Ricerca Operativa

A.A. 2007/2008

16. Branch and Bound per Programmazione Lineare Intera.

Algoritmo del B&B: schema di principio

P_0: problema di ottimizzazione iniziale;
L: lista dei nodi aperti (coppie (P_i, L_i));
$\bar{x}$: migliore soluzione ammissibile corrente;
$\bar{z}$: valore della migliore soluzione ammissibile.

0. *Inizializzazione*: Esegui una stima ottimistica L_0 della funzione obiettivo e poni $L = \{(P_0, L_0)\}$, $\bar{x} = \emptyset$, $\bar{z} = \pm\infty$

1. *Criterio di Stop*: Se $L = \emptyset$, allora STOP: $\bar{x}$ è la soluzione ottima.

   Se superati limiti di tempo, nodi esplorati, nodi aperti $|L|$ etc.
   STOP: $\bar{x}$ è una soluzione (non necessariamente ottima).

2. *Selezione nodo*: Seleziona $(P_i, L_i) \in L$ (regola di priorità)

3. *Branching*: Dividi P_i in t sottoproblemi P_{ij}, $j = 1..t$ $(\cup_j P_j = P_i)$

4. *Bounding*: Valuta una stima ottimistica L_{ij} (con soluzione x_{ij}^R) per ciascun sottoproblema P_{ij}

5. *Fathoming*: Se P_{ij} non è ammissibile, vai a 1.

   Se L_{ij} non è migliore di $\bar{z}$ ammissibile, vai a 1.

   Se x_{ij}^R è intera (e migliore di $\bar{z}$), poni $\bar{z} \leftarrow L_{ij}$, $\bar{x} \leftarrow x_{ij}^R$;
   elimina da L tutti i nodi k con L_k non migliore di $\bar{z}$; vai a 1.

   Altrimenti, aggiungi (P_{ij}, L_{ij}) a L e vai a 1.

B&B per Programmazione Lineare Intera

- Si consideri un problema di Programmazione Lineare Intera (PLI) nella forma:

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & \left.\begin{array}{l} Ax \geq b \\ x \in \mathbb{Z}^n_+ \end{array}\right\} x \in X \end{array}$$

$$\begin{array}{l} . \\ P = \{x \in \mathbb{R}^n_+ : Ax \geq b\} \\ X = P \cap \mathbb{Z}^n_+ \end{array}$$

- Bisogna definire le seguenti componenti principali:

(1) Calcolo del bound (lower bound, LB).

(2) Regole di Branching.

(3) Regole di Fathoming.

(4) Regole di esplorazione dell'albero di ricerca.

(5) Valutazione di soluzioni ammissibili.

(6) Criteri di arresto.

B&B per PLI: (1) calcolo del bound

Introduciamo un problema rilassato:

$$(P)\quad \begin{array}{rl} z^* = \min & f(x) \\ & x \in X \end{array} \qquad\qquad (P_R)\quad \begin{array}{rl} z_R^* = \min & f(x) \\ & x \in Y \end{array}$$

Def. P_R è un **rilassamento** di P se $X \subseteq Y$

Proprietà del rilassamento:

* Se $Y = \emptyset$ anche $X = \emptyset$: P_R inammissibile $\Rightarrow$ P inammissibile.
* $z_R^* \leq z^*$: z_R^* è una valutazione ottimistica (lower bound) di z^*

Un rilassamento si può ottenere, ad esempio, eliminando alcuni vincoli, cercando di ottenere un problema *facile* da risolvere.

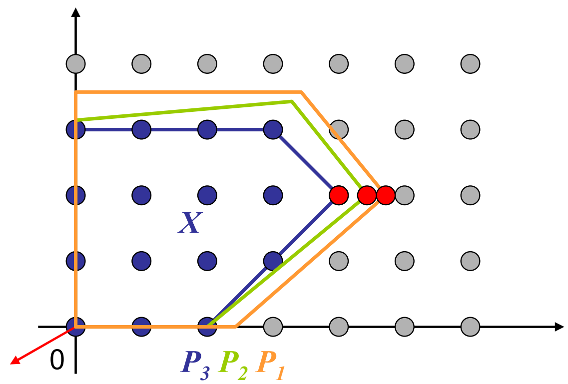
Rilassamento continuo: si elimina il vincolo di interezza delle variabili (risolvibile in modo efficiente con il *simplesso*).

$$(P)\quad \begin{array}{rl} z^* = \min & c^T x \\ & Ax \geq b \\ & x \in \mathbb{Z}_+^n \end{array} \qquad (P_R)\quad \begin{array}{rl} z_R^* = \min & c^T x \\ & Ax \geq b \\ & x \in \mathbb{R}_+^n \end{array} \qquad \boxed{z_R^* \leq z^*}$$

Considerazioni sul rilassamento continuo

Il lower bound deve essere il più **stringente** (alto) possibile.

