

Branch and Bound

Branch-and-bound

Tra le tecniche di risoluzione esatte per problemi difficili una molto popolare è quella denominata **branch-and-bound**.

Branch-and-bound

Tra le tecniche di risoluzione esatte per problemi difficili una molto popolare è quella denominata **branch-and-bound**.

Descriveremo l'algoritmo generico di branch-and-bound per problemi di massimo. Di seguito segnaleremo le piccole variazioni che vanno introdotte per problemi di minimo.

Componenti algoritmo:upper bound

Sia data la regione ammissibile S e la funzione obiettivo f dell'istanza di un problema di ottimizzazione. Si consideri un sottinsieme $T \subseteq S$ della regione ammissibile. Una limitazione superiore o *upper bound* per T è un valore $U(T)$ con la seguente proprietà

$$U(T) \geq f(x) \quad \forall x \in T.$$

Il calcolo dell'upper bound

Il valore $U(T)$ viene calcolato tramite una procedura che deve cercare di soddisfare queste due proprietà in conflitto tra loro:

Il calcolo dell'upper bound

Il valore $U(T)$ viene calcolato tramite una procedura che deve cercare di soddisfare queste due proprietà in conflitto tra loro:

- i tempi di esecuzione della procedura devono essere brevi (in particolare, il calcolo degli upper bound deve richiedere un tempo molto inferiore rispetto al tempo necessario per risolvere l'intero problema);

Il calcolo dell'upper bound

Il valore $U(T)$ viene calcolato tramite una procedura che deve cercare di soddisfare queste due proprietà in conflitto tra loro:

- i tempi di esecuzione della procedura devono essere brevi (in particolare, il calcolo degli upper bound deve richiedere un tempo molto inferiore rispetto al tempo necessario per risolvere l'intero problema);
- il valore $U(T)$ deve essere il più vicino possibile al massimo valore di f su T .

Il calcolo dell'upper bound

Il valore $U(T)$ viene calcolato tramite una procedura che deve cercare di soddisfare queste due proprietà in conflitto tra loro:

- i tempi di esecuzione della procedura devono essere brevi (in particolare, il calcolo degli upper bound deve richiedere un tempo molto inferiore rispetto al tempo necessario per risolvere l'intero problema);
- il valore $U(T)$ deve essere il più vicino possibile al massimo valore di f su T .

Spesso la scelta di una procedura per il calcolo dell'upper bound è fortemente legata al particolare problema che si sta risolvendo. Inoltre, non esiste un'unica procedura per un dato problema.

Upper bound e rilassamento

Un modo comunemente utilizzato per determinare un upper bound $U(T)$ è quello di determinare la soluzione di un suo *rilassamento*.

Upper bound e rilassamento

Un modo comunemente utilizzato per determinare un upper bound $U(T)$ è quello di determinare la soluzione di un suo *rilassamento*.

Indichiamo con:

$$\alpha(f, T) = \max_{x \in T} f(x),$$

il valore ottimo della funzione f sull'insieme T .

Upper bound e rilassamento

Un modo comunemente utilizzato per determinare un upper bound $U(T)$ è quello di determinare la soluzione di un suo *rilassamento*.

Indichiamo con:

$$\alpha(f, T) = \max_{x \in T} f(x),$$

il valore ottimo della funzione f sull'insieme T .

Si definisce rilassamento del problema, un problema:

$$\alpha(f', T') = \max_{x \in T'} f'(x)$$

dove:

$$T \subseteq T' \quad \mathbf{e} \quad f'(x) \geq f(x) \quad \forall x \in T.$$

Osservazione

Si ha che: $\alpha(f', T') \geq \alpha(f, T)$.

Osservazione

Si ha che: $\alpha(f', T') \geq \alpha(f, T)$.

Dimostrazione Sia $x^* \in T$ una soluzione ottima del problema su T , cioè:

$$f(x^*) = \alpha(f, T),$$

e sia $x' \in T'$ una soluzione ottima del rilassamento, cioè:

$$f'(x') = \alpha(f', T').$$

Osservazione

Si ha che: $\alpha(f', T') \geq \alpha(f, T)$.

Dimostrazione Sia $x^* \in T$ una soluzione ottima del problema su T , cioè:

$$f(x^*) = \alpha(f, T),$$

e sia $x' \in T'$ una soluzione ottima del rilassamento, cioè:

$$f'(x') = \alpha(f', T').$$

Si ha che $x^* \in T$ implica $x^* \in T'$.

Osservazione

Si ha che: $\alpha(f', T') \geq \alpha(f, T)$.

Dimostrazione Sia $x^* \in T$ una soluzione ottima del problema su T , cioè:

$$f(x^*) = \alpha(f, T),$$

e sia $x' \in T'$ una soluzione ottima del rilassamento, cioè:

$$f'(x') = \alpha(f', T').$$

Si ha che $x^* \in T$ implica $x^* \in T'$. Inoltre, si ha:

$$f'(x^*) \geq f(x^*).$$

Osservazione

Si ha che: $\alpha(f', T') \geq \alpha(f, T)$.

Dimostrazione Sia $x^* \in T$ una soluzione ottima del problema su T , cioè:

$$f(x^*) = \alpha(f, T),$$

e sia $x' \in T'$ una soluzione ottima del rilassamento, cioè:

$$f'(x') = \alpha(f', T').$$

Si ha che $x^* \in T$ implica $x^* \in T'$. Inoltre, si ha:

$$f'(x^*) \geq f(x^*).$$

Infine, l'ottimalità di x' implica $f'(x') \geq f'(x^*)$ e quindi:

$$\alpha(f', T') = f'(x') \geq f'(x^*) \geq f(x^*) = \alpha(f, T).$$

Un rilassamento già noto

Esistono molti possibili rilassamenti di un problema. Tra questi, uno che è già stato incontrato è il *rilassamento lineare* per problemi di PLI.

Un rilassamento già noto

Esistono molti possibili rilassamenti di un problema. Tra questi, uno che è già stato incontrato è il *rilassamento lineare* per problemi di PLI.

Sia dato il generico problema di PLI:

$$\begin{aligned} \max \quad & \mathbf{c}\mathbf{x} \\ \text{A}\mathbf{x} \leq & \mathbf{b} \\ \mathbf{x} \geq & \mathbf{0} \quad \mathbf{x} \in \mathbb{Z}^n. \end{aligned}$$

Questo è un particolare problema di ottimizzazione con:

$$f(\mathbf{x}) = \mathbf{cx} \quad T = \{\mathbf{x} \in Z^n : \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}.$$

Questo è un particolare problema di ottimizzazione con:

$$f(\mathbf{x}) = \mathbf{c}\mathbf{x} \quad T = \{\mathbf{x} \in Z^n : \mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}.$$

Il rilassamento lineare di tale problema è un particolare rilassamento con:

$$f'(\mathbf{x}) \equiv f(\mathbf{x}) \quad T' = \{\mathbf{x} \in R^n : \mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\},$$

Il rilassamento lineare coincide con questo problema di PL:

$$\begin{aligned} \max \quad & \mathbf{c}\mathbf{x} \\ \text{Ax} \leq & \mathbf{b} \\ \mathbf{x} \geq & \mathbf{0} \end{aligned}$$

Il rilassamento lineare coincide con questo problema di PL:

$$\begin{array}{ll}\max & \mathbf{c}\mathbf{x} \\ & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}\end{array}$$

NB: come richiesto, il rilassamento lineare, essendo un problema di PL, è risolvibile in tempi molto più rapidi dell'originario problema di PLI.

Il rilassamento lagrangiano

Supponiamo che il nostro problema sia formulato come problema di PLI:

$$\begin{aligned} \max \quad & \mathbf{c}\mathbf{x} \\ & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \mathbf{C}\mathbf{x} \leq \mathbf{d} \\ & \mathbf{x} \geq \mathbf{0} \quad \mathbf{x} \in \mathbb{Z}^n. \end{aligned}$$

Quindi con:

$$f(\mathbf{x}) = \mathbf{c}\mathbf{x}, \quad T = \{\mathbf{x} \in \mathbb{Z}^n : \mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{C}\mathbf{x} \leq \mathbf{d}, \mathbf{x} \geq \mathbf{0}\}.$$

Ipotesi

Supponiamo che i vincoli $Ax \leq b$ siano "facili" (ad esempio, A è TU e b è a coordinate tutte intere).

Ipotesi

Supponiamo che i vincoli $Ax \leq b$ siano "facili" (ad esempio, A è TU e b è a coordinate tutte intere).

Quindi eliminando i vincoli "difficili" $Cx \leq d$ resta un problema di PLI facile da risolvere (basta risolverne il rilassamento lineare).

Ipotesi

Supponiamo che i vincoli $Ax \leq b$ siano "facili" (ad esempio, A è TU e b è a coordinate tutte intere).

