

## Esercizi sugli algoritmi approssimati

1. Si applichi l'algoritmo primale-duale del set covering all'esempio numerico illustrato nella slide 12 della lezione.  
2. Che valore di approssimazione ci si aspetta? E' possibile ottenere il valore tight di approssimazione su questa istanza?

### Soluzione:

Inizialmente poniamo  $x=0, y=0$ .

Siccome inizialmente nessun elemento è coperto, considero ad es. il primo ossia  $e=1$ : questo è presente in  $S_2, S_3, S_4$  e  $S_5$  che hanno costo 6, 10, 14 e 5 rispettivamente.

Perciò poniamo  $y_1=5$  così  $S_5$  diventa un insieme tight (ossia  $\sum_{e:e \in S_5} y_e = c(S_5)$ ). Di conseguenza  $S_5$  è scelto per il cover ossia  $x_5=1$ . Perciò gli elementi sinora coperti sono 1, 4 e 5.

Preleviamo il successivo elemento non coperto, ossia 2.

Questo è presente nei set  $S_3$  e  $S_4$  che hanno costo 10 e 14.

Per rendere tight  $S_3$  occorre  $\sum_{e:e \in S_3} y_e = y_1 + y_2 + y_5 = 10 \rightarrow y_2 = 5$ , mentre per rendere tight  $S_4$  occorre che  $\sum_{e:e \in S_4} y_e = y_1 + y_2 + y_4 = 14 \rightarrow y_2 = 9$ .

Perciò poniamo  $y_2=5$  in modo da rendere  $S_3$  tight.  $S_3$  è aggiunto al cover e perciò  $x_3=1$ .

L'unico elemento non ancora coperto è 3. Gli insiemi che lo contengono sono  $S_1, S_2$  e  $S_6$ .

Per rendere tight  $S_1$  occorre che  $y_3+y_5=c(1)=4 \rightarrow y_3=4$ .

Per rendere tight  $S_2$  occorre che  $y_1+y_3+y_5=c(2)=6 \rightarrow y_3=6-5=1$ .

Per rendere tight  $S_6$  occorre che  $y_3+y_4=c(6)=6 \rightarrow y_3=6$ .

Quindi poniamo  $y_3=1$  e in questo modo  $S_2$  è tight e  $x_2=1$ .

Perciò la soluzione finale è il cover  $\{S_1, S_4, S_5\}$  di costo 23.

L'elemento più frequente è il 5 con  $f=4$ .

Il valore ottimo è  $z^*=16$  dato da  $\{S_3, S_6\}$ .

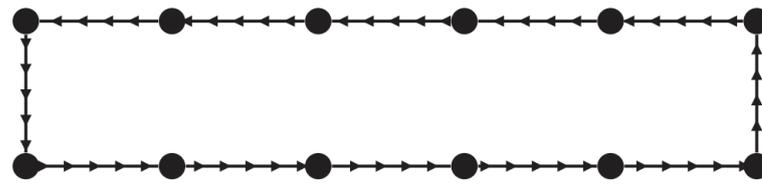
Il valore di approssimazione massimo atteso è  $f=4$  ossia un valore  $\leq 16 \cdot 4=64$  che è ampiamente rispettato avendo ottenuto una soluzione di valore 23.

Su questa istanza non è possibile ottenere il valore di approssimazione tight perchè la somma dei costi di tutti i set è  $45 < f \cdot z^*=64$ .

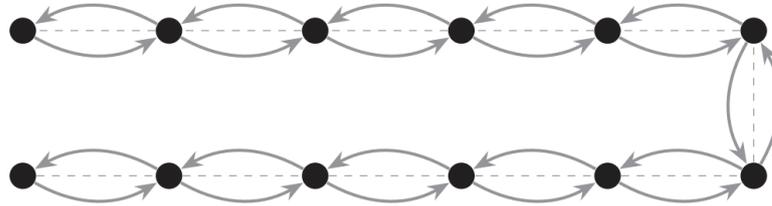
2. Si consideri l'algoritmo del doppio albero per il TSP. Si costruisca un'istanza tight ossia dove l'algoritmo del doppio albero fornisce una soluzione ammissibile che è il doppio del valore ottimo (come previsto nel caso peggiore).

