Ottimizzazione Combinatoria

Algoritmi deterministici ad approssimazione garantita

Claudio Arbib

claudio.arbib@univaq.it

Università dell'Aquila

Sommario

1. Algoritmi di approssimazione (deterministici)

   Approssimazione garantita

   Schemi di approssimazione
2. Esempio 1: Commesso Viaggiatore
3. Esempio 2: 0-1 Knapsack

Algoritmi di approssimazione

Gli algoritmi euristici sono metodi per calcolare soluzioni ammissibili, nel nostro caso per problemi di OC

- Questi metodi non garantiscono, in generale, che la soluzione trovata sia ottima; in cambio, convergono alla soluzione in tempo polinomiale
- Alcuni problemi di OC ammettono euristiche polinomiali che dànno una soluzione X, il cui valore è "non troppo lontano" dall'ottimo X^*

Definizione 1: Un metodo euristico è ε-approssimante per un dato ε > 0 se

$$\frac{|c(X) - c(X^*)|}{c(X^*)} \leq \varepsilon$$

Nota: Supponiamo $c(X)$ non negativa e normalizzata, cioè $c(\emptyset) = 0$

min c(X) ⇔ min c(X) + Δ, quindi per ogni problema e ε potrei scegliere Δ in modo tale che

$$\frac{|c(X) + \Delta - c(X^*) - \Delta|}{c(X^*) + \Delta} \leq \varepsilon \quad \text{per ogni } X \in \Im$$

Algoritmi di approssimazione

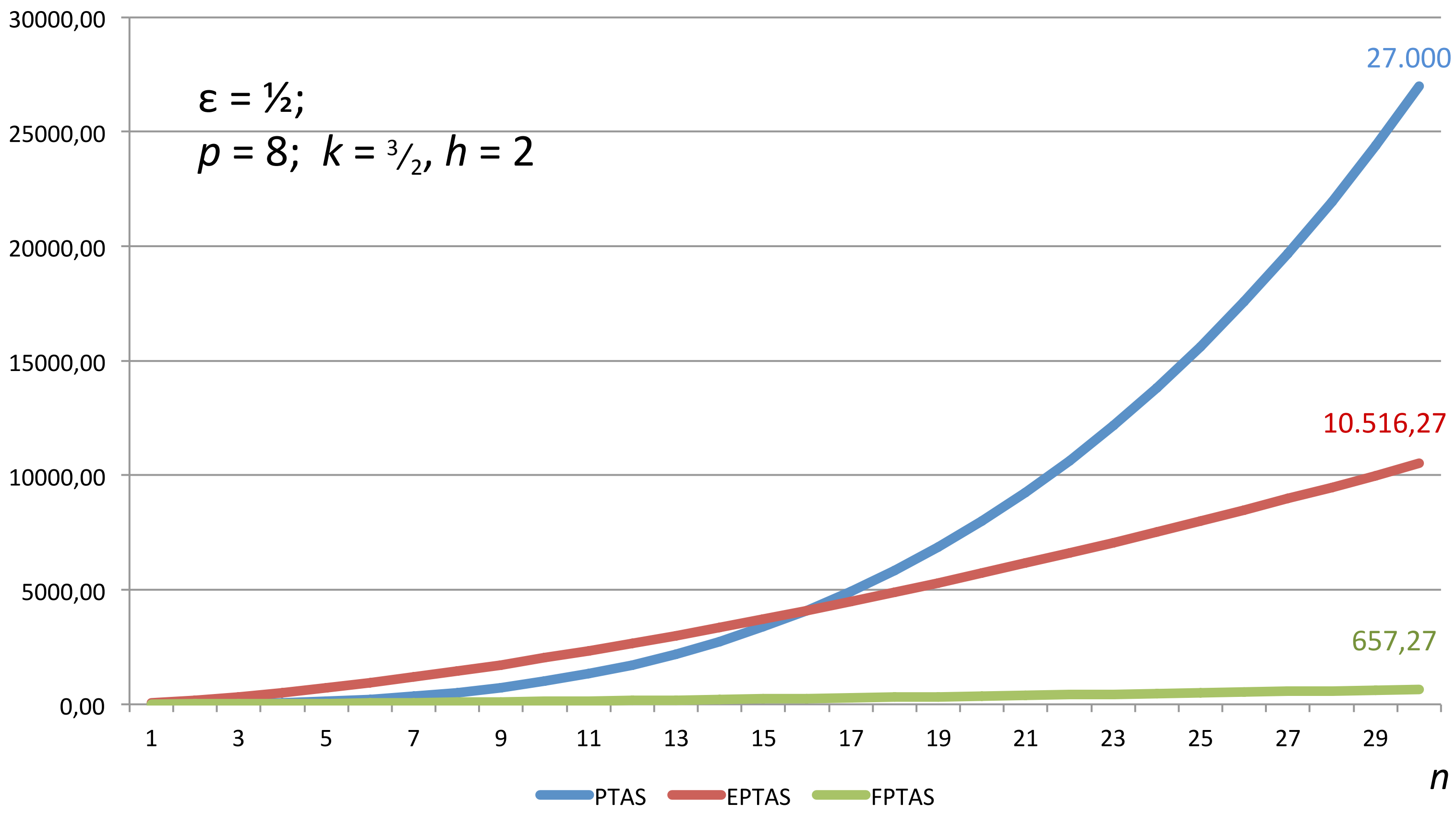
Alcuni algoritmi possono approssimare solo entro un dato ε

Altri invece entro qualunque ε: come ci si può attendere, la loro complessità (un polinomio $\chi(n)$ nella taglia n del problema) cresce al decrescere di ε

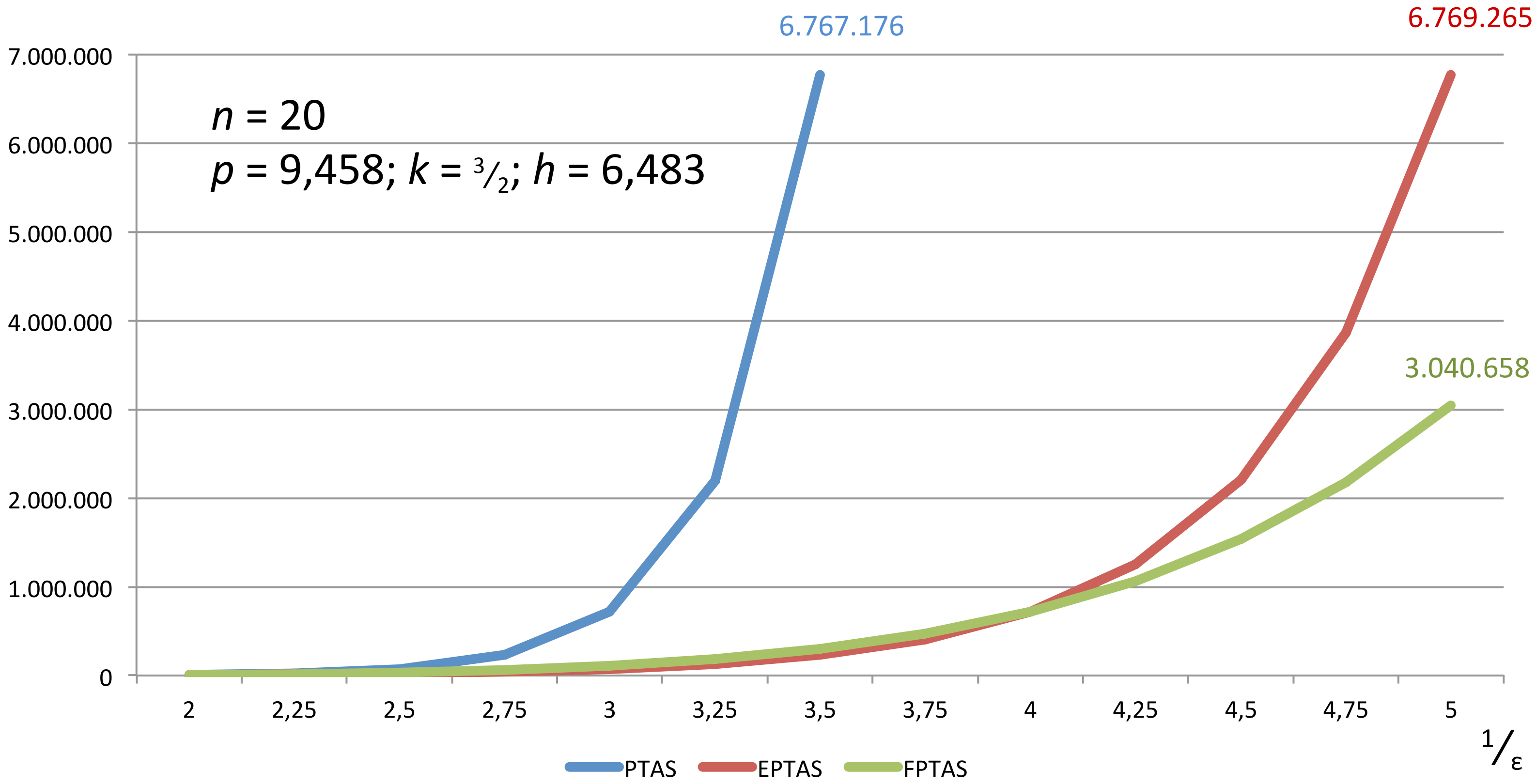
Definizione 2:

- PTAS (Polynomial-time approximation scheme): $\chi(n) = O(n^k)$ per una costante k. Questo non esclude cose come $n^{k/\varepsilon}$
- EPTAS (Efficient PTAS): $\chi(n) = O(n^k)$, con esponente k indipendente da ε. Questo non esclude $p^{1/\varepsilon} n^k$
- FPTAS (Fully PTAS): $\chi(n) = O((1/\varepsilon)^h n^k)$ per h e k costanti date

Algoritmi di approssimazione



Algoritmi di approssimazione



Complessità di PTAS, EPTAS e FPTAS al crescere di $^1/_\varepsilon$
p, k e h scelti in modo da normalizzare il dato per n = 20 e $\varepsilon = {}^1/_2$
$(8.000 = 20^{k/0,5} \approx p^{1/0,5}\, 20^k \approx 20^k/0,5^h)$

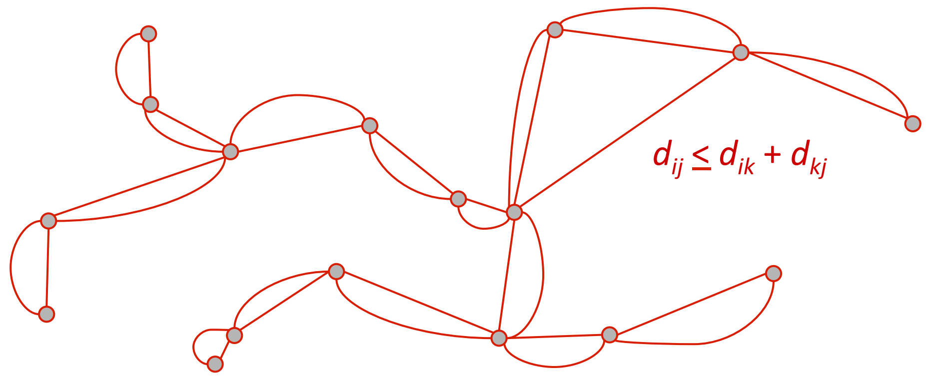
Esempio 1: COMMESSO VIAGGIATORE

Problema 1: (TSP) Trovare il percorso più breve che tocca tutte le città

Da una mappa stradale ricaviamo una matrice **D** che dà le distanze $d_{ij} \geq 0$ fra le città i e j
Supponiamo **D** simmetrica, il che è sensato per le lunghe distanze
Inoltre, per ogni città intermedia k, $d_{ij} \leq d_{ik} + d_{kj}$ (diseguaglianza triangolare)

COMMESSO VIAGGIATORE

Il costo di E è $d(E) = 2d(T^*)$
Con delle scorciatoie si ha un circuito Hamiltoniano H di costo $d(H) \leq 2d(T^*)$



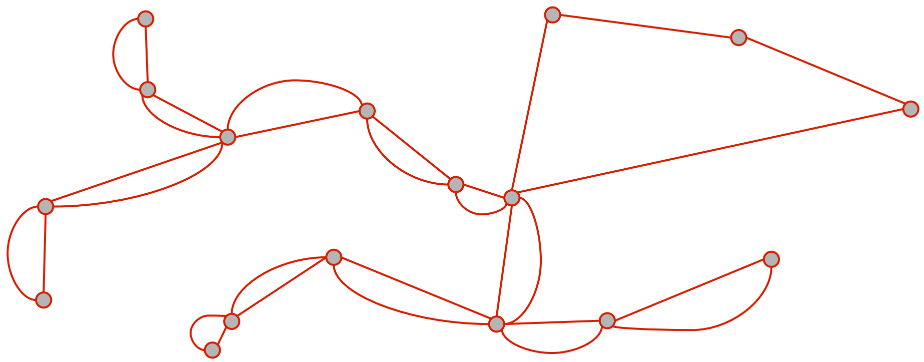
Costruiamo un circuito Euleriano E raddoppiando un albero T^* che ricopre le città al minimo costo $d(T^*)$

Da una mappa stradale ricaviamo una matrice **D** che dà le distanze $d_{ij} \geq 0$ fra le città i e j
Supponiamo **D** simmetrica, il che è sensato per le lunghe distanze
Inoltre, per ogni città intermedia k, $d_{ij} \leq d_{ik} + d_{kj}$ (diseguaglianza triangolare)

COMMESSO VIAGGIATORE

Il costo di E è $d(E) = 2d(T^*)$

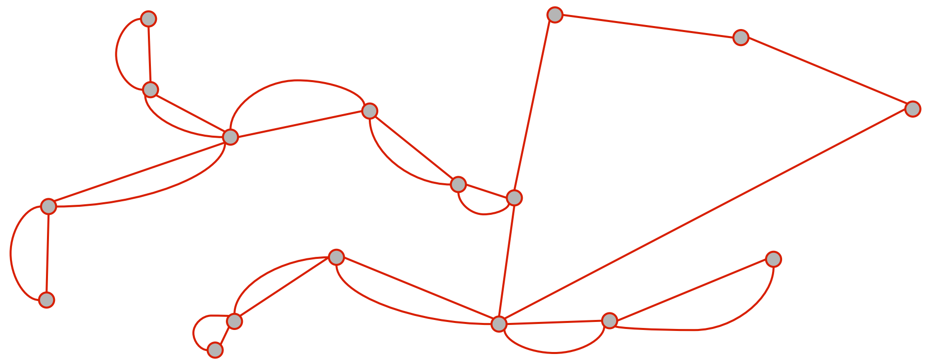
Con delle scorciatoie si ha un circuito Hamiltoniano H di costo $d(H) \leq 2d(T^*)$



