

METODI ED ALGORITMI DI OTTIMIZZAZIONE PER IL PROBLEM SOLVING

Docente:
Aristide Mingozzi

Adattamento:
Edoardo Rosa

Anno Accademico 20014-2015

CONTENTS

1	Modelli e formulazioni matematiche	1
1.1	The Traveling Salesman Problem	1
1.2	Formulazioni Matematiche del TSP	1
1.2.1	TSP asimmetrico	2
1.2.2	TSP simmetrico	2
1.2.3	Eliminazione subtours di Miller, Tucker, Zemlin (1960)	3
1.2.4	Il Traveling salesman problem con time windows (TSPTW)	4
1.3	Project scheduling with resource constraints (PSR)	5
1.3.1	Esempio di PSR	5
2	Introduzione alla programmazione lineare a numeri interi	6
2.1	6
A	Prova	7
A.1	Pippo	7

LIST OF FIGURES

1.1	Grafo orientato	3
1.2	Grafo H delle precedenze	5

Copertina: http://commons.wikimedia.org/wiki/File:Minimum_spanning_tree.svg

LIST OF TABLES

CHAPTER 1

MODELLI E FORMULAZIONI MATEMATICHE

1.1 The Traveling Salesman Problem

Il Traveling Salesman Problem (TSP) è il problema più noto dell'ottimizzazione combinatoria. Siano date n città e i costi c_{ij} per andare dalla città i alla città j . Si vuole determinare un cammino che parte da una città (diciamo i_1), visitare una ed una sola volta tutte le rimanenti città e terminare nella città di partenza i_1 . Inoltre si vuole che il costo di tale cammino sia minimo.

Ha molteplici applicazioni pratiche e teoriche perché è la struttura di molti problemi pratici. Si è soliti modellare il TSP come segue:

- è dato un grafo orientato (o non orientato) $G = (N, A)$ dove N è un insieme di n vertici e A è un insieme di m archi.

Ad ogni arco $(i, j) \in A$ è associato un costo c_{ij} .

Un circuito hamiltoniano di G è un circuito che passa per ogni vertice una ed una sola volta.

Il costo di un circuito hamiltoniano di G è pari alla somma dei costi degli archi che compongono il circuito;

- il problema del TSP è di trovare un grafo G , con una data matrice dei costi $[c_{ij}]$, un circuito hamiltoniano di costo minimo.

1.2 Formulazioni Matematiche del TSP

In letteratura esistono molteplici (e a volte fantasiose) formulazioni del TSP.

Presentiamo le due formulazioni più note e su cui si basano i metodi esatti più efficienti.

1.2.1 TSP asimmetrico

I costi c_{ij} non verificano $c_{ij} = c_{ji} \forall i, j$ con $i < j$.

Sia x_{ij} una variabile (0 – 1) associata ad ogni arco $(i, j) \in A$ dove $x_{ij} = 1$ se l'arco (i, j) è nella soluzione ottima e $x_{ij} = 0$ altrimenti.

$$\text{Min} \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij} \quad (1.1)$$

$$\text{s.t.} \sum_{i \in N} x_{ij} = 1, \quad \forall j \in N \quad (1.2)$$

$$\sum_{j \in N} x_{ij} = 1, \quad \forall i \in N \quad (1.3)$$

$$\sum_{i \in S} \sum_{j \in N \setminus S} x_{ij} \geq 1, \quad \forall S \subset N \quad (1.4)$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in A \quad (1.5)$$

Il vincolo 1.4 impone che ogni soluzione ammissibile debba contenere almeno un arco (i, j) con $i \in S$ e $j \in N \setminus S$ per ogni sottoinsieme S di N . Un'alternativa al vincolo 1.4 è:

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1, \quad \forall S \subset N \quad (1.4')$$

1.2.2 TSP simmetrico

Sia dato un grafo non-orientato $G = (N, A)$ con $c_{ij} = c_{ji}, \forall i, j \in N$.

Gli archi di A sono numerati da 1 a m . L'arco di indice l corrisponde a (α_l, β_l) con $\alpha_l < \beta_l$.