Quindi eliminando i vincoli "difficili" $Cx \leq d$ resta un problema di PLI facile da risolvere (basta risolverne il rilassamento lineare).

Per eliminarli li spostiamo nell'obiettivo.

Continua

Dato un vettore $\lambda \geq 0$, detto *vettore dei moltiplicatori di Lagrange*, delle stesse dimensioni di d , il rilassamento lagrangiano è il seguente:

$$\begin{aligned} u(\lambda) = \max \quad & \mathbf{c}\mathbf{x} + \lambda(\mathbf{d} - \mathbf{C}\mathbf{x}) \\ & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \quad \mathbf{x} \in Z^n. \end{aligned}$$

con

$$f'(\mathbf{x}) = \mathbf{c}\mathbf{x} + \lambda(\mathbf{d} - \mathbf{C}\mathbf{x})$$

e

$$T' = \{\mathbf{x} \in Z^n : \mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}.$$

Continua

Ovviamente, $T \subseteq T'$. Inoltre, per ogni $\mathbf{x} \in T$ si ha che:

$$\mathbf{Cx} \leq \mathbf{d} \Rightarrow \forall \lambda \geq \mathbf{0} : \lambda(\mathbf{d} - \mathbf{Cx}) \geq \mathbf{0} \Rightarrow f'(\mathbf{x}) \geq f(\mathbf{x}).$$

Quindi sono soddisfatte le due condizioni che devono essere soddisfatte da un rilassamento.

Infine ...

... notiamo che nel rilassamento lagrangiano rimangono solo i vincoli "facili" e quindi esso può essere risolto in tempo polinomiale, come viene richiesto per il calcolo di un upper bound.

Infine ...

... notiamo che nel rilassamento lagrangiano rimangono solo i vincoli "facili" e quindi esso può essere risolto in tempo polinomiale, come viene richiesto per il calcolo di un upper bound.

Notiamo anche che ad ogni $\lambda \geq 0$ distinto corrisponde un diverso upper bound $u(\lambda)$. Per ottenere il miglior upper bound possibile (ovvero il più piccolo), possiamo risolvere questo ulteriore problema:

$$\min_{\lambda \geq 0} u(\lambda)$$

detto *duale lagrangiano*.

Caso particolare

Scegliendo $\lambda = 0$ abbiamo un caso particolare di rilassamento lagrangiano in cui i vincoli "difficili" del problema vengono semplicemente eliminati.

Osservazione

In alcuni casi i vincoli "difficili" del problema sono vincoli di uguaglianza

$$Cx = d.$$

Osservazione

In alcuni casi i vincoli "difficili" del problema sono vincoli di uguaglianza

$$Cx = d.$$

In tal caso, il rilassamento lagrangiano si definisce nello stesso modo ma i moltiplicatori di Lagrange relativi ai vincoli di uguaglianza non sono vincolati ad assumere solo valori non negativi ma possono assumere anche valori negativi.

Lower bound

Un limite inferiore o *lower bound* per il valore ottimo del nostro problema è un valore LB che soddisfa la seguente proprietà:

$$LB \leq f(x^*) = \max_{x \in S} f(x).$$

Come si calcola?

Se prendiamo un qualsiasi elemento $\bar{x} \in S$ e valutiamo in esso la funzione f , il valore $f(\bar{x})$ è già un lower bound, dal momento che $f(\bar{x}) \leq f(x^*)$.

Come si calcola?

Se prendiamo un qualsiasi elemento $\bar{x} \in S$ e valutiamo in esso la funzione f , il valore $f(\bar{x})$ è già un lower bound, dal momento che $f(\bar{x}) \leq f(x^*)$.

Durante l'esecuzione di un algoritmo branch-and-bound la funzione f viene valutata per molti elementi $y_1, \dots, y_h \in S$ e per ognuno di essi si ha

$$f(y_i) \leq f(x^*) \quad i = 1, \dots, h.$$

Come si calcola?

Se prendiamo un qualsiasi elemento $\bar{x} \in S$ e valutiamo in esso la funzione f , il valore $f(\bar{x})$ è già un lower bound, dal momento che $f(\bar{x}) \leq f(x^*)$.

Durante l'esecuzione di un algoritmo branch-and-bound la funzione f viene valutata per molti elementi $y_1, \dots, y_h \in S$ e per ognuno di essi si ha

$$f(y_i) \leq f(x^*) \quad i = 1, \dots, h.$$

A noi interessa un valore LB il più possibile vicino al valore ottimo del problema. Quindi, poniamo

$$LB = \max\{f(y_i) : i = 1, \dots, h\} \leq f(x^*).$$

Ma ...

... da dove ricaviamo gli elementi di S in cui valutare la funzione f durante l'esecuzione dell'algoritmo?

Ma ...

... da dove ricaviamo gli elementi di S in cui valutare la funzione f durante l'esecuzione dell'algoritmo?

- Se si ha a disposizione un'euristica è buona norma valutare f nel risultato di tale euristica;

Ma ...

... da dove ricaviamo gli elementi di S in cui valutare la funzione f durante l'esecuzione dell'algoritmo?

- Se si ha a disposizione un'euristica è buona norma valutare f nel risultato di tale euristica;
- durante lo stesso calcolo degli upper bound si possono individuare uno o più elementi di S e valutare in essi f .

Ma ...

... da dove ricaviamo gli elementi di S in cui valutare la funzione f durante l'esecuzione dell'algoritmo?

- Se si ha a disposizione un'euristica è buona norma valutare f nel risultato di tale euristica;
- durante lo stesso calcolo degli upper bound si possono individuare uno o più elementi di S e valutare in essi f .
Ad esempio, se si calcola l'upper bound $U(T)$ tramite un rilassamento, nei casi in cui per la soluzione $x' \in T' \supseteq T$ valga anche $x' \in T$, allora si ha anche $x' \in S$ e si può valutare f in x' .

Ma ...

... da dove ricaviamo gli elementi di S in cui valutare la funzione f durante l'esecuzione dell'algoritmo?

- Se si ha a disposizione un'euristica è buona norma valutare f nel risultato di tale euristica;
- durante lo stesso calcolo degli upper bound si possono individuare uno o più elementi di S e valutare in essi f .
Ad esempio, se si calcola l'upper bound $U(T)$ tramite un rilassamento, nei casi in cui per la soluzione $x' \in T' \supseteq T$ valga anche $x' \in T$, allora si ha anche $x' \in S$ e si può valutare f in x' . In altri casi non si ha $x' \in T$ ma con opportune operazioni (quali arrotondamenti o approssimazioni per eccesso/difetto di valori di variabili) si può determinare partendo da $x' \notin T$ una soluzione $\bar{x}' \in T$ (un esempio di ciò lo incontreremo nell'algoritmo branch-and-bound per il problema dello zaino).

Branching

L'operazione di branching consiste nel rimpiazzare un insieme $T \subseteq S$ con una sua partizione T_1, \dots, T_m . Si ricordi che T_1, \dots, T_m formano una partizione di T se

$$T = \cup_{i=1}^m T_i \quad T_i \cap T_j = \emptyset \quad \forall i \neq j.$$

Branching

L'operazione di branching consiste nel rimpiazzare un insieme $T \subseteq S$ con una sua partizione T_1, \dots, T_m . Si ricordi che T_1, \dots, T_m formano una partizione di T se

$$T = \cup_{i=1}^m T_i \quad T_i \cap T_j = \emptyset \quad \forall i \neq j.$$

La partizione può essere rappresentata tramite una struttura ad albero: l'insieme T è un nodo dell'albero da cui partono i rami (da qui il nome branching) verso i nodi della partizione, che vengono anche detti nodi successori o nodi figli del nodo T .

Cancellazione di sottinsiemi

Il punto chiave degli algoritmi di branch-and-bound è la *cancellazione di sottinsiemi*.

Cancellazione di sottinsiemi

Il punto chiave degli algoritmi di branch-and-bound è la *cancellazione di sottinsiemi*.

Supponiamo che per un dato sottinsieme, T_2 ad esempio, si abbia

$$U(T_2) \leq LB.$$

Cancellazione di sottinsiemi

Il punto chiave degli algoritmi di branch-and-bound è la *cancellazione di sottinsiemi*.

Supponiamo che per un dato sottinsieme, T_2 ad esempio, si abbia

$$U(T_2) \leq LB.$$

Ma questo vuol dire che

$$\forall x \in T_2 \quad f(x) \leq U(T_2) \leq LB,$$

e cioè tra tutti gli elementi in T_2 non ne possiamo trovare alcuno con valore di f superiore a LB , ovvero al miglior valore di f osservato fino a questo momento. A questo punto posso *cancellare* il sottinsieme T_2 .

