Ottimizzazione Combinatoria
Programmazione lineare intera

Andrea Grosso
Dipartimento di Informatica
Università di Torino
grosso@di.unito.it – 011-6706824

Sommario

Il problema delo zaino 0/1

Branch and bound

Soluzioni ammissibili

Criteri di dominanza

Programmazione dinamica

Sommario

Il problema delo zaino 0/1

Branch and bound

Soluzioni ammissibili

Criteri di dominanza

Programmazione dinamica

(Binary) Knapsack problem

Enunciato

Dato un set di oggetti $N = \{1, 2, \dots, n\}$ con pesi $w_1, w_2, \dots, w_n \geq 0$ e profitti $p_1, p_2, \dots, p_n \geq 0$, e una capacità $b > 0$, determinare un subset $S \subseteq N$ tale che

$$w(S) = \sum_{i \in S} w_i \leq b \qquad \text{e}$$

$$p(S) = \sum_{i \in S} p_i \qquad \text{massimo possibile.}$$

Programma lineare intero

$$\begin{aligned} \max \;\; z = & \sum_{i=1}^{n} p_i x_i \\ \text{soggetto a} \;\; & \sum_{i=1}^{n} w_i x_i \leq b \\ & x_i \in \{0, 1\} \qquad i = 1, \dots, n. \end{aligned} \tag{KP}$$

(Binary) Knapsack problem

Motivazioni

- Problema-base per consumo di risorse.
- NP-completo.
- Si presenta come sottoproblema in svariate situazioni di natura algoritmica.

Approcci risolutivi

- PLI (CPLEX va molto bene).
- Branch and bound ("ad hoc").
- Programmazione dinamica.

Sommario

Rilassamento

$$
\begin{aligned}
\max \ z = & \sum_{i=1}^{n} p_i x_i \\
\text{soggetto a } & \sum_{i=1}^{n} w_i x_i \le b \\
& 0 \le x_i \le 1 \qquad i = 1, \ldots, n.
\end{aligned}
\tag{CKP}
$$

- Risolvibile con simplesso.
- Procedura più efficiente.

Rilassamento

Ipotesi. $\frac{p_1}{w_1} \geq \frac{p_2}{w_2} \geq \cdots \geq \frac{p_n}{w_n}$.
Osservazione. In ogni soluzione ottima di (CKP) risulta

$$\sum_{i=1}^{n} w_i x_i = b.$$

Proprietà

In almeno una soluzione ottima di (CKP),

$$x_j > 0 \implies x_1 = x_2 = \cdots = x_{j-1} = 1.$$

Corollario

Esiste una soluzione ottima di (CKP) dove al più una $x_s \in (0, 1)$ e ha la forma

$$x_1 = x_2 = \cdots = x_{s-1} = 1, x_s \in (0, 1), x_{s+1} = x_{s+2} = \cdots = x_n = 0.$$

Rilassamento

Dimostrazione

$\mathbf{x}^* = (x_1^*, x_2^*, \ldots, x_n^*)$ ottimo di (CKP) con

$$x_j > 0, x_i < 1, \text{per qualche } i < j.$$

Si trovano $\delta_i, \delta_j > 0$ tali che

$$\mathbf{x}' =: \begin{cases} x'_k = x^*_k & k \neq i, j, \\ x'_i = x^*_i + \delta_i, & \\ x'_j = x^*_j - \delta_j, & \end{cases} \qquad \text{sia ammissibile.}$$

$$\delta_i = \min\left\{(1 - x_i^*), \frac{w_j}{w_i} x_j^*\right\}.$$

$$\delta_j = \frac{w_i}{w_j}\delta_i$$

$$z(\mathbf{x}') - z(\mathbf{x}^*) \geq 0.$$

Rilassamento

Corollario (riscritto)

L'ottimo del problema rilassato (CKP) è dato da

$$\begin{cases} x_i^* = 1, & i = 1, \dots, s-1 \\ x_s^* = \dfrac{b - \sum_{k=1}^{s-1} w_k}{w_s}, & \\ x_i^* = 0, & i = s+1, \dots, n. \end{cases}$$

dove l'*indice critico* s è il più piccolo per cui risulta $\sum_{k=1}^{s} +1 w_i > b$.