COMMESSO VIAGGIATORE

Il costo di E è $d(E) = 2d(T^*)$

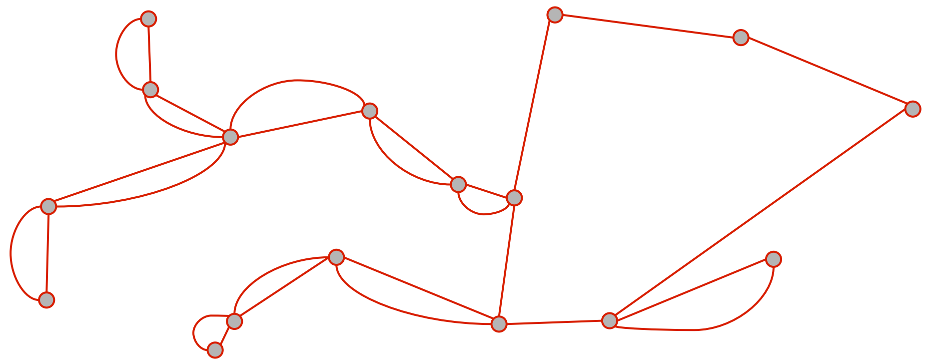
Con delle scorciatoie si ha un circuito Hamiltoniano H di costo $d(H) \leq 2d(T^*)$



COMMESSO VIAGGIATORE

Il costo di E è $d(E) = 2d(T^*)$

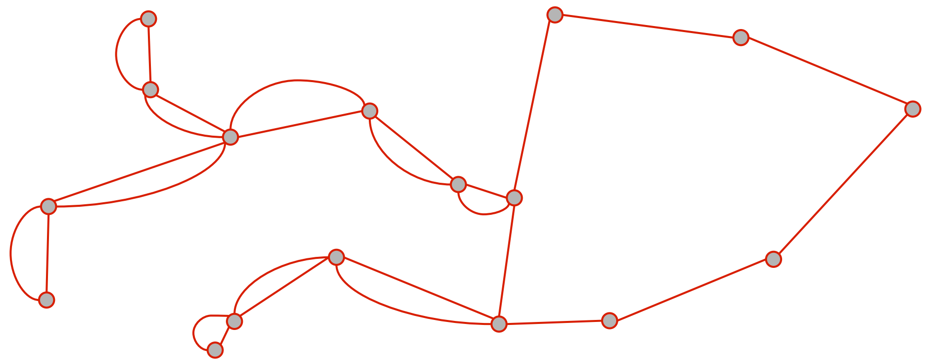
Con delle scorciatoie si ha un circuito Hamiltoniano H di costo $d(H) \leq 2d(T^*)$



COMMESSO VIAGGIATORE

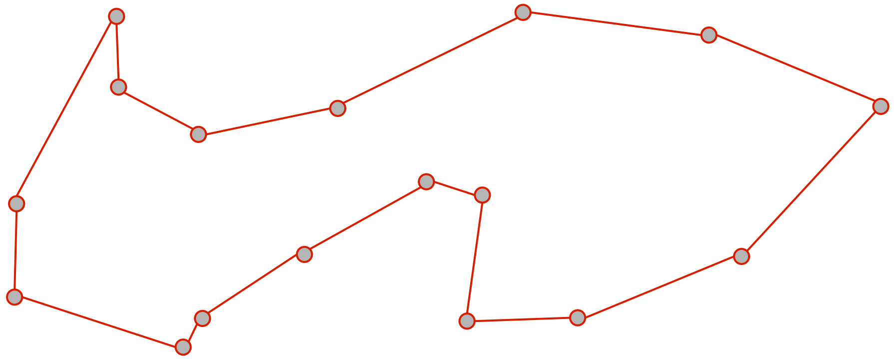
Il costo di E è $d(E) = 2d(T^*)$

Con delle scorciatoie si ha un circuito Hamiltoniano H di costo $d(H) \leq 2d(T^*)$



COMMESSO VIAGGIATORE

Il costo di E è $d(E) = 2d(T^*)$
Con delle scorciatoie si ha un circuito Hamiltoniano H di costo $d(H) \leq 2d(T^*)$



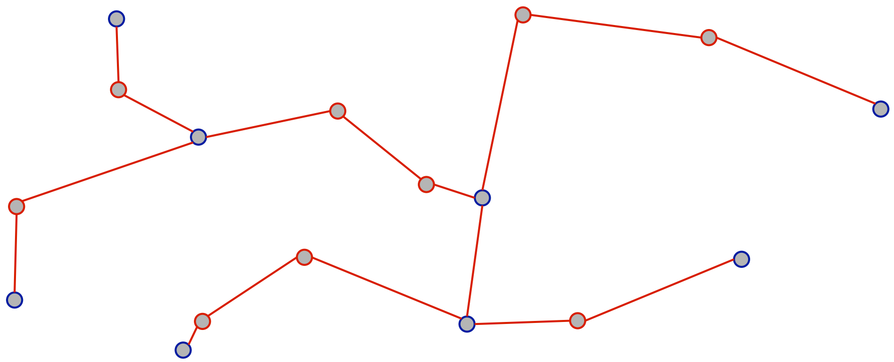
L'ottimo H^* è un circuito, cioè un albero ricoprente T più un arco pq

Quindi $d(H^*) \ = \ d(T) + d_{pq} \ > \ d(T) \ \geq \ d(T^*)$

Mettendo insieme,

$$\frac{d(H) - d(H^*)}{d(H^*)} \leq \frac{2d(T^*) - d(H^*)}{d(H^*)} \leq \frac{2d(T^*) - d(T^*)}{d(T^*)} = 1$$

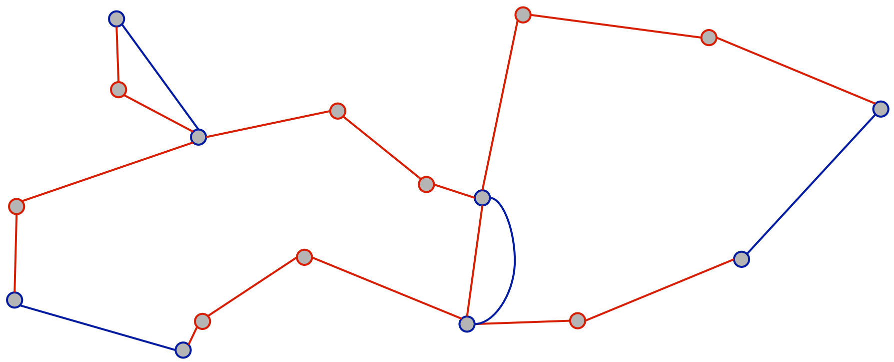
COMMESSO VIAGGIATORE



Invece di raddoppiare l'albero T^*, Christofides propose di costruire il circuito Euleriano unendo i vertici di T^* di grado dispari con un matching perfetto M^* di peso minimo

COMMESSO VIAGGIATORE

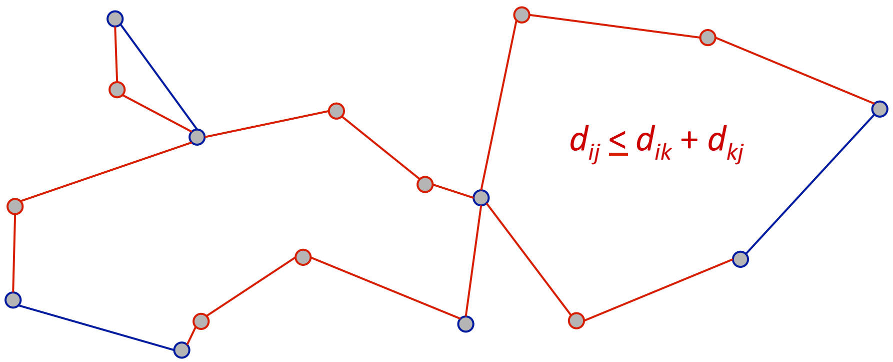
Il costo di E è $d(E) = d(T^*) + d(M^*)$



Invece di raddoppiare l'albero T^*, Christofides propose di costruire il circuito Euleriano unendo i vertici di T^* di grado dispari con un matching perfetto M^* di peso minimo
Ricordiamo che questi vertici sono sempre in numero pari, quindi M^* esiste

