Lower bound per TSP simmetrico

Vogliamo definire una tecnica di calcolo di lower bound per il problema TSP nel caso simmetrico, ovvero il caso in cui:

$$v_{ij} = v_{ji} \quad \forall\, i, j \in V.$$

Definizione: 1-tree

Dato un grafo $G = (V, A)$ non orientato e un suo nodo $a \in V$, chiamiamo 1-tree un sottografo $Q = (V, A_Q)$ di G con le seguenti proprietà:

- in A_Q ci sono esattamente due archi incidenti sul nodo a;
- se escludo da Q il nodo a e i due archi incidenti su di esso, mi rimane un albero sull'insieme di nodi $V \setminus \{a\}$.

In particolare, da questa definizione segue che $\mid A_Q \mid = \mid V \mid$.

Esempio

Dato il grafo con $V = \{a, b, c, d, e\}$ e

$$A = \{(a, b); (a, c); (b, c); (b, e); (c, d); (d, a); (d, e)\},$$

il sottografo con

$$A_Q = \{(a, b); (d, a); (b, c); (b, e); (d, e)\}$$

è un 1-tree.

Osservazione

Si può notare che ogni circuito hamiltoniano è un 1-tree.

Infatti, in un circuito hamiltoniano su ogni nodo incidono esattamente due archi ed inoltre togliendo un nodo a qualsiasi e i due archi del circuito incidenti su di esso si ottiene un albero sull'insieme di nodi $V \setminus \{a\}$.

Il viceversa non è vero (lo 1-tree dell'esempio *non* è un circuito hamiltoniano).

Quindi ...

... se indichiamo con S' l'insieme degli 1-tree su un grafo G, tale insieme contiene la regione ammissibile S del problema TSP.

In altre parole, il problema

$$\min_{Q=(V,A_Q)\in S'} \sum_{(i,j)\in A_Q} v_{ij}$$

risulta essere un rilassamento per il problema TSP simmetrico e la sua risoluzione restituisce un lower bound per il valore ottimo del problema TSP.

Calcolo del lower bound

- **Passo 1.** Si risolva il problema MST sul grafo ottenuto scartando da $G = (V, A)$ il nodo a prescelto e tutti gli archi incidenti su di esso. Sia A_T l'insieme di archi della soluzione trovata;
- **Passo 2.** Si aggiungano ad A_T i due archi (a, k) e (a, h) a distanza minima tra tutti quelli incidenti sul nodo a prescelto.
- **Passo 3.** Si restituisca $Q = (V, A_Q)$ con $A_Q = A_T \cup \{(a, k); (a, h)\}$.

Tempi di calcolo

Risoluzione del problema MST $\rightarrow$ in tempo polinomiale (ad esempio con l'algoritmo greedy).

Calcolo dei due valori minimi $\rightarrow$ in tempo polinomiale.

Quindi i tempi di calcolo complessivi sono polinomiali.

Nota bene

La scelta del nodo a è arbitraria.

Al costo di un maggiore sforzo computazionale, si possono anche calcolare $\mid V \mid$ diversi lower bound scegliendo come nodo a tutti i nodi del grafo G e calcolando per ciascuno di essi il lower bound.

Come lower bound complessivo del problema originario si utilizza il migliore (ovvero il più grande) tra tutti i $\mid V \mid$ lower bound calcolati.

Sottinsiemi di S di forma particolare

Siano dati due sottinsiemi di archi $A_0, A_1 \subseteq A$ con $A_0 \cap A_1 = \emptyset$.

I sottinsiemi di S che ci interessano sono:

$$S(A_0, A_1) = \{C = (V, A_C) \in S \ : \ \forall\, (i,j) \in A_1 \ : \ (i,j) \in A_C,\ \forall\, (i,j) \in A_0 \ : \ (i,j) \notin A_C\},$$

ovvero in $S(A_0, A_1)$ abbiamo tutti i circuiti hamiltoniani che contengono sicuramente gli archi in A_1 e che sicuramente non contengono gli archi in A_0.

Lower bound per sottoproblemi

Per il calcolo del lower bound di un sottoproblema $S(A_0, A_1)$, si utilizza la stessa procedura imponendo però la presenza degli archi in A_1 ed escludendo quella degli archi in A_0 sia nella risoluzione del problema MST sia nell'individuazione dei due archi incidenti sul nodo a.

