Metodi e Modelli per l'Ottimizzazione Combinatoria

Metodi basati su generazione di colonne

L. De Giovanni G. Zambelli

1 Un problema di taglio di tondini di ferro

Un'azienda metallurgica produce tondini di ferro di 15 mm di diametro e della lunghezza standard di 11 metri. L'azienda si occupa anche del taglio dei tondini per i clienti, che richiedono tondini di lunghezza diversa. In questo momento, l'azienda deve soddisfare la seguente commessa:

pezzo	lunghezza (m)	n. di pezzi richiesti
1	2,0	48
2	4,5	35
3	5,0	24
4	5,5	10
5	7,5	8

Determinare il numero minimo di tondini in stock da utilizzare per soddisfare la commessa.

Un possibile modello per il problema, proposto da Gilmore e Gomory nel 1960[1] è il seguente.

Insiemi

- $I = \{1, 2, 3, 4, 5\}$: insieme dei pezzi;
- J: insieme degli schemi (o pattern) di taglio (modi possibili di tagliare i pezzi dai tondini in stock).

Parametri

- W: lunghezza del tondino in stock;
- L_i: lunghezza del pezzo $i \in I$;

[1]P.C.Gilmore and R.E.Gomory, "A linear programming approach to the cutting stock problem", Operations Research, Vol. 9, No. 6 (Nov. - Dec., 1961), pp. 849-859

- R_i: numero di pezzi richiesti per il tipo $i \in I$;
- N_{ij}: numero di pezzi di tipo $i \in I$ nello schema di taglio $j \in J$.

Variabili decisionali

- x_j: numero di tondini in stock da tagliare secondo lo schema $j \in J$.

Modello

$$
\begin{array}{rrcll}
\min & \sum_{j \in J} x_j & & & \\
s.t. & \sum_{j \in J} N_{ij} x_j & \geq & R_i & \forall \;\; i \in I \\
 & x_j & \in & \mathbb{Z}_+ & \forall \;\; j \in J
\end{array}
$$

2 Soluzione del problema

Il modello è elegante ma presuppone di avere a disposizione l'insieme J e i parametri N_{ij}. Per generare tali parametri bisogna generare tutte le possibili configurazioni di taglio. Si può facilmente intuire che il numero di possibili configurazioni di taglio è molto elevato e non consente di implementare direttamente il metodo quando il numero di pezzi e/o il rapporto tra W e le lunghezze dei singoli pezzi diventano grandi.

Osserviamo anche che ha senso risolvere il rilassamento continuo del modello sopra proposto. Infatti, nei casi pratici, i volumi di produzione sono tali da richiedere un numero elevato di tondini tagliati secondo un determinato schema di taglio e pertanto, una buona soluzione euristica potrebbe essere determinata arrotondando all'intero superiore ogni x_j ottenuta dalla soluzione ottima del rilassamento continuo. Inoltre, la soluzione del rilassamento continuo potrebbe essere la base per l'applicazione di un metodo di soluzione esatta (ad esempio il Branch-and-Bound).

Consideriamo la soluzione del rilassamento continuo del problema.

Supponiamo di avere a disposizione una soluzione ammissibile del problema. Nel nostro caso una soluzione è ottenuta:

1. considerando delle configurazioni di taglio mono-pezzo, ossia $card(I)$ configurazioni che contengono $N_{ii} = \lfloor W/L_i \rfloor$ pezzi di tipo i;
2. ponendo $x_i = \lceil R_i/N_{ii} \rceil$ per ogni configurazione i (che contiene solo il pezzo i).