Cancellazione ed enumerazione implicita

La cancellazione equivale ad una *enumerazione implicita*: il confronto tra upper bound $U(T_2)$ del sottinsieme e lower bound LB ci consente di scartare tutti gli elementi in T_2 *senza dover calcolare la funzione f in essi*.

Cancellazione ed enumerazione implicita

La cancellazione equivale ad una *enumerazione implicita*: il confronto tra upper bound $U(T_2)$ del sottinsieme e lower bound LB ci consente di scartare tutti gli elementi in T_2 *senza dover calcolare la funzione f in essi*.

La regola di cancellazione appena introdotta ci fa capire perché vogliamo un valore di upper bound $U(T_2)$ il più vicino possibile al valore ottimo di f sul sottinsieme T_2 e un valore LB il più possibile vicino al valore ottimo del problema:

Cancellazione ed enumerazione implicita

La cancellazione equivale ad una *enumerazione implicita*: il confronto tra upper bound $U(T_2)$ del sottinsieme e lower bound LB ci consente di scartare tutti gli elementi in T_2 *senza dover calcolare la funzione f in essi*.

La regola di cancellazione appena introdotta ci fa capire perché vogliamo un valore di upper bound $U(T_2)$ il più vicino possibile al valore ottimo di f sul sottinsieme T_2 e un valore LB il più possibile vicino al valore ottimo del problema:

in questo modo è più semplice cancellare il sottinsieme tramite la condizione $U(T_2) \leq LB$.

L'algoritmo branch-and-bound

- **Passo 1** Si ponga $\mathcal{C} = \{S\}$ e $\mathcal{Q} = \emptyset$ (l'insieme \mathcal{C} conterrà sempre i sottinsiemi ancora da tenere in considerazione e inizialmente contiene l'intero insieme S , mentre l'insieme \mathcal{Q} , inizialmente vuoto, conterrà tutti i sottinsiemi cancellati). Si ponga $k = 1$. Si calcoli $U(S)$ e si calcoli un valore per LB (eventualmente utilizzando anche i risultati di un'euristica, se disponibile). Se non si dispone di soluzioni ammissibili, si ponga $LB = -\infty$.

L'algoritmo branch-and-bound

- **Passo 1** Si ponga $\mathcal{C} = \{S\}$ e $\mathcal{Q} = \emptyset$ (l'insieme \mathcal{C} conterrà sempre i sottinsiemi ancora da tenere in considerazione e inizialmente contiene l'intero insieme S , mentre l'insieme \mathcal{Q} , inizialmente vuoto, conterrà tutti i sottinsiemi cancellati). Si ponga $k = 1$. Si calcoli $U(S)$ e si calcoli un valore per LB (eventualmente utilizzando anche i risultati di un'euristica, se disponibile). Se non si dispone di soluzioni ammissibili, si ponga $LB = -\infty$.
- **Passo 2 (Selezione di un sottinsieme)** Si selezioni un sottinsieme $T \in \mathcal{C}$. Tra le varie regole di selezione citiamo qui quella di selezionare il sottinsieme T in \mathcal{C} con il valore di upper bound più elevato, cioè

$$U(T) = \max_{Q \in \mathcal{C}} U(Q).$$

- **Passo 3 (Branching)** Si sostituisca l'insieme T in \mathcal{C} con la sua partizione in m_k sottinsiemi T_1, \dots, T_{m_k} , ovvero

$$\mathcal{C} = \mathcal{C} \cup \{T_1, \dots, T_{m_k}\} \setminus \{T\}.$$

- **Passo 3 (Branching)** Si sostituisca l'insieme T in \mathcal{C} con la sua partizione in m_k sottinsiemi T_1, \dots, T_{m_k} , ovvero

$$\mathcal{C} = \mathcal{C} \cup \{T_1, \dots, T_{m_k}\} \setminus \{T\}.$$

- **Passo 4 (Upper bounding)** Si calcoli un upper bound $U(T_i)$, $i = 1, \dots, m_k$ per ogni sottinsieme della partizione.

- **Passo 3 (Branching)** Si sostituisca l'insieme T in \mathcal{C} con la sua partizione in m_k sottinsiemi T_1, \dots, T_{m_k} , ovvero

$$\mathcal{C} = \mathcal{C} \cup \{T_1, \dots, T_{m_k}\} \setminus \{T\}.$$

- **Passo 4 (Upper bounding)** Si calcoli un upper bound $U(T_i)$, $i = 1, \dots, m_k$ per ogni sottinsieme della partizione.
- **Passo 5 (Lower bounding)** Si aggiorni, eventualmente, il valore LB (si ricordi che il valore LB corrisponde sempre al massimo dei valori di f osservati durante l'esecuzione dell'algoritmo).

- **Passo 6 (Cancellazione sottinsiemi)** Si escludano da \mathcal{C} tutti i sottinsiemi Q per cui $U(Q) \leq LB$, ovvero

$$\mathcal{C} = \mathcal{C} \setminus \{Q : U(Q) \leq LB\}.$$

e si trasferiscano tali sottinsiemi in \mathcal{Q} , cioè:

$$\mathcal{Q} = \mathcal{Q} \cup \{Q : U(Q) \leq LB\}.$$

- **Passo 6 (Cancellazione sottinsiemi)** Si escludano da \mathcal{C} tutti i sottinsiemi Q per cui $U(Q) \leq LB$, ovvero

$$\mathcal{C} = \mathcal{C} \setminus \{Q : U(Q) \leq LB\}.$$

e si trasferiscano tali sottinsiemi in \mathcal{Q} , cioè:

$$\mathcal{Q} = \mathcal{Q} \cup \{Q : U(Q) \leq LB\}.$$

- **Passo 7** Se $\mathcal{C} = \emptyset$: stop, il valore LB coincide con il valore ottimo $f(x^*)$. Altrimenti si ponga $k = k + 1$ e si ritorni al Passo 2.

Osservazione

Se $\mathcal{C} = \emptyset$, LB è il valore ottimo del nostro problema (se è pari a $-\infty$, allora $S = \emptyset$).

Osservazione

Se $\mathcal{C} = \emptyset$, LB è il valore ottimo del nostro problema (se è pari a $-\infty$, allora $S = \emptyset$).

Questa affermazione è una conseguenza del fatto che, nel momento in cui $\mathcal{C} = \emptyset$, tutti i sottinsiemi cancellati fino a quel momento, cioè la collezione \mathcal{Q} di sottinsiemi, formano una *partizione dell'intero insieme S* .

Osservazione

Se $\mathcal{C} = \emptyset$, LB è il valore ottimo del nostro problema (se è pari a $-\infty$, allora $S = \emptyset$).

Questa affermazione è una conseguenza del fatto che, nel momento in cui $\mathcal{C} = \emptyset$, tutti i sottinsiemi cancellati fino a quel momento, cioè la collezione \mathcal{Q} di sottinsiemi, formano una *partizione dell'intero insieme* S .

Quindi tra di essi ve ne è certamente uno, indicato con $T^* \in \mathcal{Q}$, che contiene x^* . Ma poiché T^* è stato cancellato si dovrà avere

$$f(x^*) \leq U(T^*) \leq LB \leq f(x^*),$$

da cui segue immediatamente che $LB = f(x^*)$.

Modifiche per problemi di minimo

- ad un sottinsieme $Q \subseteq S$ dovrà essere associato un valore di lower bound $L(Q)$;

Modifiche per problemi di minimo

- ad un sottinsieme $Q \subseteq S$ dovrà essere associato un valore di lower bound $L(Q)$;
- al posto del valore LB avremo un valore UB con la proprietà

$$UB \geq f(x^*) = \min_{x \in S} f(x).$$

Il valore UB sarà il minimo tra i valori osservati della funzione obiettivo in punti della regione ammissibile S .

Modifiche per problemi di minimo

- ad un sottinsieme $Q \subseteq S$ dovrà essere associato un valore di lower bound $L(Q)$;
- al posto del valore LB avremo un valore UB con la proprietà

$$UB \geq f(x^*) = \min_{x \in S} f(x).$$

Il valore UB sarà il minimo tra i valori osservati della funzione obiettivo in punti della regione ammissibile S .

- Il sottinsieme Q viene cancellato se è vero che $L(Q) \geq UB$.

Modifiche per problemi di minimo

- ad un sottinsieme $Q \subseteq S$ dovrà essere associato un valore di lower bound $L(Q)$;
- al posto del valore LB avremo un valore UB con la proprietà

$$UB \geq f(x^*) = \min_{x \in S} f(x).$$

Il valore UB sarà il minimo tra i valori osservati della funzione obiettivo in punti della regione ammissibile S .

