Linear ordering.

Il problema dell'ordinamento lineare (Linear Ordering Problem) consiste nel determinare una sequenza dei nodi di un dato grafo orientato, completo e pesato, in modo da minimizzare la somma dei pesi degli archi (i, j) tali che i precede j nell'ordinamento dei nodi.

Scrivere la formulazione del problema, classificarlo e risolvere l'esempio descritto nel seguito.

Discutere ottimalità e unicità della soluzione ottenuta.

Suggerimento: un grafo completo orientato non contiene cicli se e solo se non contiene cicli di ordine 3.

Esempio.

Il grafo ha 7 nodi ed è descritto dalla matrice dei pesi riportata in tabella 1.

	1	2	3	4	5	6	7
1	0	68	81	23	45	20	37
2	12	0	25	51	57	89	78
3	34	27	0	12	9	71	20
4	95	55	42	0	8	23	44
5	60	60	51	34	0	2	40
6	93	22	48	45	24	0	77
7	75	64	36	25	16	21	0

Tabella 1: Pesi degli archi (i, j): i=riga, j=colonna.

Soluzione.

Dati. Sono dati l'insieme indicizzato N dei nodi del digrafo e la matrice c (completa) dei costi degli archi.

Il problema si può formulare in diversi modi ed è interessante confrontare pregi e difetti di ciascuno.

Formulazione 1: selezione archi orientati.

Un primo modo di formulare il problema consiste nello scegliere quali archi debbano essere orientati da un nodo precedente ad un nodo successivo sicché il loro costo contribuisce a determinare il valore dell'obiettivo. I vincoli devono imporre che tali scelte corrispondano effettivamente ad una sequenza di nodi cioè che gli archi scelti formino un digrafo aciclico completo.

Variabili. Ad ogni arco (i, j) è associata una variabile binaria che vale 1 se e solo se l'arco (i, j) appartiene al digrafo aciclico, cioè se e solo se i precede j nella sequenza dei nodi.

Vincoli. Per imporre che gli archi scelti formino un digrafo aciclico completo è necessario imporre che per ogni coppia di nodi esista uno dei due archi che li collegano (digrafo completo) e che gli archi scelti non formino cicli (digrafo aciclico).

La prima condizione si esprime facilmente come

$$x_{ij} + x_{ji} = 1 \quad \forall i \in N, j \in N : i < j.$$

E' importante notare che il vincolo va imposto solo per coppie di indici diversi tra loro ($i \neq j$). Inoltre, per motivi di efficienza, sarebbe inutile scrivere lo stesso vincolo due volte con indici scambiati: perciò si può restringere la condizione a considerare ogni coppia una volta sola ($i < j$).

La seconda condizione, come suggerito nel testo, si può immporre rendendo inammissibile l'esistenza di cicli di ordine 3:

$$x_{ij} + x_{jk} + x_{ki} \leq 2 \quad i \in N, j \in N, k \in N : (i < j) \wedge (i < k) \wedge (j \neq k).$$

Per non scrivere ciascuno di questi vincoli tre volte con gli indici permutati, si può arbitrariamente imporre che i sia l'indice più piccolo dei tre. Anche in questo caso, per evitare di scrivere vincoli ridondanti è necessario imporre che i tre indici siano tutti diversi tra loro.

Obiettivo. La funzione obiettivo da minimizzare in questa formulazione è semplicemente la somma pesata delle variabili, ciascuna moltiplicata per il peso dell'arco corrispondente.

$$\text{minimize } z = \sum_{i \in N, j \in N : i \neq j} c_{ij} x_{ij}.$$

Si ottiene così un modello di PLI con n^2 variabili binarie e $n(n-1) + n(n-1)(n-2)/3$ vincoli lineari.

Formulazione 2: vincoli di precedenza.

Una formulazione simile alla precedente assume le stesse variabili e lo stesso obiettivo, ma, anziché proibire cicli di ordine 3, impone vincoli di transitività per assicurare che il grafo sia aciclico: se i precede j e j precede k allora i deve precedere k.

$$x_{ij} + x_{jk} \leq x_{ik} + 1 \quad \forall i \in N, j \in N, k \in N.$$

Poiché $x_{ik} = 1 - x_{ki}$, per sostituzione si possono ricavare i vincoli della formulazione precedente da questi (e viceversa).

Anche in questo caso si ha un modello di PLI con un numero $O(n^2)$ di variabili binarie ed un numero $O(n^3)$ di vincoli lineari.

Formulazione 3: sequenziamento dei nodi.

In questa formulazione, le variabili definiscono la sequenza dei nodi mentre i costi degli archi che contribuiscono a determinare il valore dell'obiettivo vengono determinati di conseguenza.

Variabili. Le variabili binarie x_{ip} indicano se il nodo $i \in N$ viene inserito in posizione $p \in P$. Le posizioni sono tante quante i nodi; quindi l'insieme P è un insieme indicizzato da 1 a n come l'insieme N (si potrebbe anche usare lo stesso insieme).

Vincoli. Le variabili di assegnamento devono essere soggette ai classici vincoli di assegnamento, per imporre che ogni nodo sia assegnato ad una posizione ogni posizione sia assegnata ad un nodo.

$$\sum_{i \in N} x_{ip} = 1 \quad \forall p \in P$$

$$\sum_{p \in P} x_{ip} = 1 \quad \forall i \in N.$$

Obiettivo. In questa formulazione risulta più complesso esprimere l'obiettivo. Si paga il costo di un arco (i, j) se e solo se le posizioni p e q a cui sono rispettivamente assegnati i nodi i e j sono tali che $p < q$. Una possibile formulazione dell'obiettivo è

$$\text{minimize } z = \sum_{i \in N, j \in N, p \in P, q \in P : p < q} c_{ij} x_{ip} x_{jq},$$

che però è non-lineare, a causa del prodotto tra le variabili binarie. Linearizzare l'obiettivo è possibile a patto di aumentare (di molto) il numero di variabili, introducendo variabili $0 \leq y_{ipjq} \leq 1$ per ogni coppia di variabili x con indici diversi, con gli ulteriori vincoli

$$x_{ip} + x_{jq} \leq y_{ipjq} + 1 \quad \forall i \in N, j \in N, p \in P, q \in P : p < q$$

per forzare le y a 1 quando entrambe le corrispondenti x valgono 1 e i vincoli

$$x_{ip} \geq y_{ipjq} \quad x_{jq} \geq y_{ipjq} \quad \forall i \in N, j \in N, p \in P, q \in P : p < q$$

per forzare le y a 0 quando almeno una delle due corrispondenti x vale 0. Le variabili y sono molte ma possono essere dichiarate come continue, non come binarie.

Si ottiene così un modello misto-intero con $n(n-1)$ variabili binarie ed un numero $O(n^4)$ di variabile continue e di vincoli lineari.

La soluzione ottima dell'esempio è la sequenza di nodi $(3, 1, 7, 4, 5, 6, 2)$ che costa 662.

La soluzione è garantita essere ottima; non è garantito che sia unica.