COMMESSO VIAGGIATORE

Il costo di E è $d(E) = d(T^*) + d(M^*)$
Di nuovo, le scorciatoie danno un circuito H di costo $d(H) \leq d(T^*) + \mathrm{d}(M^*)$

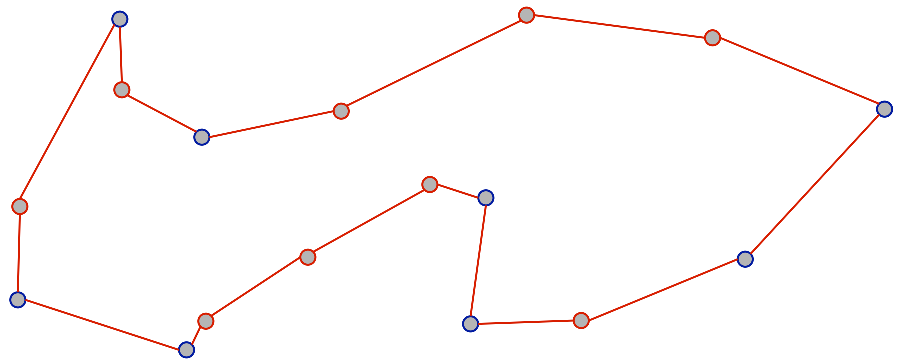


Invece di raddoppiare l'albero T^*, Christofides propose di costruire il circuito Euleriano unendo i vertici di T^* di grado dispari con un matching perfetto M^* di peso minimo
Ricordiamo che questi vertici sono sempre in numero pari, quindi M^* esiste

COMMESSO VIAGGIATORE

Il costo di *E* è $d(E) = d(T^*) + d(M^*)$

Di nuovo, le scorciatoie danno un circuito *H* di costo $d(H) \leq d(T^*) + \mathrm{d}(M^*)$

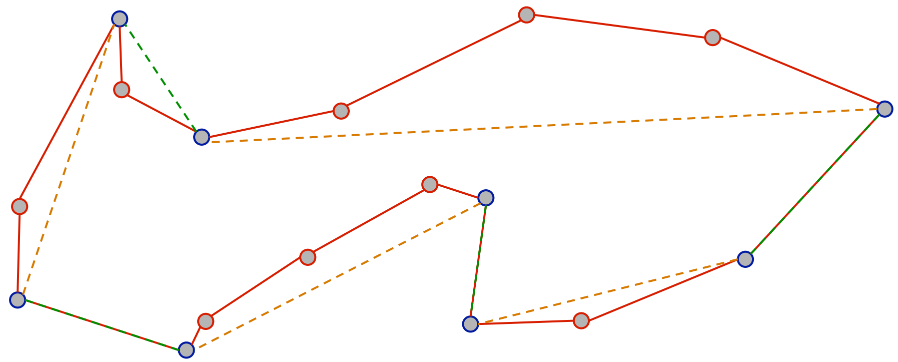


L'ottimo H^* è un albero ricoprente *T* più un arco *pq*, perciò di nuovo $d(H^*) \geq d(T^*)$

COMMESSO VIAGGIATORE

Il costo di E è $d(E) = d(T^*) + d(M^*)$

Di nuovo, le scorciatoie danno un circuito H di costo $d(H) \leq d(T^*) + \mathrm{d}(M^*)$



L'ottimo H^* è un albero ricoprente T più un arco pq, perciò di nuovo $d(H^*) \geq d(T^*)$

Ora usiamo H^* prendendo scorciatoie in modo da toccare solo i nodi blu

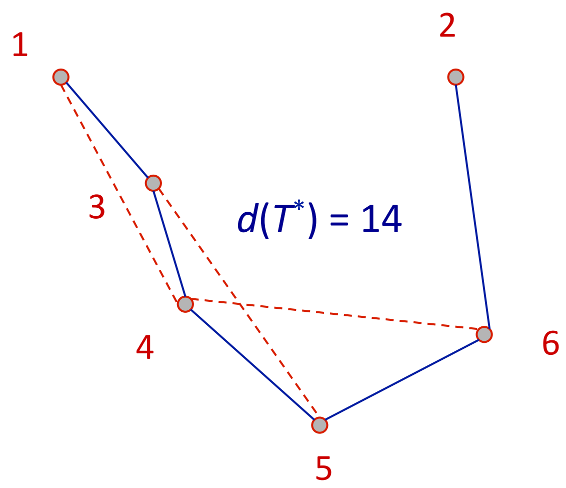
Il circuito $M' \cup M''$ non è più lungo di H^*: $d(H^*) \geq d(M') + d(M'') \geq 2d(M^*)$

Quindi,

$$\frac{d(H) - d(H^*)}{d(H^*)} \leq \frac{d(T^*) + d(M^*) - d(H^*)}{d(H^*)} \leq \frac{d(T^*) + d(M^*) - d(T^*)}{2d(M^*)} = \tfrac{1}{2}$$

Esempio numerico

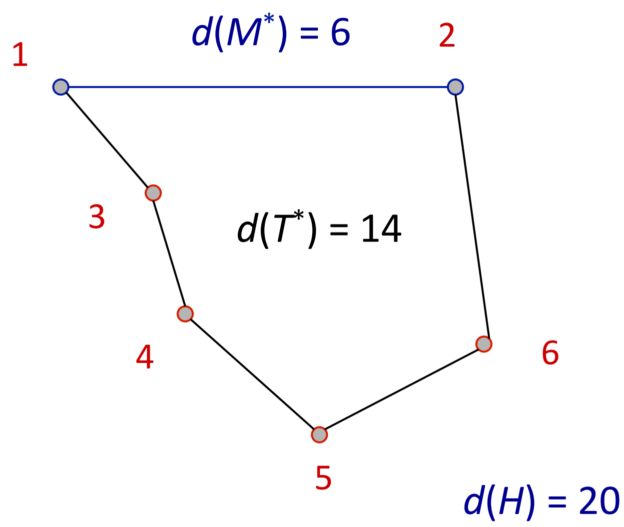
$$\mathbf{D} = \begin{array}{c|cccccc|} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & - & 6 & 2 & 4 & 7 & 7 \\ 2 & 6 & - & 5 & 6 & 6 & 4 \\ 3 & 2 & 5 & - & 2 & 4 & 5 \\ 4 & 4 & 6 & 2 & - & 3 & 4 \\ 5 & 7 & 6 & 4 & 3 & - & 3 \\ 6 & 7 & 4 & 5 & 4 & 3 & - \\ \hline \end{array}$$



Calcolo un albero ricoprente T^* di peso minimo

Esempio numerico

$$
\mathbf{D} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array}
\begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\
\begin{bmatrix}
- & 6 & 2 & 4 & 7 & 7 \\
6 & - & 5 & 6 & 6 & 4 \\
2 & 5 & - & 2 & 4 & 5 \\
4 & 6 & 2 & - & 3 & 4 \\
7 & 6 & 4 & 3 & - & 3 \\
7 & 4 & 5 & 4 & 3 & -
\end{bmatrix} \end{array}
$$



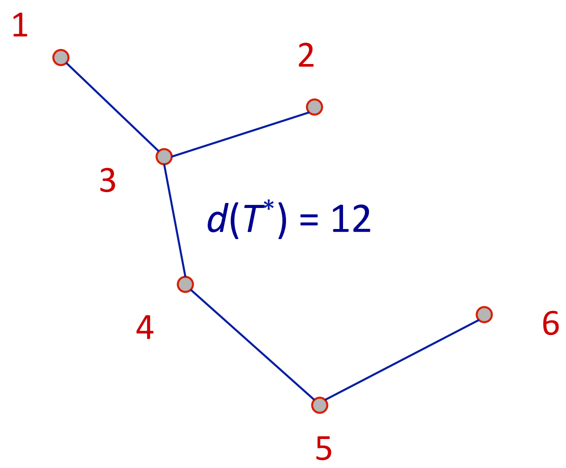
Calcolo un albero ricoprente T^* di peso minimo