- Il sottinsieme Q viene cancellato se è vero che $L(Q) \geq UB$.
- Al Passo 2 della procedura di branch-and-bound si seleziona un nodo con lower bound più piccolo, ovvero un nodo T tale che

$$L(T) = \min_{Q \in \mathcal{C}} L(Q).$$

Branch-and-bound per PLI

Un esempio di problema di PLI:

$$P_0 : \quad \max \quad x_1 + 3x_2$$

$$(u_1) \quad x_1 \geq \frac{1}{2}$$

$$(u_2) \quad -5x_1 + 3x_2 \leq 5$$

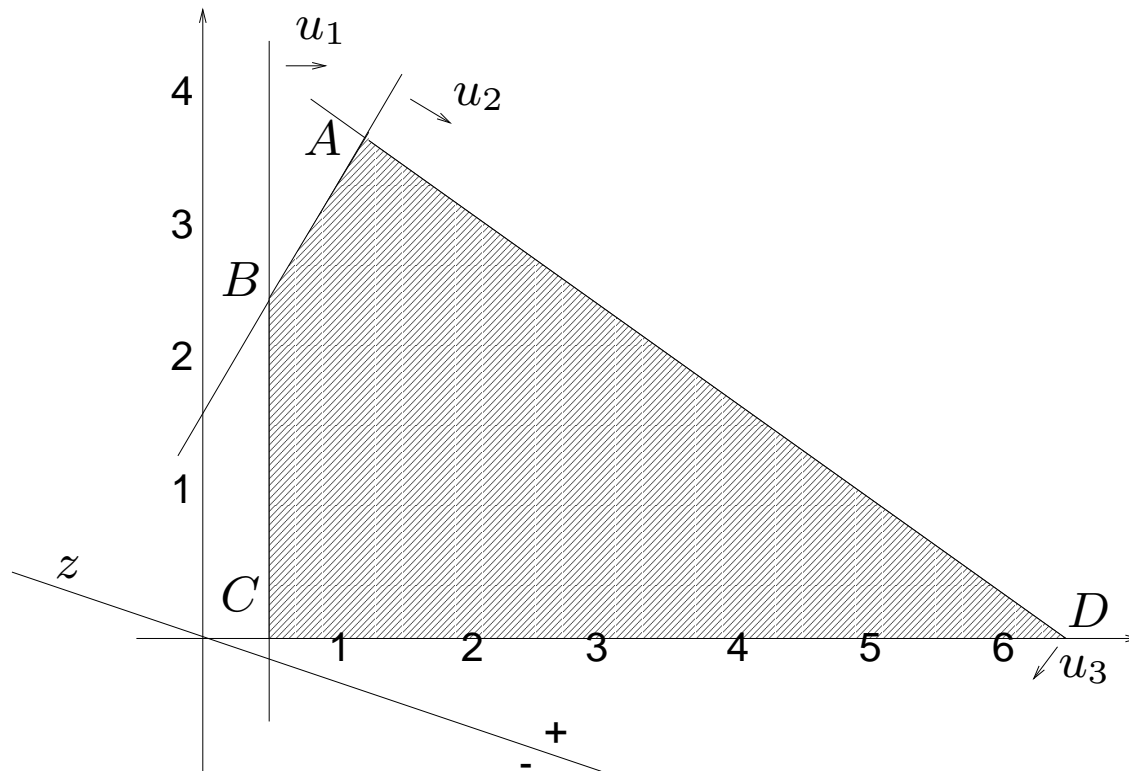
$$(u_3) \quad x_1 + \frac{7}{5}x_2 \leq \frac{13}{2}$$

$$x_1, x_2 \geq 0$$

$$x_1, x_2 \in I$$

Continua

Risoluzione (grafica) del rilassamento lineare P'_0 di P_0 .



Continua

$$S_{ott} = \{A\} \quad A = \left(\frac{5}{4}, \frac{15}{4}\right)$$

Continua

$$S_{ott} = \{A\} \quad A = \left(\frac{5}{4}, \frac{15}{4}\right)$$

Valore ottimo $= U(P_0) = \frac{50}{4}$

Continua

$$S_{ott} = \{A\} \quad A = \left(\frac{5}{4}, \frac{15}{4}\right)$$

Valore ottimo= $U(P_0) = \frac{50}{4}$

$U(P_0)$ = **upper bound** o limitazione superiore del valore ottimo di P_0

Divide et impera: Branching

Suddivisione del problema originario in due sottoproblemi P_1 e P_2 . Come?

Divide et impera: Branching

Suddivisione del problema originario in due sottoproblemi P_1 e P_2 . Come?

Seleziono, *secondo una determinata regola*, una variabile x_i con valore non intero x_i^* nella soluzione ottima del rilassamento lineare di P_0 e:

Divide et impera: Branching

Suddivisione del problema originario in due sottoproblemi P_1 e P_2 . Come?

Seleziono, *secondo una determinata regola*, una variabile x_i con valore non intero x_i^* nella soluzione ottima del rilassamento lineare di P_0 e:

- creo P_1 aggiungendo ai vincoli di P_0 , il vincolo $x_i \leq \lfloor x_i^* \rfloor$;

Divide et impera: Branching

Suddivisione del problema originario in due sottoproblemi P_1 e P_2 . Come?

Seleziono, *secondo una determinata regola*, una variabile x_i con valore non intero x_i^* nella soluzione ottima del rilassamento lineare di P_0 e:

- creo P_1 aggiungendo ai vincoli di P_0 , il vincolo $x_i \leq \lfloor x_i^* \rfloor$;
- creo P_2 aggiungendo ai vincoli di P_0 , il vincolo $x_i \geq \lfloor x_i^* \rfloor + 1$

Divide et impera: Branching

Suddivisione del problema originario in due sottoproblemi P_1 e P_2 . Come?

Seleziono, *secondo una determinata regola*, una variabile x_i con valore non intero x_i^* nella soluzione ottima del rilassamento lineare di P_0 e:

- creo P_1 aggiungendo ai vincoli di P_0 , il vincolo $x_i \leq \lfloor x_i^* \rfloor$;
- creo P_2 aggiungendo ai vincoli di P_0 , il vincolo $x_i \geq \lfloor x_i^* \rfloor + 1$

Esistono molte regole di scelta della variabile. Qui ne useremo una semplicissima non preoccupandoci dell'efficienza: seleziona quella con valore non intero nel rilassamento lineare di P_0 con indice più piccolo.

Nell'esempio

Sottoproblema P_1

$$P_1 : \quad \max \quad x_1 + 3x_2$$

$$(u_1) \quad x_1 \geq \frac{1}{2}$$

$$(u_2) \quad -5x_1 + 3x_2 \leq 5$$

$$(u_3) \quad x_1 + \frac{7}{5}x_2 \leq \frac{13}{2}$$

$$(u'_4) \quad x_1 \leq \left\lfloor \frac{5}{4} \right\rfloor = 1$$

$$x_1, x_2 \geq 0$$

$$x_1, x_2 \in I$$

Nell'esempio

Sottoproblema P_2

$$P_2 : \quad \max \quad x_1 + 3x_2$$

$$(u_1) \quad x_1 \geq \frac{1}{2}$$

$$(u_2) \quad -5x_1 + 3x_2 \leq 5$$

$$(u_3) \quad x_1 + \frac{7}{5}x_2 \leq \frac{13}{2}$$

$$(u_4'') \quad x_1 \geq \left\lfloor \frac{5}{4} \right\rfloor + 1 = 2$$

$$x_1, x_2 \geq 0$$

$$x_1, x_2 \in I$$

Lower Bound

LB = **Lower bound** o limitazione inferiore del valore ottimo di P_0 . È un qualsiasi valore che soddisfa:

$$LB \leq \text{opt}(P_0) = \text{valore ottimo } P_0$$

Lower Bound

$LB = \text{Lower bound}$ o limitazione inferiore del valore ottimo di P_0 . È un qualsiasi valore che soddisfa:

$$LB \leq \text{opt}(P_0) = \text{valore ottimo } P_0$$

Come si ottiene?