In particolare, si può risolvere il problema MST sempre con l'algoritmo greedy ma:

- inizializzando l'insieme di archi A_T non con l'insieme vuoto ma con tutti gli archi in A_1 non incidenti sul nodo a;
- non considerando gli archi in A_0 durante l'esecuzione dell'algoritmo greedy.

Inoltre ...

- se in A_1 non sono presenti archi incidenti sul nodo a, metteremo in A_Q i due archi a distanza minima tra tutti quelli incidenti sul nodo a e al di fuori di A_0;
- se in A_1 è già presente un arco incidente sul nodo a questo entrerà in A_Q insieme a quello a distanza minima tra tutti quelli incidenti sul nodo a e al di fuori di A_0 e A_1;
- se in A_1 sono già presenti due archi incidenti sul nodo a, solo questi entreranno in A_Q.

Esempio

Supponiamo di avere il seguente problema del TSP simmetrico

	1	2	3	4	5
1	–	5	8	3	5
2	5	–	4	6	2
3	8	4	–	10	3
4	3	6	10	–	1
5	5	2	3	1	–

Proviamo a calcolare il lower bound per il sottoproblema $S(A_0, A_1)$ con $A_0 = \{(1,3);(4,5)\}$ e $A_1 = \{(1,5);(2,4)\}$. Utilizziamo come nodo a il nodo 1.

Continua

Per prima cosa dobbiamo risolvere il problema MST sull'insieme di nodi $V \setminus \{1\}$ imponendo la presenza dell'arco $(2, 4)$ che è in A_1 ed escludendo quella degli archi in A_0.

Utilizzando l'algoritmo greedy con A_T inizializzato con gli archi in A_1 non incidenti sul nodo 1 (in questo caso il solo arco $(2, 4)$) ed escludendo la possibilità di inserire gli archi in A_0, arriviamo al seguente albero su $V \setminus \{1\}$

$$A_T = \{(2, 4); (2, 5); (3, 5)\}.$$

Continua

Notiamo che in A_1 è presente l'arco $(1,5)$ incidente sul nodo 1.

Ad A_T dobbiamo quindi aggiungere, oltre a questo arco $(1,5)$, l'arco a distanza minima tra tutti quelli incidenti sul nodo 1 e al di fuori di A_0 e A_1, ovvero $(1,4)$.

Quindi lo 1-tree ottimo ha l'insieme di archi

$$A_Q = \{(2,4); (2,5); (3,5); (1,5); (1,4)\}$$

con valore ottimo (e quindi lower bound per il sottoproblema $S(A_0, A_1)$) pari a 19.

Si noti anche come $Q = (V, A_Q)$ non sia un circuito hamiltoniano e quindi non possa essere utilizzato per aggiornare (eventualmente) il valore di upper bound.

Modello matematico problema 1-tree

- Abbiamo una variabile binaria x_{ij} per ogni arco (i, j);
- imponiamo che ci siano esattamente due archi incidenti sul nodo a:
$$\sum_{i \in V,\ i \neq a} x_{ia} = 2;$$
- gli archi incidenti sui soli nodi in $V \setminus \{a\}$ devono formare un albero.

Albero su $V \setminus \{a\}$

Per imporre che gli archi selezionati formino un albero su $V \setminus \{a\}$ dobbiamo richiedere che:

- il numero di tali archi sia pari a $| V \setminus \{a\} | -1$, ovvero pari a $| V | -2$, cioé:

$$\sum_{i,j \in V \setminus \{a\}} x_{ij} = | V | -2;$$

- tali archi non formino cicli.

Eliminazione cicli in $V \setminus \{a\}$

Dato $U \subseteq V \setminus \{a\}$, sia

$$E(U) = \{(i,j) \ : \ i,j \in U\}$$

Osservando che un ciclo sui nodi in U dovrebbe contenere $\mid U \mid$ archi in $E(U)$, per eliminare cicli imporremo

$$\sum_{(i,j) \in E(U)} x_{ij} \leq \mid U \mid -1 \qquad \forall\, U \subseteq V \setminus \{a\}$$

NB: Per $\mid U \mid \leq 2$ i vincoli risultanti sono banali e possono essere omessi.