Calcolo un matching minimo M^* fra i nodi di grado dispari di T^*

Se occorre cerco le scorciatoie e ottengo un circuito Hamiltoniano

Altro esempio numerico

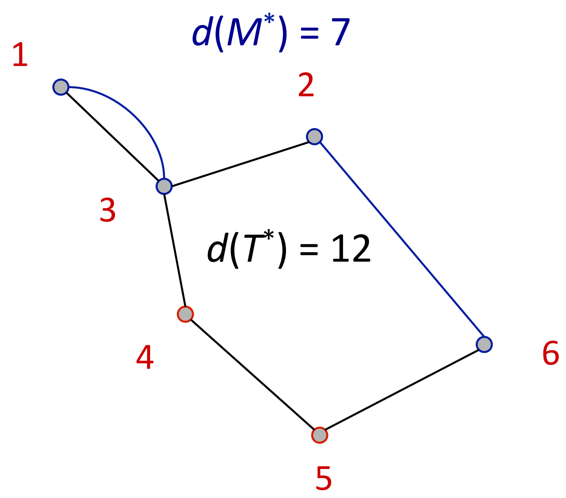
$$
\mathbf{D} = \begin{array}{c|cccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
1 & - & 4 & 2 & 4 & 7 & 7 \\
2 & 4 & - & 2 & 4 & 5 & 5 \\
3 & 2 & 2 & - & 2 & 4 & 5 \\
4 & 4 & 4 & 2 & - & 3 & 4 \\
5 & 7 & 5 & 4 & 3 & - & 3 \\
6 & 7 & 5 & 5 & 4 & 3 & - \\
\hline
\end{array}
$$



Calcolo un albero ricoprente T^* di peso minimo

Altro esempio numerico

$$\mathbf{D} = \begin{array}{c|cccccc|} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & - & 4 & 2 & 4 & 7 & 7 \\ 2 & 4 & - & 2 & 4 & 5 & 5 \\ 3 & 2 & 2 & - & 2 & 4 & 5 \\ 4 & 4 & 4 & 2 & - & 3 & 4 \\ 5 & 7 & 5 & 4 & 3 & - & 3 \\ 6 & 7 & 5 & 5 & 4 & 3 & - \\ \hline \end{array}$$

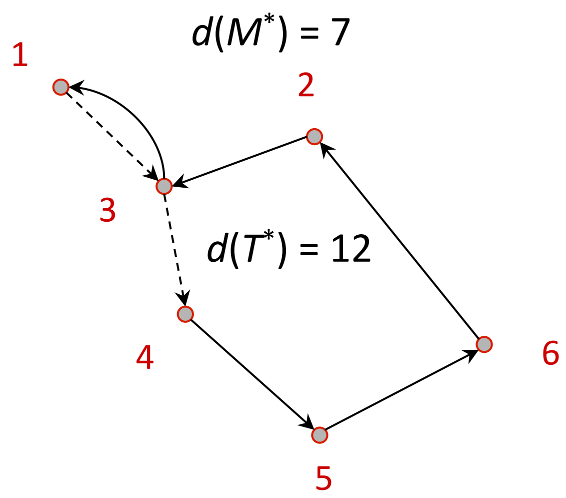


Calcolo un albero ricoprente T^* di peso minimo

Calcolo un matching minimo M^* fra i nodi di grado dispari di T^*

Altro esempio numerico

$$\mathbf{D} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \left[\begin{array}{cccccc} - & 4 & 2 & 4 & 7 & 7 \\ 4 & - & 2 & 4 & 5 & 5 \\ 2 & 2 & - & 2 & 4 & 5 \\ 4 & 4 & 2 & - & 3 & 4 \\ 7 & 5 & 4 & 3 & - & 3 \\ 7 & 5 & 5 & 4 & 3 & - \end{array}\right] \end{array}$$



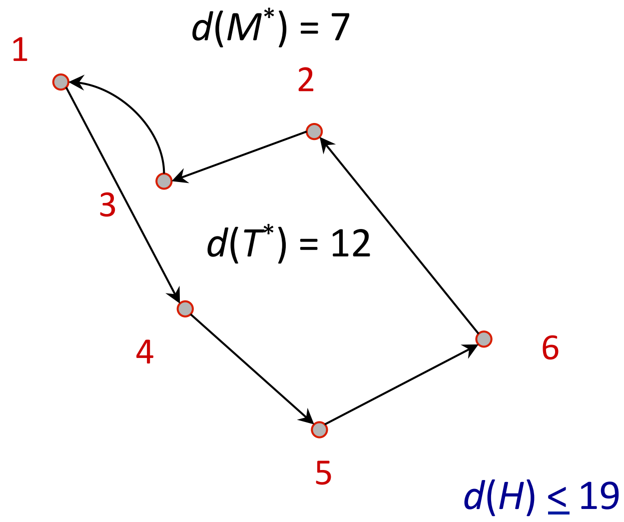
Calcolo un albero ricoprente T^* di peso minimo

Calcolo un matching minimo M^* fra i nodi di grado dispari di T^*

Se occorre cerco le scorciatoie e ottengo un circuito Hamiltoniano

Altro esempio numerico

$$\mathbf{D} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \left[\begin{array}{cccccc} - & 4 & 2 & 4 & 7 & 7 \\ 4 & - & 2 & 4 & 5 & 5 \\ 2 & 2 & - & 2 & 4 & 5 \\ 4 & 4 & 2 & - & 3 & 4 \\ 7 & 5 & 4 & 3 & - & 3 \\ 7 & 5 & 5 & 4 & 3 & - \end{array}\right] \end{array}$$



Calcolo un albero ricoprente T^* di peso minimo

Calcolo un matching minimo M^* fra i nodi di grado dispari di T^*

Se occorre cerco le scorciatoie e ottengo un circuito Hamiltoniano

Esempio 2: 0-1 KNAPSACK

Dati $n + 1$ numeri positivi $a_1, \ldots, a_n, b$

- uno knapsack è un $K \subseteq N = \{1, \ldots, n\}$ per cui $a(K) = \sum_{k \in K} a_k \leq b$

Sia $\Im$ la classe di tutti gli knapsack di N

Problema 3: (0-1 KNAPSACK) Data una funzione utilità $c: N \rightarrow \mathrm{IR}_+$, trovare uno knapsack $X^* \in \Im$ tale che $c(X^*) \geq c(X)$ per ogni $X \in \Im$

Semplifico: Elimino gli $i \in N$ per cui $a_i > b$

Se $X, Y \in \Im$, $c(X) = c(Y)$ e $a(X) < a(Y)$, allora elimino Y

L'ultima semplificazione è cruciale per un metodo di programmazione dinamica: per ogni k e c interessa solo un $X \subseteq \{1, \ldots, k\}$ with $c(X) = c$

Sia $X_{k,c}$ un X con $a(X)$ minimo fra tutti i sottoinsiemi di $\{1, \ldots, k\}$ e $c(X_{k,c}) = c$

Esempio 2: 0-1 KNAPSACK

Vale la seguente formula di ricorrenza

$$a(X_{k,c}) = a(X_{k-1,c}) \quad \text{se } c < c_k$$

$$a(X_{k,c}) = \min\{\underbrace{a(X_{k-1,c})}_{\text{non prendo } k},\ \underbrace{a_k + a(X_{k-1,c-c_k})}_{\text{prendo } k}\} \quad \text{se } c \geq c_k$$