### Soluzione:

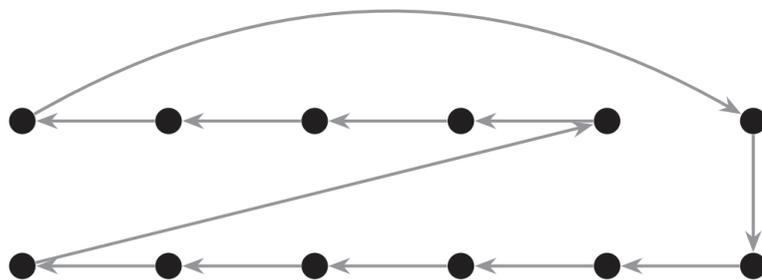
Si consideri una griglia  $n \times 2$ -grid nel piano Euclideo con lati di lunghezza 1 e distanza Euclidea per tutti gli altri lati (vedere Figure 3.1a). Un tour ottimo (come mostrato in figura) ha lunghezza  $2n$  (dato che non è possibile un tour più corto che visita tutti i nodi).



(a) TSP instance and optimal solution



(b) minimum spanning tree and Euler tour



(c) solution provided by the double tree algorithm

D'altra parte, si consideri lo spanning tree e il corrispondente tour Euleriano rappresentato in Figura 3.1b. Partendo dal vertice sinistro inferiore e seguendo il tour Euleriano verso il basso si ottiene il ciclo Hamiltoniano di Figura 3.1c.

Per una griglia  $2 \times n$  con distanze euclidee, questo fornisce un ciclo Hamiltoniano di lunghezza.

$$3n - 3 + \sqrt{1 + (n - 2)^2} > 4n - 5.$$

Thus, the approximation ratio of this solution is

$$\frac{4n - 5}{2n} = 2 - \frac{5}{2n} \rightarrow 2 \quad \text{for } n \rightarrow \infty.$$

3. Si scriva la formulazione duale del rilassamento lineare del problema del Minimum Weight Node Cover Problem (MWNCP) presentato nella slide 20: che cosa rappresenta tale formulazione? In che modo può consentire di costruire un algoritmo approssimato per il MWNCP?

**Soluzione:**

Il rilassamento lineare del MWNCP è:

$$\min z = \sum_{i=1}^n c_i x_i$$

$$x_i + x_j \geq 1 \quad \forall [i, j] \in E$$

$$x_i \geq 0 \quad \text{for } i = 1, \dots, n$$

si noti che non occorre imporre  $x_i \leq 1$  perchè la f.o. è da minimizzare con coefficienti  $c_i \geq 0$  e i vincoli di copertura non richiedono alle variabili di assumere valori maggiori di 1.

Il duale del suo rilassamento è:

$$\begin{aligned} \max \sum_{[i,j] \in E} y_{ij} \\ \sum_{[i,j] \in E} y_{ij} \leq c_i \quad \forall i = 1, \dots, n \quad (1) \\ y_{ij} \geq 0 \quad \forall [i,j] \in E \end{aligned}$$

Il significato del duale è di scegliere il maggior numero di lati (anche più volte) con il vincolo che per ogni nodo  $i$ , il numero di lati scelti (contati con la loro molteplicità) che sono incidenti al nodo  $i$  non deve superare il costo del nodo  $i$ .

Si noti che nel caso particolare in cui tutti i coefficienti  $c_i=1$  (o sono costanti, perché in tal caso i vincoli (1) si possono semplificare in modo da avere i termini noti unitari) il problema duale rappresenta il matching di cardinalità massima perché i vincoli (1) modellano la condizione che per ogni nodo ci sia al più un lato incidente.

Tornando al caso generale, attraverso il problema duale possiamo ottenere un algoritmo approssimato nel seguente modo. Partiamo dalla soluzione duale  $y_{ij} = 0 \quad \forall [i,j] \in E$  e definiamo inizialmente il cover  $C=\{\}$ . Chiaramente questa soluzione è ammissibile, perchè i vincoli (1) sono soddisfatti. Scegliamo un lato qualunque  $[u,v] \in E$  e incrementiamo il più possibile il valore di  $y_{uv}$  sinchè il vincolo (1) sarà attivo ossia soddisfatto con l'uguaglianza o per il nodo  $u$  o per il nodo  $v$ . Ossia  $y_{uv} = y_{uv} + \min \{c_u - \sum_{[u,j] \in E} y_{uj}, c_v - \sum_{[v,j] \in E} y_{vj}\}$ . Supponiamo che sia  $u$  il nodo che rende attivo il vincolo (1), allora aggiungiamo il nodo  $u$  al cover  $C$  ed eliminiamo  $[u,v]$  dall'insieme  $E$  dei lati. Andiamo avanti così sinché non ci sono più lati.

Alla fine della procedura  $C$  è un vertex cover, perchè un lato viene eliminato solo se uno dei suoi nodi estremi è stato inserito in  $C$ . Vogliamo ora dimostrare che il costo di  $C$  è al massimo il doppio di quello ottimo.