Lower Bound

LB = **Lower bound** o limitazione inferiore del valore ottimo di P_0 . È un qualsiasi valore che soddisfa:

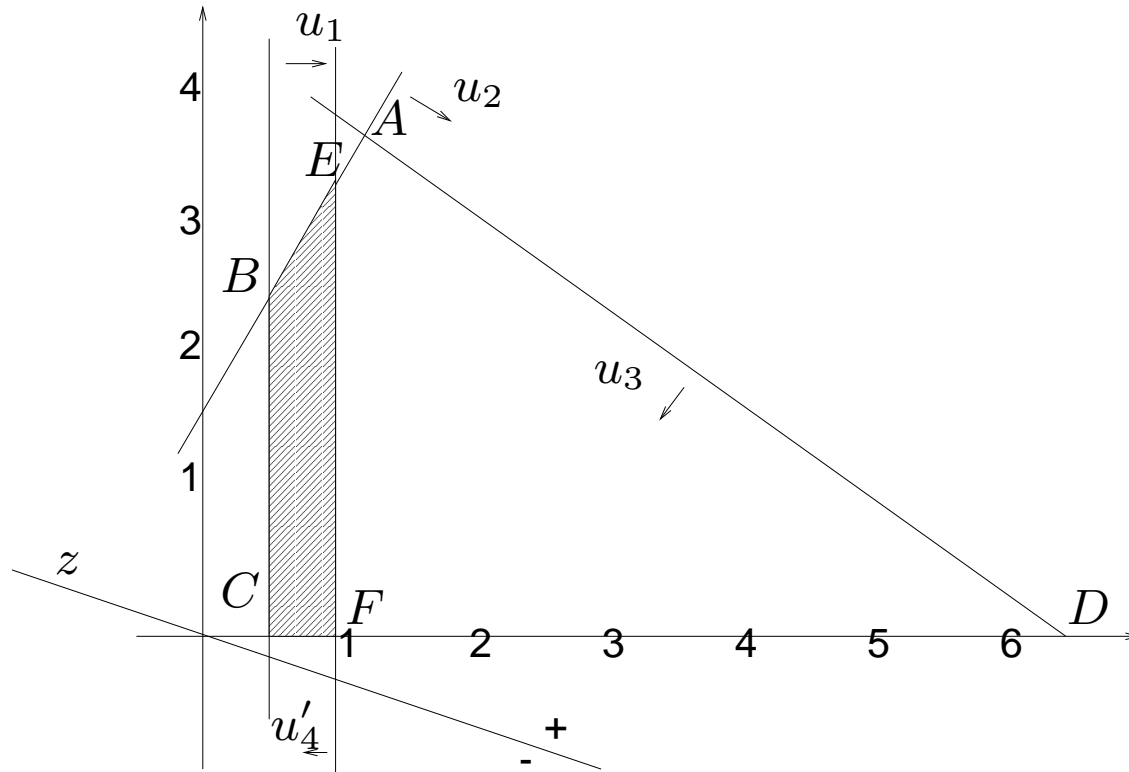
$$LB \leq \text{opt}(P_0) = \text{valore ottimo } P_0$$

Come si ottiene?

Se durante l'esecuzione dell'algoritmo (tipicamente durante la risoluzione dei rilassamenti lineari di P_0 e dei suoi sottoproblemi) si osserva il valore della funzione obiettivo $\mathbf{c}\mathbf{x}$ nei punti $\mathbf{y}_1, \dots, \mathbf{y}_h \in Z_a$, ciascuno di questi valori è utilizzabile come lower bound, in quanto: $\mathbf{c}\mathbf{y}_i \leq \text{opt}(P_0)$ per $i = 1, \dots, h$. Per avere un valore di lower bound il più vicino possibile a $\text{opt}(P_0)$ si pone:

$$LB = \max\{\mathbf{c}\mathbf{y}_i : i = 1, \dots, h\}$$

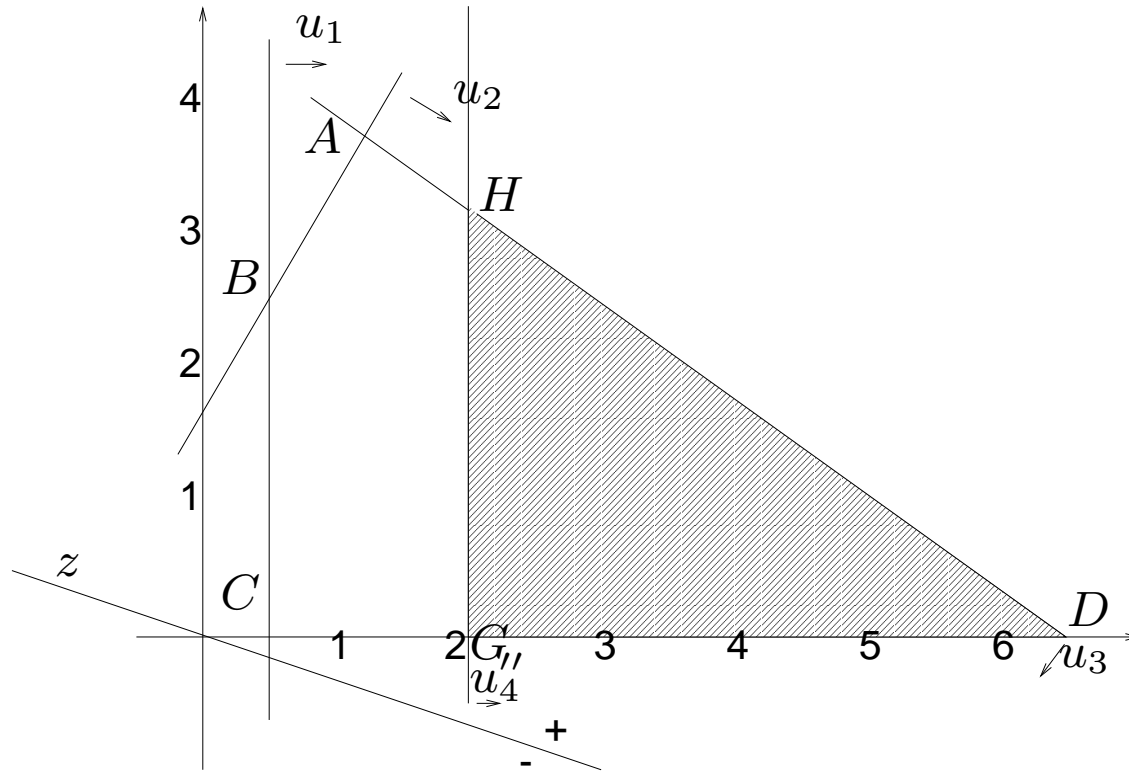
Rilassamento lineare di P_1



$$S_{ott} = \{E\}, E = (1, 10/3)$$

$$U(P_1) = 11$$

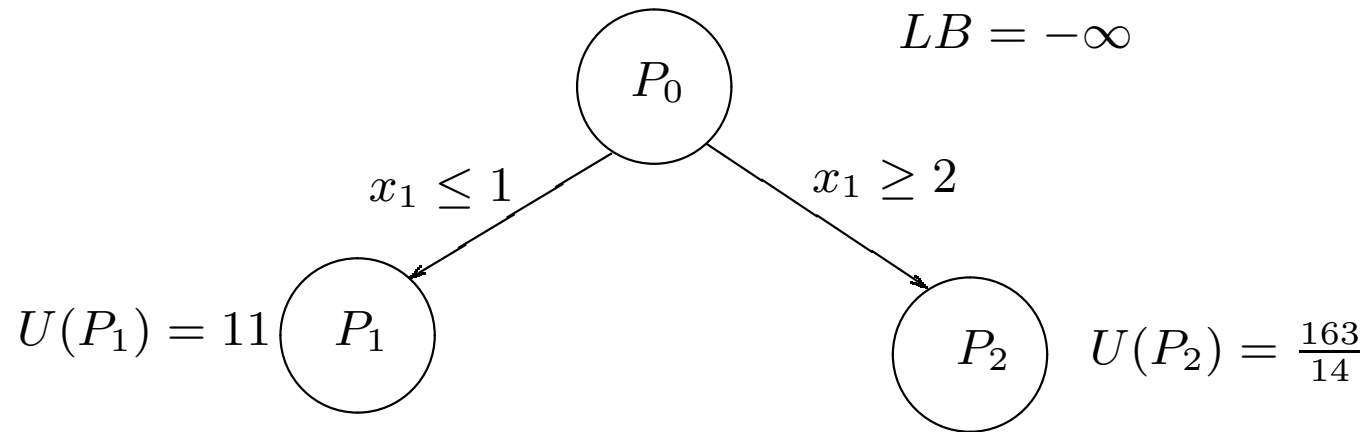
Rilassamento lineare di P_2



$$S_{ott} = \{H\}, H = (2, 45/14)$$

$$U(P_2) = 163/14$$

Albero di branch-and-bound



Ulteriore suddivisione

Dopo aver suddiviso il problema P_0 nei due sottoproblemi P_1 e P_2 , possiamo selezionare uno dei due sottoproblemi.

Ulteriore suddivisione

Dopo aver suddiviso il problema P_0 nei due sottoproblemi P_1 e P_2 , possiamo selezionare uno dei due sottoproblemi.

Quale nodo selezionare?

Ulteriore suddivisione

Dopo aver suddiviso il problema P_0 nei due sottoproblemi P_1 e P_2 , possiamo selezionare uno dei due sottoproblemi.

Quale nodo selezionare?