La stessa soluzione potrebbe essere ottenuta applicando il metodo del simplesso al modello sopra proposto (eliminando i vincoli di interezza) dove si considerano le sole variabili relative ai pattern mono-pezzo considerati:

$$
\begin{array}{rccccccccccc}
\min & x_1 & + & x_2 & + & x_3 & + & x_4 & + & x_5 & & \\
s.t. & 5x_1 & & & & & & & & & \geq & 48 \\
 & & & 2x_2 & & & & & & & \geq & 35 \\
 & & & & & 2x_3 & & & & & \geq & 24 \\
 & & & & & & & 2x_4 & & & \geq & 10 \\
 & & & & & & & & & x_5 & \geq & 8 \\
 & x_1 & & x_2 & & x_3 & & x_4 & & x_5 & \geq & 0
\end{array}
$$

In effetti, la soluzione con ampl-cplex del problema proposto e considerando solo i pattern mono-pezzo è:

```
    x [*] :=
1 9.6
2 17.5
3 12
4 5
5 8
```

Supponiamo adesso di considerare un nuovo possibile pattern con 1 pezzo di tipo 1 e un pezzo di tipo 5: avendo a disposizione questo nuovo pattern, la soluzione precedente rimane ottima? Come abbiamo visto, per rispondere a questa domanda possiamo ricorrere alla teoria della dualità o alla teoria del simplesso. Ricordiamo che il simplesso ci fornisce una soluzione ammissibile (di base B) per il problema primale e una soluzione duale (i moltiplicatori $u^T = c_B^T B^{-1}$) in scarti complementari (soluzione che, alla fine del simplesso, è anche duale-ammissibile). Il nuovo pattern 6 corrisponde all'inserimento di una nuova variabile nel problema primale con costo in funzione obiettivo pari a 1 (consuma un tondino) e la cui colonna nella matrice dei vincoli è:

$$
A_6 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}
$$

Tale variabile introduce un nuovo vincolo del problema duale. Bisogna pertanto capire se tale vincolo è violato dalla soluzione duale corrente (u^T), che equivale a dire che il costo ridotto della nuova variabile rispetto alla base ottima corrente è negativo. Il vincolo duale corrispondente alla nuova variabile è:

$$
1u_1 + 0u_2 + 0u_3 + 0u_4 + 1u_5 \leq 1
$$

Considerando la soluzione duale corrispondente alla soluzione ottima corrente $u = c_B^T B^{-1}$, ottenibile interrogando direttamente AMPL (`Fill[i]` è il nome del vincolo relativo alla misura `i`):

```
Fill.dual [*] :=
    20 0.2
    45 0.5
    50 0.5
    55 0.5
    75 1.0
```

Si ottiene $0.2 + 1 = 1.2 > 1$. Il vincolo duale è quindi violato dalla soluzione u che, per costruzione, è in scarti complementari con la soluzione primale x corrente. Ciò significa che la soluzione corrente (dove $x_6 = 0$), pur essendo ammissibile, non è ottima (o meglio, potrebbe non essere ottima). Verifichiamo che il vincolo duale violato corrisponde al fatto che il costo ridotto della variabile associata al nuovo pattern:

$$\bar{c}_6 = c_6 - u^T A_6 = 1 - \begin{bmatrix} 0,2 & 0,5 & 0,5 & 0,5 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = -0,2$$

è negativo. Conviene quindi far entrare in base x_6 e, per far questo, modifichiamo il problema di partenza inserendo la nuova variabile in funzione obiettivo. Il problema diventa:

$$\begin{array}{lccccccccccccc}
\min & x_1 & + & x_2 & + & x_3 & + & x_4 & + & x_5 & + & x_6 & & \\
s.t. & 5x_1 & & & & & & & & & + & x_6 & \geq & 48 \\
 & & & 2x_2 & & & & & & & & & \geq & 35 \\
 & & & & & 2x_3 & & & & & & & \geq & 24 \\
 & & & & & & & 2x_4 & & & & & \geq & 10 \\
 & & & & & & & & & x_5 & + & x_6 & \geq & 8 \\
 & x_1 & & x_2 & & x_3 & & x_4 & & x_5 & & x_6 & \geq & 0
\end{array}$$

Analogamente a quanto visto prima, la nuova soluzione di base ottima ottenuta con il simplesso applicato al nuovo problema corrisponde a dei nuovi valori ottimi delle variabili duali. Se si disponesse di una nuova possibile configurazione di taglio *ammissibile*, potremmo quindi valutare la convenienza a far entrare in base, *e quindi nel problema*, la nuova variabile relativa alla nuova configurazione.

