Objective Bounds |      Work
28|  Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time
29|
30|      0      0 229.61033  0   3          - 229.61033      -   -   0s
31|      0      0 229.30837  0   3          - 229.30837      -   -   0s
32|      0      2 229.30837  0   3          - 229.30837      -   -   0s
33| * 12      6          9 220.0000000 224.56863  2.08%  1.2   0s
34| * 22      3          6 221.0000000 224.53846  1.60%  1.0   0s
35| * 23      1          5 223.0000000 224.53846  0.69%  1.0   0s
36|
37| Explored 25 nodes (30 simplex iterations) in 0.00 seconds (0.00 work units)
38| Thread count was 1 (of 20 available processors)
39|
40| Solution count 3: 223 221 220
41|
42| Optimal solution found (tolerance 1.00e-04)
43| Best objective 2.230000000000e+02, best bound 2.230000000000e+02, gap 0.0000%
```

Domande

1. Il problema è di massimizzazione o di minimizzazione? Quante variabili e quanti vincoli ha?

2. Quale riga dà il bound del rilassamento lineare del nodo radice? Quale valore di UB fornisce?
3. Quali righe riportano nuove soluzioni incumbent, e come vengono trovate? Indicare il lower bound LB dopo ciascuna di esse.
4. Ricalcolare il gap della riga 33.
5. Quanti nodi vengono esplorati? La soluzione finale è certificata ottima?

Soluzione

1. Riga 13: (Max), quindi è un problema di **massimizzazione**, con 3 righe (vincoli) e 10 colonne (variabili), tutte intere e non binarie.
2. Riga 25: il rilassamento alla radice ha valore ottimo 229.61033 (esattamente $z(LP) = \frac{48907}{213}$). I dati sono interi, quindi $UB = \lfloor \frac{48907}{213} \rfloor = 229$. La colonna **BestBd** della riga 30 riporta lo stesso valore; a quel punto non esiste alcun incumbent (il trattino in **Incumbent**).
3.
 - Riga 33 (nodo 12): $LB = 220$.
 - Riga 34 (nodo 22): $LB = 221$.
 - Riga 35 (nodo 23): $LB = 223$.

Le tre righe iniziano con *: ogni soluzione incumbent è la soluzione ottima intera del problema LP di un nodo (potatura per integralità); le euristiche sono disattivate, quindi nessuna riga inizia con H.

4. Nella riga 33 abbiamo $Incumbent = 220$ e $BestBd = 224.569$, quindi

$$gap = \frac{|224.569 - 220|}{220} \approx 2.08\%,$$

come nella colonna **Gap** (2.08%).

5. Riga 37: vengono esplorati 25 nodi, con 30 iterazioni del simplesso. Riga 42: la soluzione di valore 223 è ottima con la tolleranza predefinita 10^{-4} ; l'ultima riga riporta best bound 223 e gap 0.

LB cresce da 220 a 223 con soluzioni intere dei nodi, mentre UB scende da 229 a 223 con il branching: dopo 25 nodi $LB = UB = 223$ e la soluzione è **certificata ottima**.