Diverse regole possibili. Qui ne vedremo una soltanto:
seleziona un sottoproblema/nodo foglia con upper bound massimo.

Perché?

Ulteriore suddivisione

Dopo aver suddiviso il problema P_0 nei due sottoproblemi P_1 e P_2 , possiamo selezionare uno dei due sottoproblemi.

Quale nodo selezionare?

Diverse regole possibili. Qui ne vedremo una soltanto:
seleziona un sottoproblema/nodo foglia con upper bound massimo.

Perché?

Ci si aspetta di trovare più facilmente la soluzione ottima del problema in un sottoproblema con un valore di upper bound elevato.

Nell'esempio

Nel nostro esempio selezioniamo P_2 .

Nell'esempio

Nel nostro esempio selezioniamo P_2 .

P_2 suddiviso in due sottoproblemi P_3 e P_4 . Come?

Nell'esempio

Nel nostro esempio selezioniamo P_2 .

P_2 suddiviso in due sottoproblemi P_3 e P_4 . Come?

Esattamente come abbiamo fatto per P_0 : facendo branching su una variabile con valore non intero nella soluzione ottima del rilassamento lineare di P_2 .

Il sottoproblema P_3

$$P_3 : \quad \max \quad x_1 + 3x_2$$

$$(u_1) \quad x_1 \geq \frac{1}{2}$$

$$(u_2) \quad -5x_1 + 3x_2 \leq 5$$

$$(u_3) \quad x_1 + \frac{7}{5}x_2 \leq \frac{13}{2}$$

$$(u_4'') \quad x_1 \geq 2$$

$$(u_5') \quad x_2 \leq \left\lfloor \frac{45}{14} \right\rfloor = 3$$

$$x_1, x_2 \geq 0$$

$$x_1, x_2 \in I$$

Il sottoproblema P_4

$$P_4 : \quad \max \quad x_1 + 3x_2$$

$$(u_1) \quad x_1 \geq \frac{1}{2}$$

$$(u_2) \quad -5x_1 + 3x_2 \leq 5$$

$$(u_3) \quad x_1 + \frac{7}{5}x_2 \leq \frac{13}{2}$$

$$(u_4'') \quad x_1 \geq 2$$

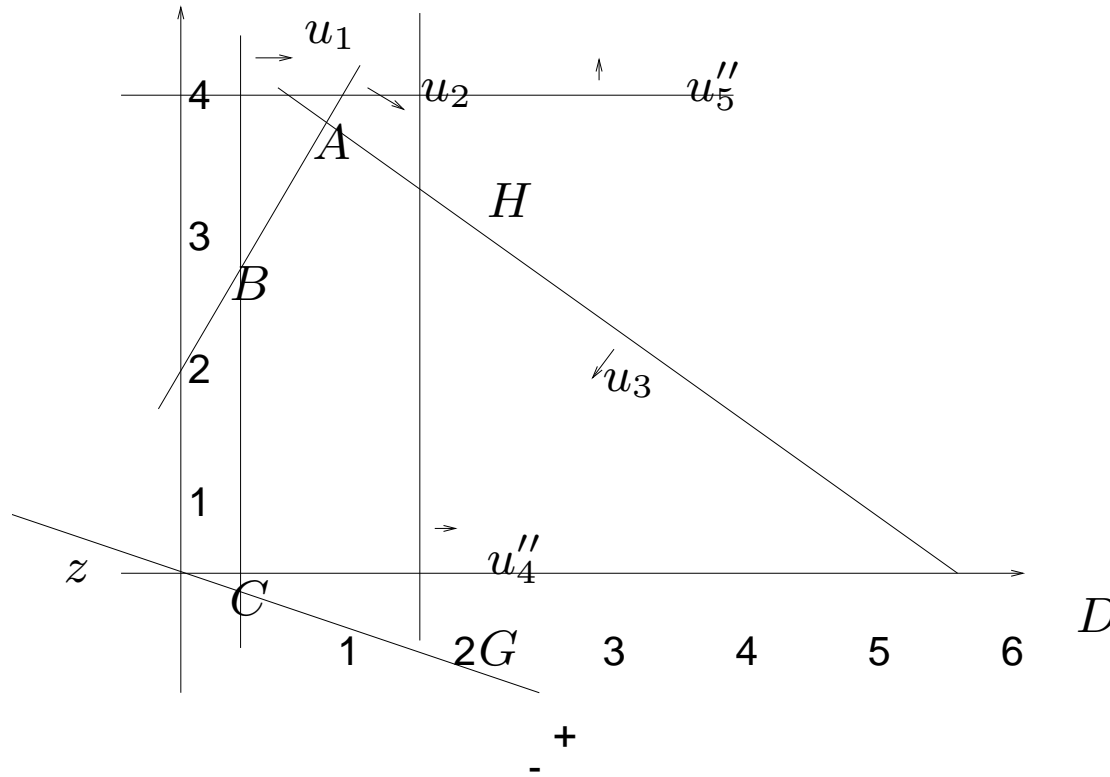
$$(u_5'') \quad x_2 \geq \left\lfloor \frac{45}{14} \right\rfloor + 1 = 4$$

$$x_1, x_2 \geq 0$$

$$x_1, x_2 \in I$$

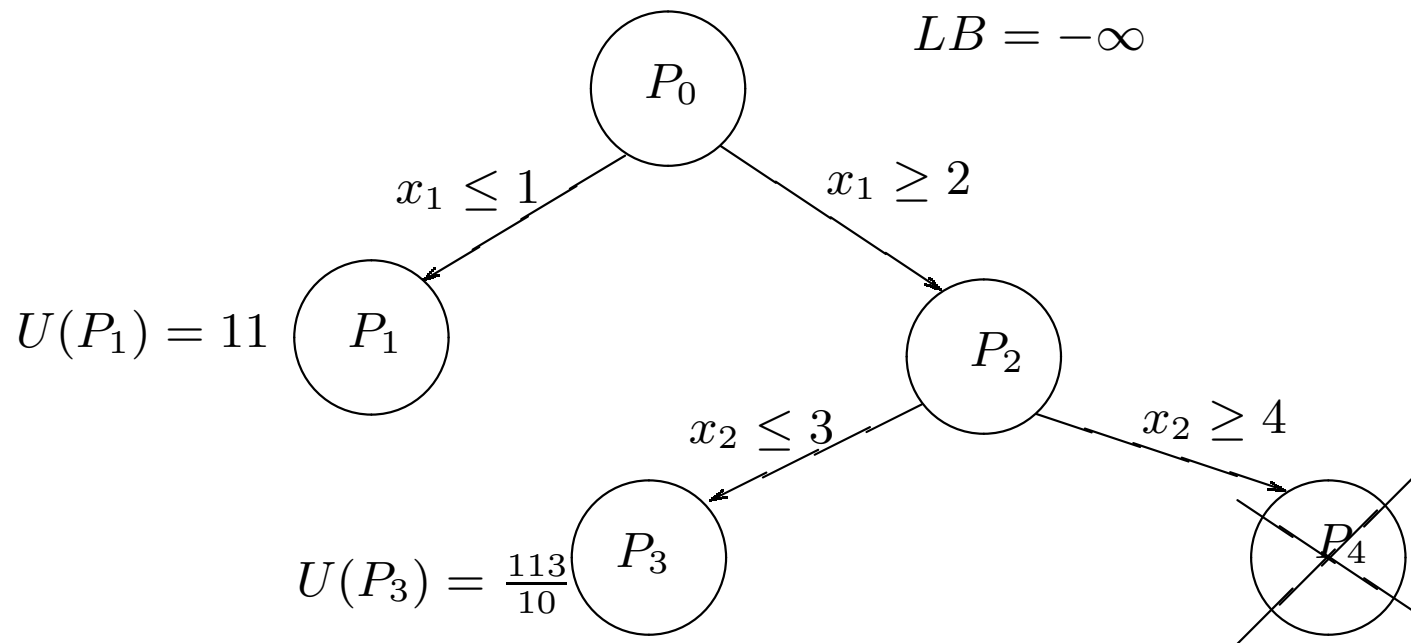
$$U(P_3) = 113/10$$

Il rilassamento lineare di P_4



$S_a = \emptyset$, quindi P_4 é risolto $\rightarrow P_4$ ha regione ammissibile vuota.

Albero di branch-and-bound



Continua

Al momento: il problema iniziale P_0 é suddiviso in tre sottoproblemi P_1 , P_3 , P_4 , di cui P_4 é già risolto (cancellazione del nodo nell'albero).

Continua

Al momento: il problema iniziale P_0 é suddiviso in tre sottoproblemi P_1 , P_3 , P_4 , di cui P_4 é già risolto (cancellazione del nodo nell'albero).