A_i è il sottoinsieme degli indici degli archi che incidono sul vertice i :

$$A_i = \{l : l = 1, m \text{ s.t. } \alpha_l = i \text{ or } \beta_l = i\}$$

Per una dato $S \in N$ e $\bar{S} = N \setminus S$ indichiamo con (S, \bar{S}) il sottoinsieme degli indici degli archi per cui $\alpha_l \in S$ e $\beta_l \in \bar{S}$ oppure $\alpha_l \in \bar{S}$ e $\beta_l \in S$.

Ad ogni arco di indice l è associato un costo $d_l = c_{\alpha_l \beta_l}$ e $x_l \in \{0, 1\}$ è una variabile che vale 1 se e solo se l'arco di indice l è nella soluzione ottima.

$$\text{Min} \sum_{l=1} d_l x_l \quad (1.6)$$

$$\text{s.t.} \sum_{l \in A_i} x_l = 2, \quad \forall i \in N \quad (1.7)$$

$$\sum_{l \in (S, \bar{S})} x_l \geq 1, \quad \forall S \subset N \quad (1.8)$$

$$x_l \in \{0, 1\}, \quad l = 1, \dots, m \quad (1.9)$$

1.2.3 Eliminazione subtours di Miller, Tucker, Zemlin (1960)

Sia u_i una variabile intera il cui valore rappresenta la posizione che il vertice i occupa nel tour.

Es. tour (1,4,5,3,2,1) per TSP con $n=5$ vertici, si ha $u_1 = 1, u_2 = 5, u_3 = 4, u_4 = 2, u_5 = 3$

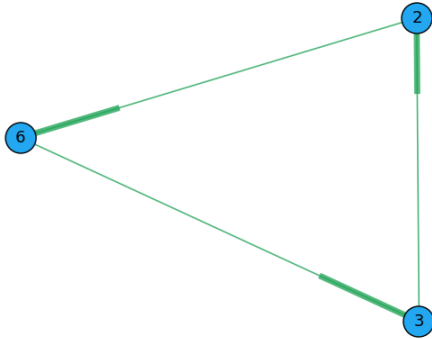
Miller, Tucker e Zemlin propongono in alternativa a:

$$\sum_{i \in S} \sum_{j \in N \setminus S} x_{ij} \geq 1, \quad \forall S \subset N \quad (*)$$

hanno imposto i seguenti vincoli:

$$u_i - u_j + n x_{ij} \leq n - 1, \quad i = 1, \dots, n, \quad j = 2, \dots, n \quad (1.10)$$

Ogni tour hamiltoniano soddisfa questi vincoli e ogni subtour li viola.



$$u_2 - u_6 + n \cdot x_{2,6} \leq n - 1$$

$$u_6 - u_3 + n \cdot x_{6,3} \leq n - 1$$

$$u_3 - u_2 + n \cdot x_{3,2} \leq n - 1$$

↓

$$3n \leq 3(n - 1)$$

Figure 1.1: Grafo orientato

1.2.4 Il Traveling salesman problem con time windows (TSPTW)

È una variante del TSP che ha molte applicazioni.

Sia dato un grafo orientato $G = (V, A)$ di $n + 1$ vertici ($V = \{0, 1, \dots, n\}$).

Ad ogni arco $(i, j) \in A$ sono associati

- un costo $c_{ij} \geq 0$
- un tempo di percorrenza $\theta_{ij} \geq 0$

Ad ogni vertice è associato un intervallo $[r_i, d_i]$ chiamato "time window" che rappresenta l'orario in cui il vertice i può essere visitato dal "salesman".

Ovvero il salesman può visitare i ad ogni tempo $t \in \mathbb{Z}^+$ con $r_i \leq t \leq d_i$.

Il problema consiste nel trovare una sequenza dei vertici di G che parte dal vertice 0 al tempo 0 e finisce al nodo 0 tale che sia il minimo il costo del circuito e il tempo di arrivo al nodo i sia nell'intervallo $[r_i, d_i]$, $\forall i \in V$.

Si consideri la sequenza $(0, i, \dots, i_{k-1}, i_k, \dots, i_n, 0)$ e sia t_{i_k} il tempo di arrivo al vertice i_k , $k = 0, 1, \dots, n + 1$.