Esercizio 9.4 — Lettura del log di un solver: un problema di minimizzazione

Gurobi risolve un problema PLI generico con $n = 12$ variabili intere $x_j \in \{0, \dots, 3\}$ e $m = 4$ vincoli, $\min \mathbf{c}'\mathbf{x}$ soggetto a $\mathbf{A}\mathbf{x} \geq \mathbf{b}$, con dati interi pseudo-casuali (Python `random.Random(3)`): a_{ij} uniforme in $\{1, \dots, 20\}$, c_j uniforme in $\{5, \dots, 25\}$, e $b_i = \lfloor \frac{2}{5} \cdot 3 \sum_j a_{ij}/3 \rfloor$, con i parametri predefiniti (`Threads = 1`, `Seed = 0`). Il log è:

Log di Gurobi: problema di minimizzazione generico, parametri predefiniti

```

01| Gurobi Optimizer version 13.0.3 build v13.0.3rc0 (linux64 - "Ubuntu 26.04 LTS")
02|
03| CPU model: Intel(R) Core(TM) i5-14500T, instruction set [SSE2|AVX|AVX2]
04| Thread count: 20 physical cores, 20 logical processors, using up to 1 threads
05|
06| Non-default parameters:
07| LogToConsole 0
08| Threads 1
09|
10| Optimize a model with 4 rows, 12 columns and 48 nonzeros (Min)
11| Model fingerprint: 0x33d6ab84
12| Model has 12 linear objective coefficients
13| Variable types: 0 continuous, 12 integer (0 binary)
14| Coefficient statistics:
15|   Matrix range    [1e+00, 2e+01]
16|   Objective range [1e+01, 2e+01]
17|   Bounds range    [3e+00, 3e+00]
18|   RHS range       [5e+01, 6e+01]
19|
20| Found heuristic solution: objective 161.0000000
21| Presolve time: 0.00s
22| Presolved: 4 rows, 12 columns, 48 nonzeros
23| Variable types: 0 continuous, 12 integer (0 binary)
24|
25| Root relaxation: objective 6.192991e+01, 5 iterations, 0.00 seconds (0.00 work units)
26|
27|      Nodes      |      Current Node      |      Objective Bounds      |      Work
28|  Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd    Gap | It/Node Time
29|
30|      0       0   61.92991    0    3  161.00000   61.92991   61.5%    -    0s
31| H      0       0               90.0000000   61.92991   31.2%    -    0s
32| H      0       0               83.0000000   61.92991   25.4%    -    0s
33| H      0       0               67.0000000   61.92991    7.57%    -    0s
34|      0       0      cutoff    0          67.00000   67.00000    0.00%    -    0s
35|
36| Cutting planes:
37|   Gomory: 1
38|
39| Explored 1 nodes (9 simplex iterations) in 0.00 seconds (0.00 work units)
40| Thread count was 1 (of 20 available processors)
41|
42| Solution count 4: 67 83 90 161
43|
44| Optimal solution found (tolerance 1.00e-04)
45| Best objective 6.7000000000000e+01, best bound 6.7000000000000e+01, gap 0.0000%
```

Domande

1. In questo problema di minimizzazione, quale colonna dà l'upper bound e quale il lower bound?

2. Che cosa succede nelle righe 20, 22 e 25? Indicare i bound dopo la riga 25, usando il fatto che i dati sono interi.
3. Ricalcolare il gap della riga 33.
4. Spiegare le righe 34 e 37. Perché il problema è risolto al nodo radice?

Soluzione

1. In un problema di minimizzazione i ruoli si scambiano: **Incumbent** è il valore della migliore soluzione ammissibile, un **upper bound** UB sul valore ottimo, e **BestBd** è un **lower bound** LB.
2.
 - Riga 20: un'euristica trova una soluzione ammissibile di valore 161 prima del presolve, quindi $UB = 161$.
 - Riga 22: il presolve non riduce il modello (4 righe, 12 colonne).
 - Riga 25: il rilassamento lineare del nodo radice ha valore 61.92991 (esattamente $\frac{13253}{214}$). La funzione obiettivo assume valori interi, quindi $LB = \lceil \frac{13253}{214} \rceil = 62$, e $62 \leq z(ILP) \leq 161$.
3. Nella riga 33, $Incumbent = 67$ e $BestBd = 61.9299$:

$$gap = \frac{|61.9299 - 67|}{67} \approx 7.57\%,$$

come nella colonna **Gap**. Il gap è diviso per il valore dell'incumbent, come nella definizione del gap di ottimalità.

4. Le righe che iniziano con H sono nuove soluzioni incumbent trovate dalle euristiche alla radice, 90, 83 e 67. Nella riga 34 il nodo radice viene chiuso (**cutoff**): dopo il ciclo dei tagli il best bound raggiunge 67, il valore dell'incumbent. Riga 37: viene aggiunto un taglio di Gomory.

Le euristiche abbassano UB da 161 a 67, il rilassamento lineare e un taglio di Gomory alzano LB da 62 a 67: $LB = UB = 67$ alla radice, e il solver esplora un solo nodo.