Il problema a questo punto è che non è immediato individuare una variabile, e quindi una colonna della matrice dei vincoli, con costo ridotto negativo. Infatti, il numero di potenziali colonne è molto elevato e, inoltre, *le colonne non sono note in modo esplicito.* La domanda è quindi:

> *Data una soluzione di base ottima per il problema con un ridotto numero di variabili, come determiniamo, se esiste, una variabile con costo ridotto negativo (cioè un vincolo duale violato dalla corrispondente soluzione duale)?*

Possiamo trasformare questa domanda in un problema di ottimizzazione: per giudicare l'esistenza di un costo ridotto negativo, possiamo calcolare il minimo tra tutti i costi ridotti delle potenziali variabili del problema e verificare se è negativo.

$$\begin{array}{ll} \min & \bar{c} = 1 - u^T z \\ s.t. & z \text{ è una possibile colonna della matrice dei vincoli} \end{array}$$

Ricordiamo che ogni colonna della matrice dei vincoli corrisponde ad uno schema di taglio e ogni elemento della colonna indica quanti pezzi di un certo tipo sono contenuti nello stesso schema. Affinché z sia una possibile colonna della matrice dei vincoli è sufficiente che:

$$\begin{array}{c} z \in \mathbb{Z}_+ \\ \sum_{i \in I} L_i z_i \leq W \end{array}$$

Il problema dell'individuazione di una variabile a costo ridotto negativo si traduce quindi nella soluzione del seguente problema di programmazione lineare intera:

$$\begin{array}{lrcl} \min & \bar{c} = 1 - \sum_{i \in I} u_i z_i & & \\ s.t. & \sum_{i \in I} L_i z_i & \leq & W \\ & z & \in & \mathbb{Z}_+^{card(I)} \end{array}$$

che equivale (passando a una funzione obiettivo di massimo e eliminando la costante additiva) a:

$$\begin{array}{lrcl} \max & \sum_{i \in I} u_i z_i & & \\ s.t. & \sum_{i \in I} L_i z_i & \leq & W \\ & z & \in & \mathbb{Z}_+^{card(I)} \end{array}$$

I coefficienti z_i di una colonna a costo ridotto negativo possono essere ottenuti risolvendo un problema di zaino intero.

In effetti, si può verificare come $z^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \end{bmatrix}$ sia la soluzione ottima del problema

$$\begin{array}{rrcrcrcrcrcl} \max & 0,2z_1 & + & 0,5z_2 & + & 0,5z_3 & + & 0,5z_4 & + & z_5 & & \\ s.t. & 2,0z_1 & + & 4,5z_2 & + & 5,0z_3 & + & 5,5z_4 & + & 7,5z_5 & \leq & 11 \\ & z_1 & & z_2 & & z_3 & & z_4 & & z_5 & \in & \mathbb{Z}_+ \end{array}$$

3 Un algoritmo per il problema del taglio unidimensionale (one-dimensional cutting-stock problem)

Il procedimento esemplificato nella precedente sezione può essere generalizzato come un algoritmo per la soluzione di problemi di taglio ad una dimensione.

Problema 1 (Taglio unidimensionale)*: dati*

- *un insieme I di oggetti,*
- *per ogni oggetto $i \in I$, la lunghezza L_i, e il numero di pezzi da produrre R_i,*
- *la lunghezza standard W dei "maxi-oggetti" da tagliare,*

si determini il numero minimo di "maxi-oggetti" necessari per soddisfare le richieste degli oggetti in I.

Il problema può essere modellato come segue:

$$\begin{array}{llcll} \min & \sum_{j \in J} x_j & & & \\ s.t. & \sum_{j \in J} N_{ij} x_j & \geq & R_i & \forall \;\; i \in I \\ & x_j & \in & \mathbb{Z}_+ & \forall \;\; j \in J \end{array}$$

dove:

- J: insieme degli schemi (o pattern) di taglio per ottenere oggetti in I da "maxi-oggetti" di lunghezza W;
- N_{ij}: numero di pezzi di tipo $i \in I$ nello schema di taglio $j \in J$.
- x_j: numero di "maxi-oggetti" da tagliare secondo lo schema $j \in J$.