Modello matematico 1 -tree

$$
\begin{array}{lll}
\min & \sum_{i,j \in V,\ i<j} v_{ij} x_{ij} & \\
& \sum_{i \in V,\ i \neq a} x_{ia} = 2 & \\
& \sum_{(i,j) \in E(U)} x_{ij} \leq \mid U \mid -1 & \forall\, U \subseteq V \setminus \{a\}\ :\mid U \mid \geq 3 \\
& \sum_{i,j \in V \setminus \{a\}} x_{ij} = \mid V \mid -2 & \\
& x_{ij} \in \{0,1\} & \forall\, i,j \in V,\ i<j
\end{array}
$$

Esempio

Problema TSP simmetrico con la seguente tabella di distanze:

	1	2	3	4
1	$-$	12	9	14
2	12	$-$	8	9
3	9	8	$-$	1
4	14	9	1	$-$

Modello matematico: esempio

Nodo $a = 1$

$$\min \quad 12x_{12} + 9x_{13} + 14x_{14} + 8x_{23} + 9x_{24} + x_{34}$$
$$x_{12} + x_{13} + x_{14} = 2$$
$$x_{23} + x_{24} + x_{34} \leq 2$$
$$x_{23} + x_{24} + x_{34} = 2$$
$$x_{12}, x_{13}, x_{14}, x_{23}, x_{24}, x_{34} \in \{0, 1\}$$

Si noti come in questo caso il vincolo $x_{23} + x_{24} + x_{34} \leq 2$ sia implicato dall'altro vincolo $x_{23} + x_{24} + x_{34} = 2$ e quindi può essere omesso.

Osservazione

Si può dimostrare che un modello valido per il problema TSP simmetrico è identico a quello visto per il problema 1-tree ma con l'aggiunta che su *tutti* i nodi in V incidano esattamente due archi, ovvero:

$$\begin{array}{lrll}
\min & \sum_{i,j\in V,\ i<j} v_{ij}x_{ij} & & \\
 & \sum_{i\in V,\ i\neq j} x_{ij} = 2 & \forall\ j\in V & \\
 & \sum_{(i,j)\in E(U)} x_{ij} \leq\mid U\mid -1 & \forall\ U\subseteq V\setminus\{a\}\ :\mid U\mid\geq 3 & \\
 & \sum_{i,j\in V\setminus\{a\}} x_{ij} =\mid V\mid -2 & & \\
 & x_{ij}\in\{0,1\} & \forall\ i,j\in V,\ i<j &
\end{array}$$

Esempio

$$\begin{aligned}
\min \quad 12x_{12} + 9x_{13} + 14x_{14} + 8x_{23} + 9x_{24} + x_{34}\\
x_{12} + x_{13} + x_{14} = 2\\
x_{12} + x_{23} + x_{24} = 2\\
x_{13} + x_{23} + x_{34} = 2\\
x_{14} + x_{24} + x_{34} = 2\\
x_{23} + x_{24} + x_{34} = 2\\
x_{12}, x_{13}, x_{14}, x_{23}, x_{24}, x_{34} \in \{0, 1\}
\end{aligned}$$

In pratica ...

... possiamo vedere il problema 1-tree come il rilassamento del problema TSP simmetrico ottenuto omettendo da questo tutti i vincoli che richiedono che vi siano esattamente due archi incidenti su ogni nodo, tranne quello relativo al nodo a.

Come già visto, l'omissione di vincoli può essere vista come un caso particolare di rilassamento lagrangiano in cui tutti i moltiplicatori di Lagrange sono fissati a 0.

Ma allora possiamo vedere cosa succede se prendiamo moltiplicatori di Lagrange diversi da 0.

Rilassamento lagrangiano

Introduciamo ora moltiplicatori di Lagrange $\lambda = (\lambda_k)_{k \in V \setminus \{a\}}$ per i vincoli che richiedono che esattamente due archi incidano sui nodi, con l'unica eccezione del nodo a selezionato.