Ora: seleziono il nodo foglia/sottoproblema non ancora cancellato con upper bound maggiore (nell'esempio P_3) e lo suddivido in due sottoproblemi (nell'esempio P_5 e P_6).

Il sottoproblema P_5

$$P_5 : \quad \max \quad x_1 + 3x_2$$

$$(u_1) \quad x_1 \geq \frac{1}{2}$$

$$(u_2) \quad -5x_1 + 3x_2 \leq 5$$

$$(u_3) \quad x_1 + \frac{7}{5}x_2 \leq \frac{13}{2}$$

$$(u_4'') \quad x_1 \geq 2$$

$$(u_5') \quad x_2 \leq 3$$

$$(u_6') \quad x_1 \leq \left\lfloor \frac{23}{10} \right\rfloor = 2$$

$$x_1, x_2 \geq 0$$

$$x_1, x_2 \in I$$

Il sottoproblema P_6

$$P_6 : \quad \max \quad x_1 + 3x_2$$

$$(u_1) \quad x_1 \geq \frac{1}{2}$$

$$(u_2) \quad -5x_1 + 3x_2 \leq 5$$

$$(u_3) \quad x_1 + \frac{7}{5}x_2 \leq \frac{13}{2}$$

$$(u_4'') \quad x_1 \geq 2$$

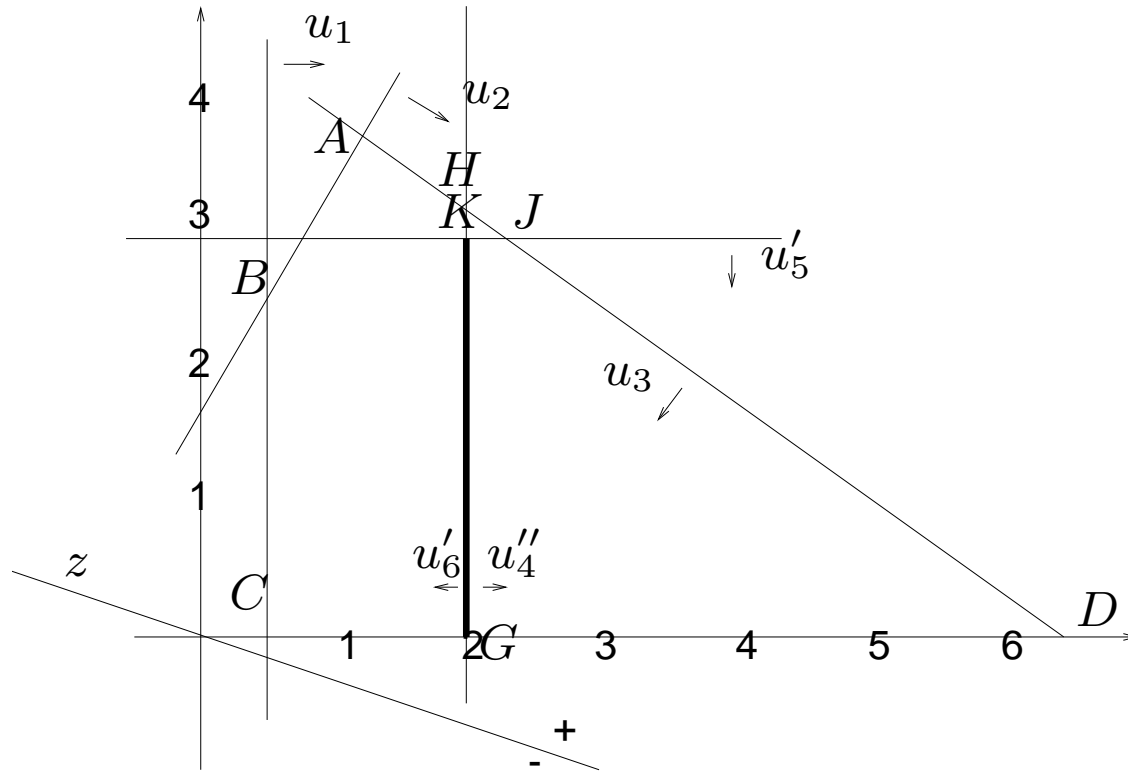
$$(u_5') \quad x_2 \leq 3$$

$$(u_6'') \quad x_1 \geq \left\lfloor \frac{23}{10} \right\rfloor + 1 = 3$$

$$x_1, x_2 \geq 0$$

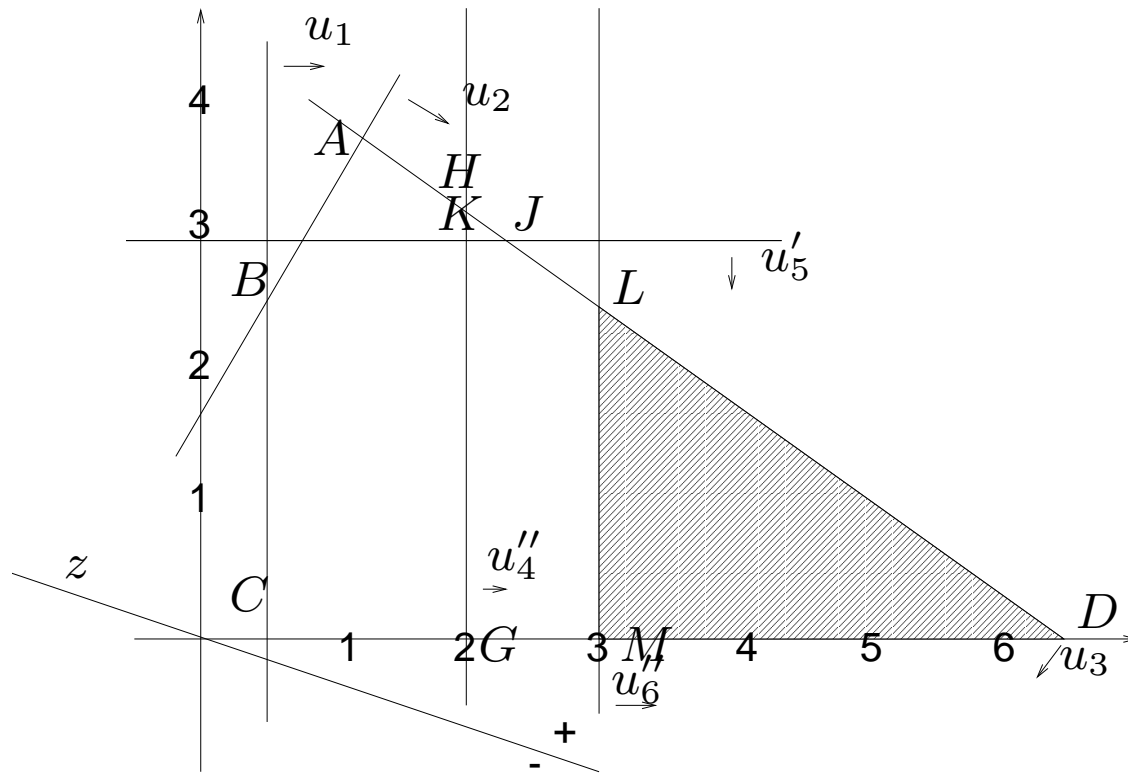
$$x_1, x_2 \in I$$

Il rilassamento lineare di P_5



$S_{ott} = \{K\}$, $K = (2, 3)$: coordinate intere, quindi K risolve anche P_5 ! Valore ottimo di $P_5 = 11$
Posso aggiornare $LB \rightarrow LB = 11$

Il rilassamento lineare di P_6



$$S_{ott} = \{L\}, L = (3, 5/2)$$

$$U(P_6) = 21/2$$

Cancellazione nodi

Oltre ai nodi/sottoproblemi “risolti”, posso anche cancellare nodi foglia/sottoproblemi “non risolti”:

cancella un nodo P_i se $U(P_i) \leq LB$

Cancellazione nodi

Oltre ai nodi/sottoproblemi “risolti”, posso anche cancellare nodi foglia/sottoproblemi “non risolti”:

cancella un nodo P_i se $U(P_i) \leq LB$

In tal caso tutte le soluzioni in P_i sono non migliori rispetto alla migliore osservata sino ad ora (valore LB) e quindi non vale la pena esplorarle.