Un possible algoritmo per la soluzione del problema si basa sulla soluzione del rilassamento continuo del modello descritto, ossia il modello ottenuto sostituendo i vincoli $x_j \in \mathbb{Z}_+ \forall j \in J$ con i vincoli $x_j \in \mathbb{R}_+ \forall j \in J$.

Osservando che $card(J)$ può essere grande e tale da non rendere conveniente/attuabile la definizione esplicita di tutti i possibili schemi di taglio, si applica il seguente algoritmo:

Passo 0: inizializzazione
Si generi un insieme J' di schemi di taglio tale che il problema ammetta soluzione (un esempio è l'adozione di $card(I)$ schemi mono-pezzo).

Passo 1: soluzione del problema master
Si risolva il problema master

$$\begin{array}{rrcll} \min & \sum_{j \in J'} x_j & & & \\ s.t. & \sum_{j \in J'} N_{ij} x_j & \geq & R_i & \forall \ i \in I \\ & x_j & \in & \mathbb{R}_+ & \forall \ j \in J' \end{array}$$

ottenendo una soluzione ottima primale x^* e una soluzione ottima duale u corrispondente (cioè in scarti complementari con u^* (ad esempio con il metodo del simplesso).

Passo 2: soluzione del sotto-problema (problema slave)
Si risolva il seguente problema di programmazione lineare intera con $card(I)$ variabili e un vincolo:

$$\begin{array}{rrcll} \max & \sum_{i \in I} u_i^* z_i & & & \\ s.t. & \sum_{i \in I} L_i z_i & \leq & W & \\ & z_i & \in & \mathbb{Z}_+ & \forall \ i \in I \end{array}$$

ottenendo la soluzione ottima z^*.

Passo 3: test di ottimalità
Se $\sum_{i \in I} u_i^* z_i^* \leq 1$ STOP: x^* è la *soluzione ottima* del rilassamento continuo (esteso a tutto J). Altrimenti, *aggiorna il problema master* aggiungendo a J' lo schema di taglio γ con $N_{i\gamma} = z_i^*$ (aggiungi la colonna z^* alle colonne della matrice dei vincoli del problema master) e torna al Passo 1.

Per passare dalla soluzione ottima del rilassamento continuo x^* ad una soluzione *euristica* (quindi non necessariamente ottima ma sperabilmente buona) del problema originario (a variabili intere) si può, in alternativa:

- arrotondare per eccesso le componenti del vettore x^* (efficace se tali componenti assumono valori elevati: 765.3 non è poi così diverso da 766...);
- applicare un metodo di programmazione lineare intera (ad esempio il Branch-and-Bound) all'ultimo problema master generato, ossia risolvere il problema originario limitato all'insieme dei soli pattern "buoni" generati dalla soluzione dei sotto-problemi.

4 Metodi di basati su generazione di colonne per problemi di programmazione lineare

L'idea sviluppata per il problema di taglio monodimensionale può essere applicata per la soluzione di problemi di programmazione lineare (NON intera, almeno direttamente) tutte le volte che non sia possibile/conveniente considerare esplicitamente tutte le variabili decisionali coinvolte.

Sia dato il problema:

$$\begin{array}{rrrl} (P) & \min & c^T x & \\ & \text{s.t.} & Ax & = b \\ & & x & \geq 0 \end{array}$$

con $A \in \mathbb{R}^{m \times n}$ e si supponga che $\dim(x) = n$ sia molto elevato e/o le colonne di A non siano tutte note a priori. La soluzione del problema può essere ottenuta applicando il seguente metodo. Si ricorda che al problema (P) è possibile associare il corrispondente problema duale

$$\begin{array}{rrrl} (D) & \max & u^T b & \\ & \text{s.t.} & u^T A & \leq \; c^T \\ & & u & \; \textit{libero} \end{array}$$

Passo 0: inizializzazione

Si espliciti un sottoinsieme delle colonne di A e si consideri la sotto-matrice di A corrispondente a tali colonne. Assicuriamo che il problema P abbia una soluzione ammissibile e limitata usando le sole variabili associate alle colonne esplicitate. Chiamiamo tale matrice $E \in \mathbb{R}^{m \times q} (q < n)$ e x_E, c_E i vettori delle corrispondenti variabili e dei corrispondenti costi.