Modello rilassamento lagrangiano

$$
\begin{aligned}
u(\lambda) \quad = \min \quad & \sum_{i,j\in V,\ i<j} v_{ij}x_{ij} + \sum_{k\in V\setminus\{a\}} \lambda_k (2 - \sum_{i\in V,\ i\neq k} x_{ik}) \\
& \sum_{i\in V,\ i\neq a} x_{ia} = 2 \\
& \sum_{(i,j)\in E(U)} x_{ij} \leq\mid U \mid -1 \qquad \forall\, U \subseteq V \\
& \sum_{i,j\in V\setminus\{a\}} x_{ij} =\mid V \mid -2 \\
& x_{ij} \in \{0,1\} \qquad \forall\, i,j \in V
\end{aligned}
$$

Nota bene

Essendo i vincoli di uguaglianza, possiamo considerare valori dei moltiplicatori di Lagrange $\lambda_k, \; k \in V \setminus \{a\}$, anche negativi e non solo maggiori o uguali a zero.

Si vede che il rilassamento visto prima basato sul calcolo dello 1-tree minimo è un caso particolare di questo rilassamento lagrangiano in cui $\lambda_k = 0$ per tutti i $k \in V \setminus \{a\}$.

Continua

Per comodità di notazione si include nell'obiettivo anche un termine relativo al vincolo di incidenza di esattamente due archi sul nodo a con il relativo moltiplicatore di Lagrange λ_a, imponendo però che questo possa assumere il solo valore 0.

In tal modo avremo a che fare con un vettore di moltiplicatori di Lagrange $\lambda = (\lambda_k)_{k \in V}$ le cui componenti possono assumere valori positivi, negativi o nulli con la sola eccezione della componente relativa al nodo a che può assumere solo valore nullo.

Prima riscrittura del modello

$$u(\lambda) = \min \quad \sum_{i,j \in V,\ i<j} v_{ij} x_{ij} + \sum_{k \in V} \lambda_k \left(2 - \sum_{i \in V,\ i \neq k} x_{ik}\right)$$

$$\sum_{i \in V,\ i \neq a} x_{ia} = 2$$

$$\sum_{(i,j) \in E(U)} x_{ij} \leq \mid U \mid -1 \qquad \forall\, U \subseteq V \setminus \{a\} \ : \mid U \mid \geq 3$$

$$\sum_{i,j \in V \setminus \{a\}} x_{ij} = \mid V \mid -2$$

$$x_{ij} \in \{0,1\} \qquad \forall\, i,j \in V,\ i<j$$

Modello finale

$$
\begin{aligned}
u(\lambda) = \min \quad & \sum_{i,j \in V,\ i<j} (v_{ij} - \lambda_i - \lambda_j) x_{ij} + 2 \sum_{k \in V} \lambda_k \\
& \sum_{i \in V,\ i \neq a} x_{ia} = 2 \\
& \sum_{(i,j) \in E(U)} x_{ij} \leq \mid U \mid -1 \qquad \forall\, U \subseteq V \setminus \{a\}\ : \mid U \mid \geq 3 \\
& \sum_{i,j \in V \setminus \{a\}} x_{ij} = \mid V \mid -2 \\
& x_{ij} \in \{0,1\} \qquad \forall\, i,j \in V,\ i<j
\end{aligned}
$$

Risolvere il rilassamento lagrangiano

Per valori fissati dei moltiplicatori di Lagrange λ_k, $k \in V$, il rilassamento lagrangiano è facilmente risolvibile con la procedura vista in precedenza per l'individuazione dello 1-tree minimo.

Infatti, il problema consiste nell'individuare lo 1-tree minimo tenendo conto che le distanze degli archi sono ora definite come segue

$$v'_{ij} = v_{ij} - \lambda_i - \lambda_j.$$

Esempio - prima riscrittura

$$
\begin{aligned}
u(\lambda_1, \ldots, \lambda_4) = \min \quad & 12x_{12} + 9x_{13} + 14x_{14} + 8x_{23} + 9x_{24} + x_{34} + \\
& \lambda_1(2 - x_{12} - x_{13} - x_{14}) + \lambda_2(2 - x_{12} - x_{23} - x_{24}) + \\
& + \lambda_3(2 - x_{13} - x_{23} - x_{34}) + \lambda_4(2 - x_{14} - x_{24} - x_{34}) \\
& x_{12} + x_{13} + x_{14} = 2 \\
& x_{23} + x_{24} + x_{34} = 2 \\
& x_{12}, x_{13}, x_{14}, x_{23}, x_{24}, x_{34} \in \{0, 1\}
\end{aligned}
$$