Cancellazione nodi

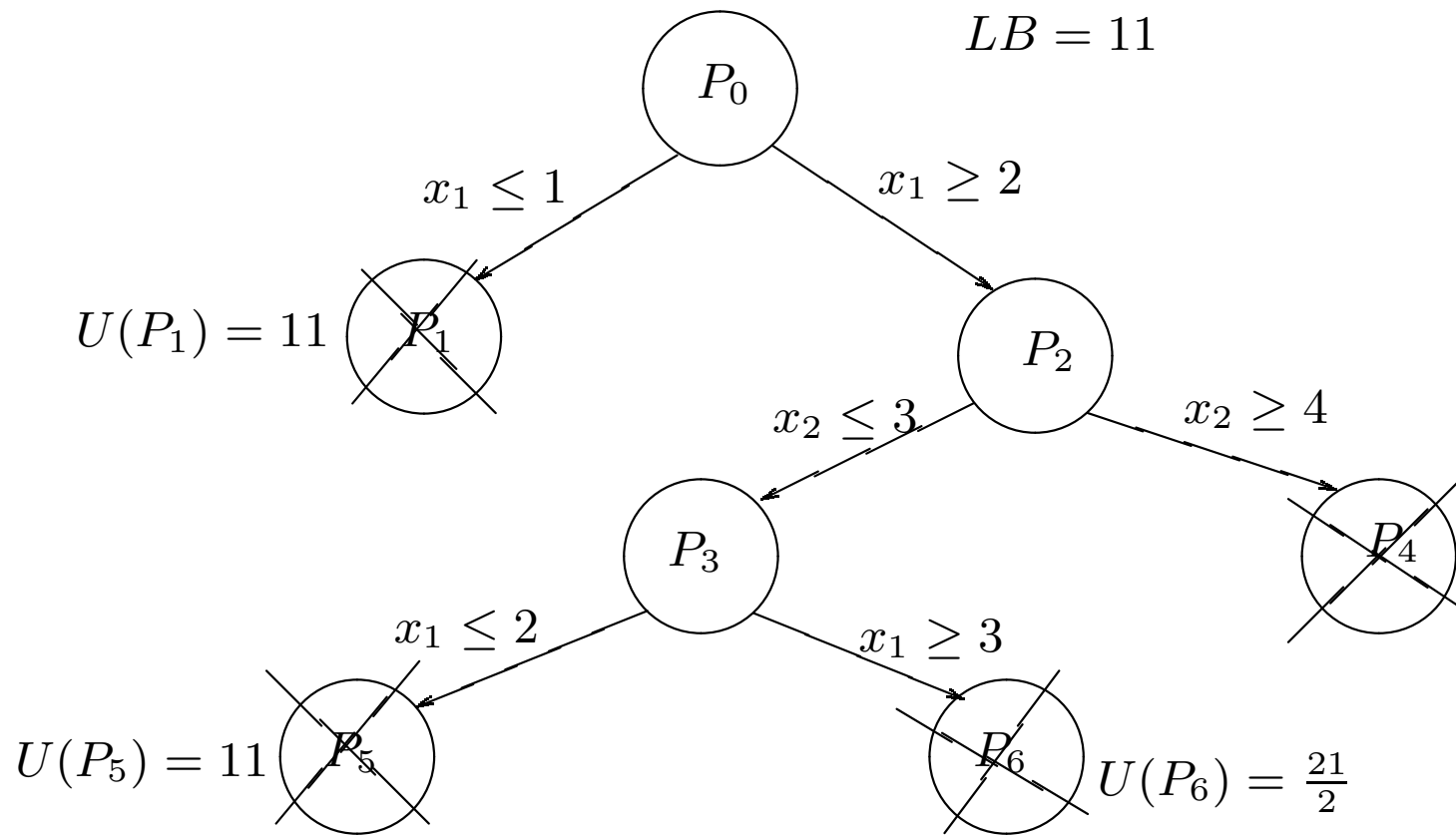
Oltre ai nodi/sottoproblemi “risolti”, posso anche cancellare nodi foglia/sottoproblemi “non risolti”:

cancella un nodo P_i se $U(P_i) \leq LB$

In tal caso tutte le soluzioni in P_i sono non migliori rispetto alla migliore osservata sino ad ora (valore LB) e quindi non vale la pena esplorarle.

Nel nostro esempio, cancella P_1 e P_6 .

Albero di branch-and-bound



Regola di arresto

STOP quando *tutti* i nodi foglia sono stati cancellati.

Regola di arresto

STOP quando *tutti* i nodi foglia sono stati cancellati.

Al momento dell'arresto LB restituisce il valore ottimo di P_0 .

Regola di arresto

STOP quando *tutti* i nodi foglia sono stati cancellati.

Al momento dell'arresto LB restituisce il valore ottimo di P_0 .

Se $LB = -\infty$, allora P_0 ha regione ammissibile vuota.

Algoritmo generale

- **Inizializzazione** Inizializza \mathcal{F} (insieme nodi foglia *non cancellati*) con P_0 , $\mathcal{F} = \{P_0\}$. Risolvi il rilassamento lineare P'_0 di P_0 e sia $(x_1^*(P_0), \dots, x_n^*(P_0))$ la soluzione ottima trovata e $U(P_0)$ il corrispondente valore ottimo. **Se** tutti i valori $x_i^*(P_0)$, $i = 1, \dots, n$ sono interi, **STOP**: la soluzione $(x_1^*(P_0), \dots, x_n^*(P_0))$ é soluzione ottima anche per P_0 ed il valore ottimo di P_0 é pari a $U(P_0)$. **Altrimenti** si ponga $LB = -\infty$ e si vada al Passo 1.

Algoritmo generale

- **Inizializzazione** Inizializza \mathcal{F} (insieme nodi foglia *non cancellati*) con P_0 , $\mathcal{F} = \{P_0\}$. Risolvi il rilassamento lineare P'_0 di P_0 e sia $(x_1^*(P_0), \dots, x_n^*(P_0))$ la soluzione ottima trovata e $U(P_0)$ il corrispondente valore ottimo. **Se** tutti i valori $x_i^*(P_0)$, $i = 1, \dots, n$ sono interi, **STOP**: la soluzione $(x_1^*(P_0), \dots, x_n^*(P_0))$ é soluzione ottima anche per P_0 ed il valore ottimo di P_0 é pari a $U(P_0)$. **Altrimenti** si ponga $LB = -\infty$ e si vada al Passo 1.
- **Passo 1** Seleziona in \mathcal{F} un nodo $Q \in \mathcal{F}$ tale che

$$U(Q) = \max_{P \in \mathcal{F}} U(P)$$

Sia $(x_1^*(Q), \dots, x_n^*(Q))$ una soluzione ottima del rilassamento lineare Q' di Q .

Continua

- **Passo 2** Seleziona una variabile $x_i^*(Q)$ a valore non intero (quella con indice minimo).

Continua

- **Passo 2** Seleziona una variabile $x_i^*(Q)$ a valore non intero (quella con indice minimo).
- **Passo 3** Rimuovi il nodo Q da \mathcal{F} (cioé $\mathcal{F} = \mathcal{F} \setminus \{Q\}$) e suddividi Q in due nuovi nodi Q_1 e Q_2 ottenuti aggiungendo ai vincoli di Q rispettivamente il vincolo $x_i \leq \lfloor x_i^*(Q) \rfloor$ (in Q_1) ed il vincolo $x_i \geq \lfloor x_i^*(Q) \rfloor + 1$ (in Q_2).

Continua

- **Passo 4** Per ciascuno dei due nuovi nodi Q_i , $i = 1, 2$, risolvi il rilassamento lineare Q'_i . Se Q'_i ha regione ammissibile vuota, cancella il nodo Q_i e non lo si aggiunge a \mathcal{F} . Altrimenti se la soluzione ottima $(x_1^*(Q_i), \dots, x_n^*(Q_i))$ é a coordinate intere si cancelli il nodo Q_i senza aggiungerlo a \mathcal{F} ed inoltre, se $U(Q_i) > LB$ si ponga

$$LB = U(Q_i) \quad \text{e} \quad y_1 = x_1^*(Q_i), \dots, y_n = x_n^*(Q_i).$$

Altrimenti, se la soluzione ottima di Q'_i non é a coordinate intere, si aggiunga Q_i a \mathcal{F} , ovvero si ponga $\mathcal{F} = \mathcal{F} \cup \{Q_i\}$.

Continua

- **Passo 5** Si cancellino tutti i nodi in \mathcal{F} con valore dell'upper bound non superiore a LB , ovvero si ponga

$$\mathcal{F} = \mathcal{F} \setminus \{P \in \mathcal{F} : U(P) \leq LB\}.$$

Continua

- **Passo 5** Si cancellino tutti i nodi in \mathcal{F} con valore dell'upper bound non superiore a LB , ovvero si ponga

$$\mathcal{F} = \mathcal{F} \setminus \{P \in \mathcal{F} : U(P) \leq LB\}.$$

- **Passo 6** **Se** $\mathcal{F} = \emptyset$, **STOP**: se LB ha valore pari a $-\infty$, allora il problema P_0 non ha soluzioni ammissibili, altrimenti il valore LB é il valore ottimo del problema P_0 e (y_1, \dots, y_n) é una soluzione ottima di tale problema. **Altrimenti** si ritorni al Passo 1.