Passo 1: soluzione del Problema Master

Si risolva il seguente problema master

$$\begin{array}{rrrl} (MP) & \min & c_E^T x_E & \\ & \text{s.t.} & E x_E & = b \\ & & x_E & \geq 0 \end{array}$$

ottenendo una coppia di soluzioni ammissibili e ottime primale-duale x_E^M e u^M.

> *Nota teorica.* Consideriamo la partizione $A = \begin{bmatrix} E | H \end{bmatrix}$ (Explicit, Hidden columns) e le corrispondenti $x = \begin{bmatrix} x_E \\ x_H \end{bmatrix}$ e $c^T = \begin{bmatrix} c_E^T | c_H^T \end{bmatrix}$. Si fa notare che, ponendo $x = \begin{bmatrix} x_E = x_E^M \\ x_H = 0^{n-q} \end{bmatrix}$ si ottiene una soluzione *ammissibile* per il problema di partenza (P) (tutti i vincoli sono rispettati!). Inoltre, la soluzione $u = u^M$ è una soluzione del problema duale (D): ha lo stesso numero di componenti, visto che (P) e (MP) hanno lo stesso numero di vincoli. Infine,

u e x sono in scarti complementari, con riferimento alla coppia di problemi di partenza (P)-(D). Infatti:

$$(c^T - u^T A)x = ([c_E^T | c_H^T] - u^T[E|H]) \begin{bmatrix} x_E \\ x_H \end{bmatrix} = (c_E^T - u^T E)x_E + (c_H^T - u^T H)\underbrace{x_H}_{=0} =$$

$$\underbrace{(c_E^T - (u^M)^T E)x_E^M}_{=0} + (c_H^T - (u^M)^T H) \cdot 0 = 0 \cdot x_E^M + 0 = 0$$

essendo x_E^M e u^M in scarti complementari (ottime) ed essendo $x_H = 0$.

Una coppia di soluzioni ammissibili e ottime primale-duale per (MP) è ottenibile *ad esempio* con il metodo del simplesso.

Passo 2: soluzione del Problema Slave (sotto-problema di generazione di colonne)
Determinare uno (o più) vettori $z \in \mathbb{R}^m$ che soddisfino i seguenti vincoli:

(i) z rappresenta i coefficienti nei vincoli di una variabile x_j (ossia z è una possibile colonna A_j di A) di costo c_j;

(ii) $c_j - (u^M)^T z < 0$.

Nota teorica. Le condizioni identificano l'esistenza di un vincolo del problema duale (D) (originario) violato dalla soluzione $u = u^M$. Si noti che (D) contiene anche tutti i vincoli del duale di (MP), corrispondenti alle variabili in x_E. Ovviamente, tali vincoli sono soddisfatti, vista l'ammissibilità di u^M per il duale di (MP).

Passo 3: test di ottimalità
Se non esiste nessun vettore z che soddisfi i vincoli precedenti, allora **STOP**: la soluzione $x = \begin{bmatrix} x_D^M \\ 0 \end{bmatrix}$ è la soluzione ottima di (P).

Nota teorica. Abbiamo visto che $x = \begin{bmatrix} x_E = x_E^M \\ x_H = 0^{n-q} \end{bmatrix}$ e $u = u^M$ formano una coppia di soluzioni primale-duale per (P)-(D) in scarti complementari. Il fatto che il sotto-problema di generazione di colonne sia inammissibile indica che non esistono vincoli del problema duale (D) violati, cioè: $u = u^M$ è anche duale-ammissibile. Siamo quindi in presenza di una coppia di soluzioni primale-duale per (P)-(D) ammissibili e in scarti complementari. Per il teorema della dualità forte, x e u sono ottime per (P) e (D), rispettivamente.

Passo 4: Iterazione
Aggiornare il problema master aggiungendo una (o più) colonne generate al Passo 2 alla matrice E e i corrispondenti variabili e costi in x_E e c_E. Iterare dal Passo 1.

> *Nota teorica.* Abbiamo visto che vincoli duali violati corrispondono a variabili a costo ridotto negativo, che conviene considerare per avere la possibilità di migliorare il valore della funzione obiettivo.