Esercizio 9.5 — Il gap come criterio di arresto

Gurobi risolve il problema PLI generico dell'esercizio «Lettura del log di un solver» (un problema PLI generico con $n = 10$ variabili intere $x_j \in \{0, \dots, 3\}$ e $m = 3$ vincoli, $\max \mathbf{c}'\mathbf{x}$ soggetto a $\mathbf{A}\mathbf{x} \leq \mathbf{b}$, con dati interi pseudo-casuali (Python `random.Random(4)`): a_{ij} uniforme in $\{1, \dots, 20\}$, c_j uniforme in $\{5, \dots, 25\}$, e $b_i = \lfloor 3 \sum_j a_{ij}/3 \rfloor$) con gli stessi parametri e, in più, $\text{MIPGap} = 0.02$. Il log è:

Log di Gurobi: la stessa istanza con $\text{MIPGap}=0.02$

```

01| Gurobi Optimizer version 13.0.3 build v13.0.3rc0 (linux64 - "Ubuntu 26.04 LTS")
02|
03| CPU model: Intel(R) Core(TM) i5-14500T, instruction set [SSE2|AVX|AVX2]
04| Thread count: 20 physical cores, 20 logical processors, using up to 1 threads
05|
06| Non-default parameters:
07| MIPGap 0.02
08| Heuristics 0
09| Cuts 0
10| LogToConsole 0
11| Presolve 0
12| Threads 1
13|
14| Optimize a model with 3 rows, 10 columns and 30 nonzeros (Max)
15| Model fingerprint: 0x92726b84
16| Model has 10 linear objective coefficients
17| Variable types: 0 continuous, 10 integer (0 binary)
18| Coefficient statistics:
19|   Matrix range    [1e+00, 2e+01]
20|   Objective range [7e+00, 2e+01]
21|   Bounds range    [3e+00, 3e+00]
22|   RHS range       [8e+01, 1e+02]
23|
24| Variable types: 0 continuous, 10 integer (0 binary)
25|
26| Root relaxation: objective 2.296103e+02, 4 iterations, 0.00 seconds (0.00 work units)
27|
28|      Nodes      |      Current Node      |      Objective Bounds      |      Work
29|  Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time
30|
31|    0     0  229.61033    0   3         -  229.61033       -   -   0s
32|    0     0  229.30837    0   3         -  229.30837       -   -   0s
33|    0     2  229.30837    0   3         -  229.30837       -   -   0s
34| *   12     6                9   220.0000000  224.56863   2.08%   1.2   0s
35|
36| Explored 13 nodes (18 simplex iterations) in 0.00 seconds (0.00 work units)
37| Thread count was 1 (of 20 available processors)
38|
39| Solution count 1: 220
40|
41| Optimal solution found (tolerance 2.00e-02)
42| Best objective 2.200000000000e+02, best bound 2.240000000000e+02, gap 1.8182%

```

Domande

1. Perché il solver si ferma? Ricalcolare il gap finale dall'ultima riga.
2. Che cosa garantisce la frase della riga 41 sulla soluzione di valore 220?

3. Con il MIPGap predefinito il valore ottimo di questa istanza è 223. È coerente con il log?

Soluzione

1. Riga 42: miglior valore della funzione obiettivo 220 e best bound 224. Il valore 224.569 di **BestBd** nella riga 34 è arrotondato per difetto dal solver, poiché tutti i dati sono interi. Il gap finale è

$$\frac{224 - 220}{220} \approx 1.82\% < 2\%,$$

sotto MIPGap: il solver si ferma dopo 13 nodi (riga 36) invece di 25.

2. Garantisce solo che $220 \leq z(\text{ILP}) \leq 224$: la soluzione dista al più il 2% dall'ottimo, **non** che sia ottima. Il solver scrive «Optimal solution found» con la tolleranza in uso, qui $2 \cdot 10^{-2}$.
3. Sì: 223 sta nell'intervallo $[220, 224]$ dato dai due bound. La soluzione incumbent è 3 unità, circa l'1.35%, sotto l'ottimo, e servirebbero i nodi rimanenti per trovarlo e dimostrarne l'ottimalità.

Con $\text{MIPGap} = 0.02$ il solver si ferma con $\text{LB} = 220$ e $\text{UB} = 224$: la soluzione è certificata solo entro il 2%, e infatti **non è ottima** ($z(\text{ILP}) = 223$).