Rilassamento

Upper Bound

Poni $b' := b, k := 1$;
while $b' \geq w_k$ **do**
    Poni $x_k^* = 1,\ b' := b' - w_k,\ k := k + 1$;
end while
if $b' > 0$ **then**
    Poni $s := k$;
    Poni $x_s^* = \frac{b'}{w_s}$;
    Poni $x_{s+1}^*, \dots, x_n^* = 0$;
else
    Poni $x_k, \dots, x_n = 0$;
end if
return x, $\sum_{i=1}^{n} p_i x_i$;

Schema di branch

Variabile critica x_s

$$x_s = 1$$
$$x_s = 0$$

Ad ogni nodo si hanno due liste

$$N_1 = \{\text{indici di variabili fissate a } 1\}$$
$$N_0 = \{\text{indici di variabili fissate a } 0\}$$

Il rilassamento è

$$\begin{aligned} \max \;\; z = & p(N_1) + \sum_{i \in N \setminus (N_0 \cup N_1)} p_i x_i \\ \text{soggetto a } & \sum_{i \in N \setminus (N_0 \cup N_1)} w_i x_i \leq [b - w(N_1)] \\ & x_i \in [0,1] \qquad i \in N \setminus (N_0 \cup N_1) \end{aligned}$$

Schema di branch

Chiusura nodi

Un nodo k dell'albero di branch viene chiuso se:

- $\mathrm{UB}_k \leq \hat{z}$ (dove $\hat{z}$ = best integer)
- $w(N_1) > b$ (nodo privo di soluzioni)
- rilassamento con soluzione intera.

Sommario

Euristica

Lower Bound

Poni $b' := b$;
for $i \in N \setminus (N_0 \cup N_1)$ **do**
  if $w_i \le b'$ **then**
    Poni $x_i = 1$, $b' := b' - w_i$;
  else
    Poni $x_i = 0$;
  end if
end for

Si può effettuare ad ogni nodo, “in simultanea” con il calcolo dell’upper bound.

Sommario

Il problema delo zaino 0/1

Branch and bound

Soluzioni ammissibili

Criteri di dominanza

Programmazione dinamica

Dominanza

Proprietà

In ogni soluzione ottima $\mathbf{x}^* = (x_1^*, \ldots, x_n^*)$, per ogni coppia di oggetti i, j che soddisfa le condizioni

$$w_i \leq w_j, \;\; p_i \geq p_j$$

con almeno una disuguaglianza stretta, vale che

$$x_j^* = 1 \implies x_i^* = 1.$$

Variable fixing

Se ad un nodo risulta $b - w(N_1) < w_i$ allora $x_i = 0$ in ogni soluzione intera.

Sommario

Programmazione dinamica

Problema dello zaino

$$\begin{aligned} \max \;\; z = & \sum_{i=1}^{n} p_i x_i \\ \text{soggetto a } & \sum_{i=1}^{n} w_i x_i \leq b \\ & x_i \in \{0,1\} \qquad i = 1, \ldots, n. \end{aligned} \tag{KP}$$

$$f_k(s) = \max \;\; z = \left\{ \sum_{i=1}^{n} p_i x_i : \; \sum_{i=k}^{n} w_i x_i \leq s, \;\; x_i \in \{0,1\} \qquad i = 1, \ldots, n. \right\}$$

$$k = 1, \ldots, n$$

$$s = 0, \ldots, b$$

(KP(k,s))

Programmazione dinamica

- $f_1(b)$ risolve il problema.
- $f_n(s)$ ha soluzione banale.

Relazione ricorsiva ($k = 1, \ldots, n-1$)

Per risolvere $f_k(s)$ bisogna decidere $x_k = 0$ oppure $x_k = 1$ e riempire al meglio lo spazio residuo.

$$f_k(s) = \max \begin{cases} f_{k+1}(s) & (x_k = 0) \\ f_{k+1}(s - w_k) + p_k & (x_k = 1) \end{cases}$$

Si assume $f_k(s) = -\infty$ se $s < 0$.
$f_k(0) = 0 \ \forall k$.
Tempo di calcolo $\mathcal{O}(nb)$.

Programmazione dinamica

Variante profit-based

$$f_k(s) = \min\left\{ \sum_{i=k}^{n} w_i x_i : \ \sum_{i=k}^{n} p_i x_i = s, \ x_i \in \{0,1\} \right\}$$

Inizializzazione

$f_n(p_n) = w_k$, $f_n(s) = \infty$ per $s \neq p_n$.
$f_k(0) = 0 \ \forall k$

$$f_k(s) = \min \begin{cases} f_{k+1}(s) & (x_k = 0) \\ f_{k+1}(s - p_k) + w_k & (x_k = 1) \end{cases}$$

Il problema è risolto dalla $f_1(s) \leq b$ con s più grande possibile.