5 Metodi basati su generazione di colonne: alcuni aspetti implementativi

5.1 Il sotto-problema di generazione di colonne

Il punto critico del metodo è il Passo 2 e cioè la soluzione del problema di generazione di colonne. Ovviamente non è possibile calcolare tutti i costi ridotti di tutte le variabili $x_j, j = 1 \dots n$, altrimenti il metodo si ricondurrebbe al simplesso. In effetti, n potrebbe essere molto grande (vedi il caso del cutting-stock) oppure potrebbe non essere conveniente/possibile esplicitare tutte le variabili decisionali.

> È necessario quindi studiare un algoritmo di generazione di colonne specifico per ciascun problema e l'applicabilità del metodo è condizionata dalla disponibilità e dell'efficienza di tale algoritmo.

Nel caso del taglio monodimensionale, abbiamo trasformato il sotto-problema di generazione di colonne in un problema di ottimizzazione che si riconduce ad un problema di programmazione lineare intera di semplice soluzione (risolvibile in modo efficiente con un metodo di Branch-and-Bound specializzato o con programmazione dinamica). In altri casi, la soluzione potrebbe richiedere uno sforzo computazionale tale da compromettere l'efficienza complessiva del metodo: ad esempio, trasformando in problema di ottimizzazione, si può ottenere un problema di programmazione lineare intera non facilmente risolvibile. In generale (e in modo informale), potremmo dire che se il problema di generazione di colonne è NP-Hard, il metodo sarà complessivamente inefficiente.

5.2 Convergenza del metodo

5.2.1 Ammissibilità e limitatezza dei problemi master

L'algoritmo di generazione di colonne assume come input una coppia di soluzioni ottime primale-duale. La condizione affinché il Passo 1 sia in grado di fornire tale coppia è che *il problema master sia sempre ammissibile e limitato.* Alla prima iterazione, questo può essere garantito trovando una soluzione ammissibile per (P) e considerando E come l'insieme di tutte e sole le colonne delle variabili che sono strettamente positive in tale soluzione. Inoltre, per garantire la limitatezza, si possono mettere dei *box constraints*, ossia dei vincoli $x_j \leq M$, $\forall\, j \in E$ (con M costante sufficientemente grande). Se il metodo continua ad aggiungere variabili a (MP), tutti gli (MP) saranno ammissibili (conserveranno le colonne di partenza e quindi la soluzione di partenza). Inoltre, aggiungendo dei *box constraints* sulle variabili via via introdotte al Passo 4, si garantisce la limitatezza.

5.2.2 Velocità di convergenza

La convergenza dei metodi basati su generazione di colonne è garantita dalla teoria del simplesso, a patto di disporre di un algoritmo esatto per la soluzione del problema di generazione di colonne. Tuttavia, da un punto di vista pratico, la convergenza potrebbe essere lenta, per diversi problemi, dei quali citiamo solo i seguenti.

Uno dei problemi risiede nel fatto che, se al Passo 4 introduciamo una sola variabile, bisognerà attendere diverse iterazioni prima di avere un insieme di colonne che contengano quelle relative a soluzione ottima. Per far questo, si potrebbe pensare, qualora sia possibile, di individuare (al Passo 2) e di inserire nel (MP)(al Passo 4) più colonne per iterazione.

Un altro problema è determinato dalle dimensioni del problema (MP) che, a un certo punto, potrebbe contenere un numero molto elevato di variabili, tale da pregiudicare l'efficienza della sua soluzione. Per ovviare a questo problema si potrebbe mantenere un pool di colonne attive nel (MP), tra tutte quelle generate fino a quel momento. Ciò significa eliminare dal (MP) qualche variabile, ad esempio tra quelle che non entrano in soluzioni ottime da più iterazioni. Ovviamente, nell'eliminare delle variabili dal (MP) bisogna stare attenti a non perdere l'ammissibilità dello stesso, per quanto sopra osservato. A questo punto, l'algoritmo di generazione di colonne potrebbe essere attivato solo dopo aver verificato che, tra le colonne prima generate ma correntemente non attive, non ce ne siano alcune a costo ridotto negativo.