Innanzitutto  $\sum_{i \in C} c_i = \sum_{i \in C} \sum_{[i,j] \in E} y_{ij}$  perchè ogni volta che un nodo è inserito in  $C$  è perchè rende attivo un vincolo (1).

Abbiamo quindi:

$$\sum_{i \in C} c_i = \sum_{i \in C} \sum_{[i,j] \in E} y_{ij} = \sum_{[i,j] \in E} |C \cap \{i,j\}| y_{ij} \leq \sum_{[i,j] \in E} 2y_{ij}$$

ma siccome  $y$  è soluzione ammissibile per il duale e la funzione obiettivo del duale valutata in  $y$  è un lower bound al valore ottimo  $z^*$  del primale, si ha che

$$\sum_{[i,j] \in E} 2y_{ij} \leq 2z^*$$

Abbiamo quindi dimostrato che  $\sum_{i \in C} c_i \leq 2z^*$ .

4. Si formuli mediante la PLI il problema di assegnamento di lavori alle macchine descritto nella slide 13. Se ne scriva il duale del rilassamento lineare. Si sviluppi un algoritmo approssimato primale-duale secondo lo schema della slide 27 analogamente a quanto fatto per il problema del set covering.

**Soluzione:**

Il problema può essere modellato mediante le variabili binarie  $x_{ij}=1$  se il job  $j$  è assegnato alla macchina  $i$ , 0 altrimenti, nel seguente modo:

min  $z$

$$z \geq \sum_{j=1, \dots, n} p_j x_{ij} \quad \forall i = 1, \dots, m \quad (z \text{ deve essere un maggiorante dei tempi di completamento dei lavori})$$

$$\sum_{i=1, \dots, m} x_{ij} = 1 \quad \forall j = 1, \dots, n \quad (\text{ogni job deve essere assegnato a 1 e 1 sola macchina})$$

$$x_{ij} \in \{0,1\} \quad \forall i = 1, \dots, m, j = 1, \dots, n$$

Il suo rilassamento lineare consiste nel porre le variabili  $x_{ij} \geq 0 \quad \forall i = 1, \dots, m, j = 1, \dots, n$  : non occorre porle  $\leq 1$  perché se una variabile fosse maggiore di 1 violerebbe il vincolo di assegnamento.

Il suo duale è

$$\max \sum_{j=1, \dots, n} \mu_j$$

$$-p_j \lambda_i + \mu_j \leq 0 \quad \forall i = 1, \dots, m, j = 1, \dots, n$$

$$\sum_{i=1, \dots, m} \lambda_i = 1$$

$$\lambda_i \geq 0 \quad \forall i = 1, \dots, m$$

$$\mu_j \text{ libere } \forall j = 1, \dots, n$$

Condizioni CSC primali:

$$x_{ij}=0 \text{ o altrimenti } 0 = \frac{0}{\alpha} \leq -p_j \lambda_i + \mu_j \leq 0 \quad \text{ossia } -p_j \lambda_i + \mu_j = 0$$

(per la variabile  $z$  non deve essere scritta perché la variabile è libera)

Condizioni CSC duali:

$$\lambda_i = 0 \text{ o altrimenti } 0 \leq -\sum_{j=1..n} p_j x_{ij} + z \leq 0 \beta = 0 \quad \text{ossia } z = \sum_{j=1..n} p_j x_{ij}$$

(per le variabili  $\mu_j$  le CSC non devono essere scritte perché sono libere)

Poichè i termini noti di tutte le CSC sono nulli, lo schema di pag.27 in questo caso non può essere usato per costruire un algoritmo  $\alpha\beta$  approssimato dato che nelle CSC qualunque valore di  $\alpha$  e  $\beta$  produrrebbe la stessa CSC, anche  $\alpha = \beta = 1$  e quindi porterebbe ad ottenere un'algoritmo polinomiale in grado di ottenere la soluzione ottima, il che è assurdo dato che il problema è NP-hard.

Si consideri la variante del problema di assegnamento di lavori alle macchine dove il tempo di processamento dipende dalla macchina usata ossia  $p_{ij}$  è il tempo di processamento del lavoro  $j$  sulla macchina  $i$ , ogni macchina può essere usata al massimo per un tempo  $T$  e si vuole minimizzare il tempo totale per svolgere tutti i lavori.

Si arrivi a scrivere le CSC primali e duali specificando i valori di  $\alpha$  e  $\beta$  più convenienti da utilizzare e i valori di partenza delle variabili primali e duali da utilizzare.