I tempi di arrivo sono calcolati come:

$$t_0 = 0 \quad (1.11)$$

$$t_{i_k} = \max\{t_{i_{k-1}} + \theta_{i_{k-1} i_k}, r_{i_k}\} \quad (1.12)$$

1.2.4.1 Formulazione del TSPTW

Sia x_{ij} una variabile binaria intera che assume il valore 1 se il vertice i è visitato immediatamente prima di j e $x_{ij} = 0$ altrimenti.

$$\text{Min} \quad \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (1.13)$$

$$\text{s.t.} \quad \sum_{i \in A_j^-} x_{ij} = 1, \quad \forall j \in V \quad (1.14)$$

$$\sum_{j \in A_i^+} x_{ij} = 1, \quad \forall i \in V \quad (1.15)$$

$$t_i + \theta_{ij} - t_j \leq M(1 - x_{ij}), \quad \forall (i, j) \in A, j \neq 0 \quad (1.16)$$

$$t_i \leq d_i, \quad \forall i \in V \quad (1.17)$$

$$t_i \geq r_i, \quad \forall i \in V \quad (1.18)$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in A \quad (1.19)$$

$$t_i \in \mathbb{N}^+, \quad \forall i \in V \quad (1.20)$$

dove

$$A_i^+ = \{j \in V : (i, j) \in A\}$$

$$A_i^- = \{j \in V : (i, j) \in A\}$$

M un intero grande a piacere

$$r_0 = d_0 = 0$$

1.3 Project scheduling with resource constraints (PSR)

È dato un insieme $\mathbb{X} = \{1, \dots, n\}$ di n jobs.

Sono disponibili m risorse dove ogni risorsa k ha una disponibilità b_k ad ogni istante del periodo di scheduling.

Ogni job i ha un tempo di processo d_i e la sua esecuzione, una volta iniziata, non può essere interrotta.

Il job i per essere eseguito richiede b_{ik} unità della risorsa k per ciascun intervallo di tempo in cui rimane in esecuzione.

È dato un grafo $G = (X, H)$ di precedenze, dove ogni arco $(i, j) \in H$ impone che il job j può iniziare solo dopo che il job i è stato completato.

- Si vuole determinare il tempo di inizio di processo di ogni job in modo che siano soddisfatti i vincoli di precedenza, i vincoli sulle risorse e sia minima la durata complessiva del progetto

1.3.1 Esempio di PSR

Siano dati $n = 11$ jobs e $m = 3$ risorse con $b_1 = b_2 = b_3 = 4$ e un grafo H delle precedenze corrispondenti agli archi della figura 1.2.

Si osservi che i jobs 2 e 3 non possono essere eseguiti in parallelo poiché $r_{2,1} + r_{3,1} = 5 > b_1$!

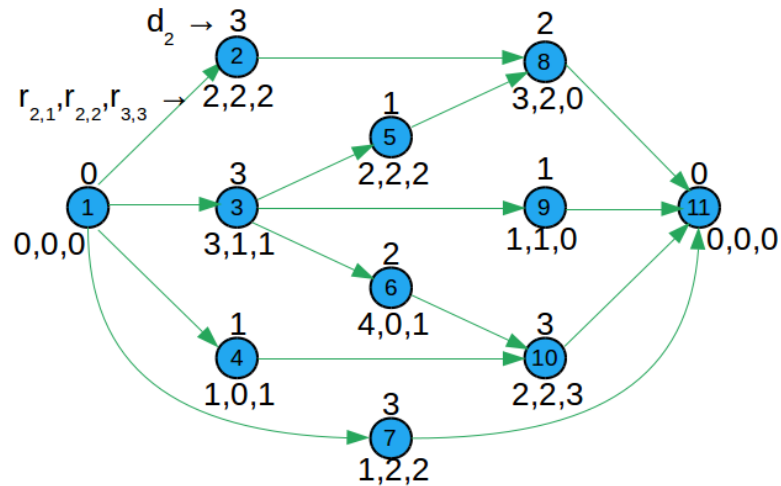


Figure 1.2: Grafo H delle precedenze

CHAPTER 2

INTRODUZIONE ALLA PROGRAMMAZIONE LINEARE A NUMERI INTERI

2.1

APPENDIX A

PROVA

A.1 Pippo