- Se k pesa più di c, inserirlo porta sicuramente a una soluzione di peso > c: quindi (per definizione) k non può far parte di $X_{k,c}$
- Se k non pesa più di c, allora dobbiamo decidere se inserirlo o no: a parità di peso c, prendiamo l'insieme con ingombro minore

Sia $X_{k,c}$ un X con $a(X)$ minimo fra tutti i sottoinsiemi di $\{1, \ldots, k\}$ e $c(X_{k,c}) = c$

Esempio 2: 0-1 Knapsack

Vale la seguente formula di ricorrenza

$a(X_{k,c}) = a(X_{k-1,c})$ se $c < c_k$

$a(X_{k,c}) = \min\{a(X_{k-1,c}), a_k + a(X_{k-1,c-c_k})\}$ se $c \geq c_k$

non prendo k ← → prendo k

Esempio: $n = 4$

$a_1 = 4, a_2 = 5, a_3 = 6, a_4 = 3, b = 10$

$c_1 = 2, c_2 = 4, c_3 = 5, c_4 = 1$

Inizializzo: $X_{k,c} = \emptyset$, $a(X_{k,c}) = 0$ per $c = 0$ e ogni k

$X_{1,c} = \{1\}$, $a(X_{1,c}) = 4$ per $c = c_1 = 2$

$a(X_{1,c}) = \infty$ per tutti gli altri c

$\infty \rightarrow$ non c'è soluzione di valore c in $\{1, \ldots, k\}$

La soluzione ottima è un $X_{n,c}$ che verifica $a(X_{n,c}) \leq b$ con c massimo

c	$a(X)$	X
0	0	$\emptyset$
1	∞	
2	4	$\{1\}$
3	∞	
4	∞	
5	∞	
6	∞	
7	∞	
8	∞	
9	∞	
10	∞	
11	∞	
12	∞	

Esempio 2: 0-1 KNAPSACK

Vale la seguente formula di ricorrenza

$a(X_{k,c}) = a(X_{k-1,c})$ se $c < c_k$

$a(X_{k,c}) = \min\{a(X_{k-1,c}), a_k + a(X_{k-1,c-c_k})\}$ se $c \geq c_k$

non prendo k ← → prendo k

Esempio: $n = 4$

$k = 2$

$a_1 = 4$, $a_2 = 5$, $a_3 = 6$, $a_4 = 3$, $b = 10$

$c_1 = 2$, $c_2 = 4$, $c_3 = 5$, $c_4 = 1$

c	$a(X)$	X
0	0	$\varnothing$
1	∞	
2	4	{1}
3	∞	
4	∞	
5	∞	
6	∞	
7	∞	
8	∞	
9	∞	
10	∞	
11	∞	
12	∞	

$a(X_{2,c}) = \min\{ a(X_{1,c}), a_2 + a(X_{1,c-c_2}) \}$ per $c \geq c_2 = 4$

$a(X_{2,c}) = a(X_{1,c})$ per $c < c_2 = 4$

Esempio 2: 0-1 KNAPSACK

Vale la seguente formula di ricorrenza

$a(X_{k,c}) = a(X_{k-1,c})$ se $c < c_k$

$a(X_{k,c}) = \min\{\underbrace{a(X_{k-1,c})}_{\text{non prendo } k}, \underbrace{a_k + a(X_{k-1,c-c_k})}_{\text{prendo } k}\}$ se $c \geq c_k$

Esempio: $n = 4$

$k = 2$

$a_1 = 4$, $a_2 = 5$, $a_3 = 6$, $a_4 = 3$, $b = 10$

$c_1 = 2$, $c_2 = 4$, $c_3 = 5$, $c_4 = 1$

c	$a(X)$	X
0	0	$\varnothing$
1	∞	
2	4	{1}
3	∞	
4	∞	
5	∞	
6	∞	
7	∞	
8	∞	
9	∞	
10	∞	
11	∞	
12	∞	

$a(X_{2,4}) = \min\{ a(X_{1,4}), a_2 + a(X_{1,0}) \}$

$a(X_{2,c}) = a(X_{1,c})$ per $c < c_2 = 4$

Esempio 2: 0-1 Knapsack

Vale la seguente formula di ricorrenza

$a(X_{k,c}) = a(X_{k-1,c})$ se $c < c_k$

$a(X_{k,c}) = \min\{a(X_{k-1,c}), a_k + a(X_{k-1,c-c_k})\}$ se $c \geq c_k$

non prendo k ← → prendo k

Esempio: $n = 4$

$k = 2$

$a_1 = 4$, $a_2 = 5$, $a_3 = 6$, $a_4 = 3$, $b = 10$

$c_1 = 2$, $c_2 = 4$, $c_3 = 5$, $c_4 = 1$

c	$a(X)$	X
0	0	$\varnothing$
1	∞	
2	4	{1}
3	∞	
4	5	{2}
5	∞	
6	∞	
7	∞	
8	∞	
9	∞	
10	∞	
11	∞	
12	∞	

$a(X_{2,4}) = \min\{ \infty , 5 + 0 \} = 5$

$a(X_{2,c}) = a(X_{1,c})$ per $c < c_2 = 4$

Esempio 2: 0-1 KNAPSACK

Vale la seguente formula di ricorrenza

$a(X_{k,c}) = a(X_{k-1,c})$ se $c < c_k$

$a(X_{k,c}) = \min\{\underbrace{a(X_{k-1,c})}_{\text{non prendo } k}, \underbrace{a_k + a(X_{k-1,c-c_k})}_{\text{prendo } k}\}$ se $c \geq c_k$

Esempio: $n = 4$

$k = 2$

$a_1 = 4$, $a_2 = 5$, $a_3 = 6$, $a_4 = 3$, $b = 10$

$c_1 = 2$, $c_2 = 4$, $c_3 = 5$, $c_4 = 1$

$a(X_{2,5}) = \min\{ a(X_{1,5}), a_2 + a(X_{1,1}) \}$

$a(X_{2,4}) = \min\{ \infty , 5 + 0 \}$

$a(X_{2,c}) = a(X_{1,c})$ per $c < c_2 = 4$

c	$a(X)$	X
0	0	$\varnothing$
1	∞	
2	4	{1}
3	∞	
4	5	{2}
5	∞	
6	∞	
7	∞	
8	∞	
9	∞	
10	∞	
11	∞	
12	∞	

Esempio 2: 0-1 KNAPSACK

Vale la seguente formula di ricorrenza

$$a(X_{k,c}) = a(X_{k-1,c}) \quad \text{se } c < c_k$$

$$a(X_{k,c}) = \min\{a(X_{k-1,c}),\ a_k + a(X_{k-1,c-c_k})\} \quad \text{se } c \geq c_k$$

non prendo k ← → prendo k

Esempio: $n = 4$

$k = 2$

$a_1 = 4$, $a_2 = 5$, $a_3 = 6$, $a_4 = 3$, $b = 10$

$c_1 = 2$, $c_2 = 4$, $c_3 = 5$, $c_4 = 1$

$a(X_{2,5}) = \min\{\ \infty\ ,\ \infty\ \} = \infty$

$a(X_{2,4}) = \min\{\ \infty\ ,\ 5 + 0\ \}$

$a(X_{2,c}) = a(X_{1,c})$ per $c < c_2 = 4$

c	$a(X)$	X
0	0	$\varnothing$
1	∞	
2	4	{1}
3	∞	
4	5	{2}
5	∞	
6	∞	
7	∞	
8	∞	
9	∞	
10	∞	
11	∞	
12	∞	

Esempio 2: 0-1 KNAPSACK

Vale la seguente formula di ricorrenza

$a(X_{k,c}) = a(X_{k-1,c})$ se $c < c_k$

$a(X_{k,c}) = \min\{\underbrace{a(X_{k-1,c})}_{\text{non prendo } k}, \underbrace{a_k + a(X_{k-1,c-c_k})}_{\text{prendo } k}\}$ se $c \geq c_k$

