Ring loading.

Bisogna dimensionare una rete locale, composta da un certo numero di calcolatori collegati in anello da opportuni cavi. Ogni calcolatore può spedire messaggi ad ogni altro calcolatore in senso orario o in senso antiorario lungo l'anello. Ogni calcolatore instrada nello stesso senso (orario o antiorario) tutti i messaggi da lui emessi e aventi la stessa destinazione. È nota, in base a rilevamenti eseguiti in precedenza, la dimensione massima del traffico (quantità di dati per unità di tempo) che si prevede ogni calcolatore trasmetterà ad ogni altro. Si vuole decidere come instradare il traffico dei dati trasmessi da ciascun calcolatore in modo da minimizzare i costi di installazione della rete. Tali costi dipendono dal numero di archi che compongono l'anello e dal costo di ciascun arco, che è proporzionale alla capacità dell'arco. La capacità di ogni arco deve essere sufficiente a smaltire il traffico previsto su di esso. Tutti gli archi della rete devono avere la stessa capacità.

Formulare il problema, classificarlo e risolvere l'esempio indicato nel seguito, discutendo ottimalità e unicità delle soluzioni ottenute.

Esempio.

Considerare anello di 10 calcolatori numerati da 1 a 10 in senso orario.

	1	2	3	4	5	6	7	8	9	10
1	9	7	7	8	7	9	9	6	6	10
2	7	7	7	7	7	8	5	7	10	9
3	7	5	8	8	8	10	6	10	9	10
4	7	10	7	10	9	8	5	10	7	9
5	10	5	6	10	5	8	7	9	8	7
6	6	7	8	7	8	10	9	5	9	7
7	9	5	8	9	7	10	8	9	10	7
8	6	5	9	5	6	8	10	6	9	8
9	7	5	5	8	8	8	10	7	9	7
10	8	5	5	7	8	9	7	6	5	8

Tabella 1: Traffico previsto tra ogni coppia di calcolatori (riga = origine; colonna = destinazione).

Soluzione.

Si tratta di un problema applicativo vero, anche se di dimensioni molto maggiori di quelle proposte nell'esempio.

Dati. Sono dati un insieme N indicizzato di vertici di un grafo costituito semplicemente da un ciclo. E' data la matrice t_{ij} con un valore di traffico previsto da $i \in N$ a $j \in N$ per ogni coppia ordinata di nodi.

Variabili. Ad ogni coppia ordinata di calcolatori (i, j), con $i \neq j$, è associata una variabile binaria x_{ij}, che indica se i messaggi da i a j devono viaggiare in senso orario (0) o antiorario (1). Naturalmente la scelta di abbinare i valori 0 e 1 alle due direzioni è arbitraria. Qualunque delle due scelte va bene, basta poi essere coerenti quando si scrivono i vincoli.

Obiettivo. La funzione obiettivo è di tipo min-max, poiché si vuole minimizzare il carico (traffico) massimo che un arco deve sopportare. Perciò la formulazione ha una funzione obiettivo del tipo minimize z e $n = |N|$ vincoli del tipo $z \geq c_k$, per ogni arco $k = 1..n$, dove c_k indica il traffico previsto sull'arco k.

Vincoli. La difficoltà dell'esercizio sta nel rappresentare il carico su ogni arco $k = 1, \ldots, n$ in funzione delle variabili x_{ij}. Occorre stabilire una corrispondenza tra l'indice k dell'arco e gli indici dei nodi ai suoi estremi: ad esempio, stabiliamo che l'arco k colleghi i nodi k e $k + 1$, con l'eccezione dell'arco n che collega il nodo n col nodo 1.

Dato un arco k, distinguiamo 6 casi a seconda delle posizioni di i e j:

1. $i < j \leq k$: t_{ij} usa l'arco k se e solo se $x_{ij} = 1$.
2. $j < i \leq k$: t_{ij} usa l'arco k se e solo se $x_{ij} = 0$.
3. $i \leq k < j$: t_{ij} usa l'arco k se e solo se $x_{ij} = 0$.
4. $j \leq k < i$: t_{ij} usa l'arco k se e solo se $x_{ij} = 1$.
5. $k < i < j$: t_{ij} usa l'arco k se e solo se $x_{ij} = 1$.
6. $k < j < i$: t_{ij} usa l'arco k se e solo se $x_{ij} = 0$.

Il traffico totale sull'arco k è quindi dato dalla somma di sei tipi di contributi, tre dei quali esistono se e solo se $x_{ij} = 1$ e agli altri tre se e solo se $x_{ij} = 0$. Pertanto i corrispondenti valori di t_{ij} vengono sommati moltiplicando gli uni per x_{ij} e gli altri per $(1 - x_{ij})$.

Si tratta di un problema di PLI con variabili binarie.

L'ottimalità è garantita, l'unicità no.