- Se c *è tutto **intero***: $z^* \geq z_R^* \quad \Rightarrow \quad z^* \geq \lceil z_R^* \rceil \quad \Rightarrow \quad \boxed{LB = \lceil z_R^* \rceil}$
- Consideriamo il problema $\min\{c^T x : x \in X \subseteq \mathbb{Z}_+^n\}$



$$P_1 = \{x \geq 0 : A_1 x \geq b_1\}$$

$$P_2 = \{x \geq 0 : A_2 x \geq b_2\}$$

$$P_3 = \{x \geq 0 : A_3 x \geq b_3\}$$

$$P_1 \cap \mathbb{Z}_+^n = P_2 \cap \mathbb{Z}_+^n = P_3 \cap \mathbb{Z}_+^n = X$$

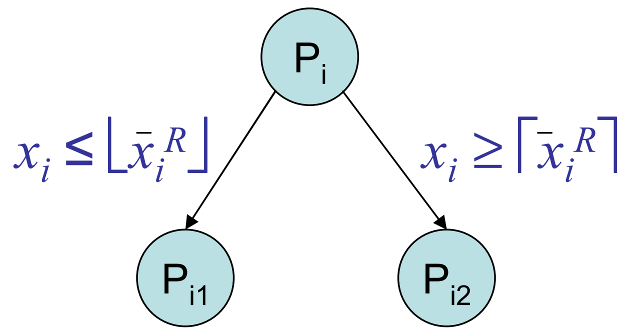
  - Esistono diverse **formulazioni equivalenti** per un problema PLI.
  - $z_R^{*1} \leq z_R^{*2} \leq z_R^{*3}$ tutti lower bound! Meglio la formulazione più stringente
  - $z_R^{*3} = z^*$: esiste (almeno in teoria) una **formulazione ideale** con tutti i vertici interi. $P_3 = conv(X)$ (involucro convesso): $z^* = \min\{c^x : x \in conv(X), x \in \mathbb{R}_+^n\}$

Esempio: vincolo i con *tutte variabili e tutti coefficienti interi, tranne* b_i :

$$a_i^T x \begin{cases} = b_i \\ \leq b_i \\ \geq b_i \end{cases} \quad \rightsquigarrow \quad a_i^T x \begin{cases} = \lfloor b_i \rfloor \\ \leq \lfloor b_i \rfloor \\ \geq \lceil b_i \rceil \end{cases}$$

B&B per PLI: (2) branching

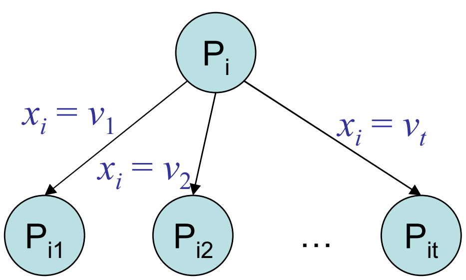
(2) Regole di Branching:
sia $\bar{x}^R$ la soluzione del rilassamento continuo e $x_i = \bar{x}_i^R$ una variabile con componente frazionaria $\varphi(\bar{x}_i^R) \neq 0$.
Proponiamo un *branching dicotomico*:



Si può dare priorità alle x_i frazionarie con $\varphi(\bar{x}_i^R)$ più prossimo a 0.5.

Nota. Da un punto di vista computazionale, si tratta di aggiungere un vincolo: passiamo da una soluzione ottima e ammissibile per P_i ad una soluzione ottima ma non ammissibile per P_{i1} e P_{i2}: i sottoproblemi possono essere risolti in modo efficiente con il *simplesso duale*!

.
Altra possibilità (meno generalizzabile): branching $t-$ario, considerando tutti i possibili valori v_k assumibili da una variabile x_i:



B&B per PLI: altre scelte implementative

(3) Regole di Fathoming:
standard.

(4) Regole di esplorazione dell'albero di ricerca:
Tra le possibili regole, proponiamo *Best Bound First*.

(5) Valutazione di soluzioni ammissibili:
Si può applicare un'euristica che trasformi la soluzione del rilassamento in una soluzione intera ammissibile, valutando il valore della funzione obiettivo, nel tentativo di migliorare la soluzione ammissibile corrente.

Una semplice euristica: arrotondare $\bar{x}_i^R$ all'intero più vicino e verificare l'ammissibilità (non molto efficace...). Si potrebbe migliorare con algoritmi di ricerca locale...

Bisogna valutare il compromesso tra la facilità (tempo) di calcolo, qualità della soluzione ottenibile e frequenza di applicazione.

(6) Criteri di arresto:
Vogliamo l'ottimo: ci fermiamo quando tutti i nodi sono *fathomed*.

Esempio

$$
\begin{aligned}
\min \ & -3x_1 - 4x_2 + x_3 \\
s.t. \ & x_1 + x_2 \leq 9/2 \\
& 2x_1 - x_2 - x_3 = 7 \\
& x_i \in \mathbb{Z}_+
\end{aligned}
$$