Esempio: $n = 4$

$k = 2$

$a_1 = 4$, $a_2 = 5$, $a_3 = 6$, $a_4 = 3$, $b = 10$

$c_1 = 2$, $c_2 = 4$, $c_3 = 5$, $c_4 = 1$

$a(X_{2,6}) = \min\{ a(X_{1,6}), a_2 + a(X_{1,2}) \}$

$a(X_{2,5}) = \min\{ \infty , \infty \}$

$a(X_{2,4}) = \min\{ \infty , 5 + 0 \}$

$a(X_{2,c}) = a(X_{1,c})$ per $c < c_2 = 4$

c	$a(X)$	X
0	0	$\emptyset$
1	∞	
2	4	{1}
3	∞	
4	5	{2}
5	∞	
6	∞	
7	∞	
8	∞	
9	∞	
10	∞	
11	∞	
12	∞	

Esempio 2: 0-1 KNAPSACK

Vale la seguente formula di ricorrenza

$a(X_{k,c}) = a(X_{k-1,c})$ se $c < c_k$

$a(X_{k,c}) = \min\{\underbrace{a(X_{k-1,c})}_{\text{non prendo } k}, \underbrace{a_k + a(X_{k-1,c-c_k})}_{\text{prendo } k}\}$ se $c \geq c_k$

Esempio: $n = 4$

$k = 2$

$a_1 = 4$, $a_2 = 5$, $a_3 = 6$, $a_4 = 3$, $b = 10$

$c_1 = 2$, $c_2 = 4$, $c_3 = 5$, $c_4 = 1$

$a(X_{2,6}) = \min\{ \infty , 5 + 4 \} = 9$

$a(X_{2,5}) = \min\{ \infty , \infty \}$

$a(X_{2,4}) = \min\{ \infty , 5 + 0 \}$

$a(X_{2,c}) = a(X_{1,c})$ per $c < c_2 = 4$

c	$a(X)$	X
0	0	$\emptyset$
1	∞	
2	4	{1}
3	∞	
4	5	{2}
5	∞	
6	9	{1, 2}
7	∞	
8	∞	
9	∞	
10	∞	
11	∞	
12	∞	

non ci sono altre soluzioni per $k = 2$

Esempio 2: 0-1 KNAPSACK

Vale la seguente formula di ricorrenza

$$a(X_{k,c}) \;=\; a(X_{k-1,c}) \qquad \text{se } c < c_k$$
$$a(X_{k,c}) \;=\; \min\{\underbrace{a(X_{k-1,c})}_{\text{non prendo } k},\ \underbrace{a_k + a(X_{k-1,c-c_k})}_{\text{prendo } k}\} \quad \text{se } c \geq c_k$$

Esempio: $n = 4$

$k = 3$ $\quad a_1 = 4,\ a_2 = 5,\ a_3 = 6,\ a_4 = 3,\ b = 10$

$c_1 = 2,\ c_2 = 4,\ c_3 = 5,\ c_4 = 1$

c	$a(X)$	X
0	0	$\varnothing$
1	∞	
2	4	{1}
3	∞	
4	5	{2}
5	∞	
6	9	{1, 2}
7	∞	
8	∞	
9	∞	
10	∞	
11	∞	
12	∞	

$$a(X_{3,c}) \;=\; \min\{\, a(X_{2,c}),\ a_3 + a(X_{2,c-c_3}) \,\} \quad \text{per } c \geq c_3 = 5$$

$$a(X_{3,c}) \;=\; a(X_{2,c}) \quad \text{per } c < c_3 = 5$$

Esempio 2: 0-1 KNAPSACK

Vale la seguente formula di ricorrenza

$a(X_{k,c}) = a(X_{k-1,c})$ se $c < c_k$

$a(X_{k,c}) = \min\{a(X_{k-1,c}), a_k + a(X_{k-1,c-c_k})\}$ se $c \geq c_k$

non prendo k — prendo k

Esempio: $n = 4$

$k = 3$

$a_1 = 4, a_2 = 5, a_3 = 6, a_4 = 3, b = 10$

$c_1 = 2, c_2 = 4, c_3 = 5, c_4 = 1$

c	$a(X)$	X
0	0	$\varnothing$
1	∞	
2	4	{1}
3	∞	
4	5	{2}
5	∞	
6	9	{1, 2}
7	∞	
8	∞	
9	∞	
10	∞	
11	∞	
12	∞	

$a(X_{3,5}) = \min\{ a(X_{2,5}), a_3 + a(X_{2,0}) \}$

$a(X_{3,c}) = a(X_{2,c})$ per $c < c_3 = 5$

Esempio 2: 0-1 KNAPSACK

Vale la seguente formula di ricorrenza

$a(X_{k,c}) = a(X_{k-1,c})$ se $c < c_k$

$a(X_{k,c}) = \min\{\underbrace{a(X_{k-1,c})}_{\text{non prendo } k},\ \underbrace{a_k + a(X_{k-1,c-c_k})}_{\text{prendo } k}\}$ se $c \geq c_k$

Esempio: $n = 4$

$k = 3$

$a_1 = 4,\ a_2 = 5,\ a_3 = 6,\ a_4 = 3,\ b = 10$

$c_1 = 2,\ c_2 = 4,\ c_3 = 5,\ c_4 = 1$

c	$a(X)$	X
0	0	$\varnothing$
1	∞	
2	4	{1}
3	∞	
4	5	{2}
5	6	{3}
6	9	{1, 2}
7	∞	
8	∞	
9	∞	
10	∞	
11	∞	
12	∞	

$a(X_{3,5}) = \min\{\ \infty\ ,\ 6 + 0\ \} = 6$

$a(X_{3,c}) = a(X_{2,c})$ per $c < c_3 = 5$

Esempio 2: 0-1 KNAPSACK

Vale la seguente formula di ricorrenza

$a(X_{k,c}) = a(X_{k-1,c})$ se $c < c_k$

$a(X_{k,c}) = \min\{a(X_{k-1,c}), a_k + a(X_{k-1,c-c_k})\}$ se $c \geq c_k$

non prendo k ← → prendo k

Esempio: $n = 4$

$k = 3$

$a_1 = 4$, $a_2 = 5$, $a_3 = 6$, $a_4 = 3$, $b = 10$

$c_1 = 2$, $c_2 = 4$, $c_3 = 5$, $c_4 = 1$

c	$a(X)$	X
0	0	$\varnothing$
1	∞	
2	4	{1}
3	∞	
4	5	{2}
5	6	{3}
6	9	{1, 2}
7	∞	
8	∞	
9	∞	
10	∞	
11	∞	
12	∞	

$a(X_{3,6}) = \min\{ a(X_{2,6}), a_3 + a(X_{2,1}) \}$

$a(X_{3,5}) = \min\{ \infty , 6 + 0 \}$

$a(X_{3,c}) = a(X_{2,c})$ per $c < c_3 = 5$

Esempio 2: 0-1 KNAPSACK

Vale la seguente formula di ricorrenza

$a(X_{k,c}) = a(X_{k-1,c})$ se $c < c_k$

$a(X_{k,c}) = \min\{\underbrace{a(X_{k-1,c})}_{\text{non prendo } k}, \underbrace{a_k + a(X_{k-1,c-c_k})}_{\text{prendo } k}\}$ se $c \geq c_k$

Esempio: $n = 4$

$k = 3$

$a_1 = 4, a_2 = 5, a_3 = 6, a_4 = 3, b = 10$

$c_1 = 2, c_2 = 4, c_3 = 5, c_4 = 1$

c	$a(X)$	X
0	0	$\varnothing$
1	∞	
2	4	{1}
3	∞	
4	5	{2}
5	6	{3}
6	9	{1, 2}
7	∞	
8	∞	
9	∞	
10	∞	
11	∞	
12	∞	