Esempio - modello finale

$$
\begin{aligned}
u(\lambda_1,\ldots,\lambda_4) = \min \qquad & (12-\lambda_1-\lambda_2)x_{12} + (9-\lambda_1-\lambda_3)x_{13} + \\
& +(14-\lambda_1-\lambda_4)x_{14} + (8-\lambda_2-\lambda_3)x_{23} + \\
& +(9-\lambda_2-\lambda_4)x_{24} + (1-\lambda_3-\lambda_4)x_{34} + 2\sum_{i=1}^{4}\lambda_i \\
& x_{12}+x_{13}+x_{14} = 2 \\
& x_{23}+x_{24}+x_{34} = 2 \\
& x_{12},x_{13},x_{14},x_{23},x_{24},x_{34} \in \{0,1\}
\end{aligned}
$$

Duale lagrangiano

Questo consisterà nell'individuare i valori $\lambda^* = (\lambda^*_k)_{k \in V}$, con $\lambda^*_a = 0$, per cui la funzione $u(\lambda)$ ha il valore più grande possibile.

In altre parole si tratta di risolvere il seguente problema

$$\max_{\lambda:\ \lambda_a = 0} \ u(\lambda)$$

Esempio

Risolvendo il rilassamento lagrangiano con $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 0$ si ottiene lo 1-tree minimo

$$(2,3) \;\; (3,4) \;\; (1,2) \;\; (1,3)$$

con un lower bound pari a 30.

Esempio - continua

Se ora però consideriamo i seguenti moltiplicatori di Lagrange:

$$\lambda_1 = 0 \quad \lambda_2 = 0 \quad \lambda_3 = -1 \quad \lambda_4 = 1,$$

arriviamo ad un problema con la seguente tabella di distanze

	1	2	3	4
1	–	12	10	13
2	12	–	9	8
3	10	9	–	1
4	13	8	1	–

Esempio - continua

Lo 1-tree minimo con queste distanze è

$$(3,4) \;\; (2,4) \;\; (1,2) \;\; (1,3)$$

e il lower bound è pari a

$$2\sum_{i\in V}\lambda_i + (\text{valore 1-tree minimo}) = 0 + (1+8+12+10) = 31,$$

migliore rispetto al precedente.

Nel caso specifico si osserva anche che quest'ultimo 1-tree minimo è anche un circuito hamiltoniano con distanza complessiva pari a 31 ed è quindi soluzione ottima del problema in questione.

Osservazione

Non ci addentreremo nelle tecniche di risoluzione del duale lagrangiano ma diamo una possibile strategia per migliorare quanto ottenuto con determinati valori $\tilde{\lambda}_i$ dei moltiplicatori di Lagrange (ad esempio, $\tilde{\lambda}_i = 0$ per ogni i).

Nella soluzione ottenuta con i moltiplicatori di Lagrange $\tilde{\lambda}_i$ dobbiamo cercare aumentare il peso degli archi incidenti su nodi con grado superiore a 2 nella soluzione e ridurre quello degli archi incidenti su nodi con grado inferiore a 2.

Per questo possiamo aggiornare i moltiplicatori di Lagrange nel modo seguente:

$$\bar{\lambda}_i = \tilde{\lambda}_i + 2 - \text{grado di } i \text{ nello 1-tree minimo}$$

Nell’esempio

Nell’esempio considerato i gradi dei nodi 1, 2, 3 e 4 nello 1-tree minimo ottenuto con tutti i moltiplicatori di Lagrange nulli sono rispettivamente 2, 2, 3 e 1 e la regola appena vista porta proprio ai moltiplicatori di Lagrange proposti precedentemente.

Aggiornamento upper bound

Per quanto riguarda l'aggiornamento dell'upper bound, ricordiamo che questo si basa sull'identificazione di soluzioni ammissibili durante l'esecuzione dell'algoritmo.

Nel nostro caso tale identificazione si può avere quando la soluzione ottima di un sottoproblema 1-tree dà origine a un circuito hamiltoniano: se la distanza di questo è inferiore all'upper bound attuale, possiamo aggiornare questo proprio con la distanza del circuito hamiltoniano individuato.