$a(X_{3,6}) = \min\{ 9 , \infty \} = 9$

$a(X_{3,5}) = \min\{ \infty , 6 + 0 \}$

$a(X_{3,c}) = a(X_{2,c})$ per $c < c_3 = 5$

Esempio 2: 0-1 KNAPSACK

Vale la seguente formula di ricorrenza

$a(X_{k,c}) = a(X_{k-1,c})$ se $c < c_k$

$a(X_{k,c}) = \min\{\underbrace{a(X_{k-1,c})}_{\text{non prendo } k}, \underbrace{a_k + a(X_{k-1,c-c_k})}_{\text{prendo } k}\}$ se $c \geq c_k$

Esempio: $n = 4$

$k = 3$

$a_1 = 4, a_2 = 5, a_3 = 6, a_4 = 3, b = 10$

$c_1 = 2, c_2 = 4, c_3 = 5, c_4 = 1$

$a(X_{3,7}) = \min\{ a(X_{2,7}), a_3 + a(X_{2,2}) \}$

$a(X_{3,6}) = \min\{ 9, \infty \}$

$a(X_{3,5}) = \min\{ \infty, 6 + 0 \}$

$a(X_{3,c}) = a(X_{2,c})$ per $c < c_3 = 5$

c	$a(X)$	X
0	0	$\emptyset$
1	∞	
2	4	{1}
3	∞	
4	5	{2}
5	6	{3}
6	9	{1, 2}
7	∞	
8	∞	
9	∞	
10	∞	
11	∞	
12	∞	

Esempio 2: 0-1 KNAPSACK

Vale la seguente formula di ricorrenza

$a(X_{k,c}) = a(X_{k-1,c})$ se $c < c_k$

$a(X_{k,c}) = \min\{\underbrace{a(X_{k-1,c})}_{\text{non prendo } k}, \underbrace{a_k + a(X_{k-1,c-c_k})}_{\text{prendo } k}\}$ se $c \geq c_k$

Esempio: $n = 4$

$k = 3$

$a_1 = 4$, $a_2 = 5$, $a_3 = 6$, $a_4 = 3$, $b = 10$

$c_1 = 2$, $c_2 = 4$, $c_3 = 5$, $c_4 = 1$

$a(X_{3,7}) = \min\{ \infty , 6 + 4 \} = 10$

$a(X_{3,6}) = \min\{ 9 , \infty \}$

$a(X_{3,5}) = \min\{ \infty , 6 + 0 \}$

$a(X_{3,c}) = a(X_{2,c})$ per $c < c_3 = 5$

c	$a(X)$	X
0	0	$\varnothing$
1	∞	
2	4	{1}
3	∞	
4	5	{2}
5	6	{3}
6	9	{1, 2}
7	10	{1, 3}
8	∞	
9	∞	
10	∞	
11	∞	
12	∞	

non ci sono altre soluzioni per $k = 3$

Esempio 2: 0-1 KNAPSACK

Vale la seguente formula di ricorrenza

$a(X_{k,c}) = a(X_{k-1,c})$ se $c < c_k$

$a(X_{k,c}) = \min\{a(X_{k-1,c}), a_k + a(X_{k-1,c-c_k})\}$ se $c \geq c_k$

Complessità: L'algoritmo riempie una tabella con $n \cdot \Sigma\, c_k$ elementi

Ad esempio, con $c_1 = 201$, $c_2 = 433$, $c_3 = 591$, $c_4 = 176$ dobbiamo considerare $4 \cdot 1.401 = 5.604$ elementi

Se invece $c_1 = 2015$, $c_2 = 4333$, $c_3 = 5910$, $c_4 = 1769$, gli elementi da considerare diventano $4 \cdot 14.027 = 56.108$

Sia $c_{max} = \max_k\{c_k\}$: per codificare il problema servono $L = n \cdot \lceil \log_2(c_{max}) \rceil$ bit

Gli $n \cdot \Sigma\, c_k = O(n^2 c_{max})$ elementi della tabella sono perciò $O(n^2 2^{L/n})$, un esponenziale nella taglia L del problema!

L'algoritmo, insomma, ha complessità esponenziale

Esempio 2: 0-1 KNAPSACK

Trucco: Dividiamo c_k per R, arrotondiamo per difetto e moltiplichiamo il risultato per R: la complessità si riduce di un fattore R

Esempio: Con R = 100 i c_k diventano $c_1^R = 200$, $c_2^R = 400$, $c_3^R = 500$, $c_4^R = 100$, e la tabella si riduce a 48 elementi (di 100 in 100)

Ad esempio, con $c_1 = 201$, $c_2 = 433$, $c_3 = 591$, $c_4 = 176$ dobbiamo considerare 4·1.401 = 5.604 elementi

Prezzo da pagare: Abbiamo sì ridotto la complessità, ma abbiamo risolto un diverso problema P^R: la sua soluzione X^R è ammissibile, perché $a(X^R) \leq b$; ma quanto vale con i c_k del problema iniziale?

$$c^R(X^R) = 200x_1^R + 400x_2^R + 500x_3^R + 100x_4^R \leq$$
$$\leq 201x_1^R + 433x_2^R + 591x_3^R + 176x_4^R = c(X^R)$$

Notiamo che 200 < 201 < 300 = 200 + R

Esempio 2: 0-1 KNAPSACK

Ora, da un lato $\underbrace{c(X^R) \;\geq\; c^R(X^R)}_{c_k \geq c_k^R} \;\underbrace{\geq\; c^R(X^*)}_{X^R \text{ ottimo di } P^R} \;\underbrace{>\; c(X^*)}_{c_k^R + R > c_k} - nR$

Dall'altro $c(X^*) \;>\; c_{max}$ (perché l'elemento più pesante è una soluzione)

Prezzo da pagare: Abbiamo sì ridotto la complessità, ma abbiamo risolto un diverso problema P^R: la sua soluzione X^R è ammissibile, perché $a(X^R) \leq b$; ma quanto vale con i c_k del problema iniziale?

Notiamo che 200 < 201 < 300 = 200 + R, cioè in generale $c_k^R \leq c_k < c_k^R + R$

Esempio 2: 0-1 KNAPSACK

Ora, da un lato $\underbrace{c(X^R) \ \geq \ c^R(X^R)}_{c_k \geq c_k^R} \ \underbrace{\geq \ c^R(X^*)}_{X^R \text{ ottimo di } P^R} \ \underbrace{> \ c(X^*)}_{c_k^R + R > c_k} - nR$

Dall'altro $\quad c(X^*) \ > \ c_{max}$ (perché l'elemento più pesante è una soluzione)

Quindi $\dfrac{c(X^*) - c^R(X^R)}{c(X^*)} \leq \dfrac{nR}{c_{max}} = \varepsilon$ cioè $\dfrac{c_{max}}{R} = \dfrac{n}{\varepsilon}$

Ricordiamo che la complessità dell'algoritmo usato è $\chi(L, n) = O(n^2 2^{L/n})$, dove L è la taglia del problema risolto

Nel problema iniziale, $L = n \cdot \lceil \log_2(c_{max}) \rceil$ bit

Nel problema P^R, $L = n \cdot \left\lceil \log_2\left(\left\lfloor \frac{c_{max}}{R} \right\rfloor\right) \right\rceil = n \cdot \log_2\left(\left\lfloor \frac{n}{\varepsilon} \right\rfloor\right) + 1 \leq n \cdot \log_2\left(\frac{n}{\varepsilon}\right) + 1$

Quindi $\chi(L, n) = O(\frac{n^3}{\varepsilon})$, un polinomio in n e $^1/_\varepsilon$ → FPTAS con $h = 1$, $k = 3$