Branching del nodo radice

Ci occuperemo ora di specificare come viene partizionata la regione ammissibile S in più sottinsiemi. Abbiamo visto che se non siamo nel caso fortunato in cui la soluzione del rilassamento è un circuito hamiltoniano, tale soluzione è uno 1-tree che contine esattamente un sottocircuito.

Forniremo una semplice regola di suddivisione il cui scopo è quello di impedire il formarsi nei nodi figli di tale sottocircuito.

Continua

Indichiamo con $\{(i^1, j^1),\ (i^2, j^2), \ldots, (i^r, j^r)\}$ gli archi del sottocircuito.

Il primo nodo figlio viene ottenuto imponendo che in esso non sia presente l'arco (i^1, j^1) (cioè si impone $x_{i^1,j^1} = 0$), il secondo nodo figlio viene ottenuto imponendo che sia presente l'arco (i^1, j^1) ma non sia presente l'arco (i^2, j^2) (cioè si impone $x_{i^1,j^1} = 1,\ x_{i^2,j^2} = 0$), e così via fino al r-esimo figlio in cui si impone che siano presenti gli archi (i^k, j^k), $k = 1, \ldots, r-1$, ma non sia presente l'arco (i^r, j^r) (cioè si impone $x_{i^k,j^k} = 1,\ k = 1, \ldots, r-1,\ x_{i^r,j^r} = 0$).

Branching di $S(A_0, A_1)$

Si ripete quanto già visto per il nodo radice con la sola differenza che potremmo imporre la non presenza solo degli archi del sottocircuito che non sono in A_1. (gli archi in A_1 sono già fissati e non possono essere rimossi).

Branching di $S(A_0, A_1)$

Indichiamo con $\{(i^1, j^1),\ (i^2, j^2), \ldots, (i^t, j^t)\}$ gli archi del sottocircuito *che non appartengono ad* A_1. Il primo nodo figlio viene ottenuto imponendo che in esso non sia presente l'arco (i^1, j^1) (cioè si impone $x_{i^1,j^1} = 0$), il secondo nodo figlio viene ottenuto imponendo che sia presente l'arco (i^1, j^1) ma non sia presente l'arco (i^2, j^2) (cioè si impone $x_{i^1,j^1} = 1,\ x_{i^2,j^2} = 0$), e così via fino al t-esimo figlio in cui si impone che siano presenti gli archi (i^k, j^k), $k = 1, \ldots, t-1$, ma non sia presente l'arco (i^t, j^t) (cioè si impone $x_{i^k,j^k} = 1,\ k = 1, \ldots, t-1,\ x_{i^t,j^t} = 0$).

Nell'esempo

Abbiamo il sottoproblema $S(A_0, A_1)$ con $A_0 = \{(1,3);(4,5)\}$ e $A_1 = \{(1,5);(2,4)\}$.

Nel calcolo del lower bound abbiamo individuato lo 1-tree $Q = (V, A_Q)$ con

$$A_Q = \{(2,4);(2,5);(3,5);(1,5);(1,4)\}$$

In questo è presente il circuito

$$1 \to 4 \to 2 \to 5 \to 1.$$

Gli archi del sottocircuito al di fuori di A_1 sono i seguenti

$$\{(1,4),\ (2,5)\}.$$

Quindi ...

Ne consegue che l'operazione di branching genererà due nodi, un nodo $S_1(A_0, A_1)$ con

$$A_0 = \{(1,3); (4,5); (1,4)\} \quad A_1 = \{(1,5); (2,4)\}$$

e l'altro nodo $S_2(A_0, A_1)$ con

$$A_0 = \{(1,3); (4,5); (2,5)\} \quad A_1 = \{(1,5); (2,4); (1,4)\}$$

Nota bene

Con questa regola di branching i nodi figli di un dato nodo continueranno ad essere sottinsiemi della regione ammissibile di forma $S(A_0, A_1)$.

Infatti, rispetto al nodo padre il primo nodo figlio aggiungerà l'arco (i^1, j^1) in A_0, il secondo nodo figlio aggiungerà l'arco (i^1, j^1) in A_1 e l'arco (i^2, j^2) in A_0, il terzo nodo figlio aggiungerà gli archi (i^1, j^1) e (i^2, j^2) in A_1 e l'arco (i^3, j^3) in A_0, e così via.