Esercizio 1: Portavalori

Poiché il problema richiede di decidere quali furgoni e quali guardie utilizzare, il modello include una variabile binaria per ogni furgone e una variabile binaria per ogni guardia. Indichiamo con *y(f)* la variabile di ogni furgone *f=1..3* e con *x(g)* la variabile per ogni guardia *g=1..3*.

Per esprimere i costi è necessario tenere conto sia del tempo di viaggio che del tempo di caricamento. Il primo è fisso, il secondo dipende da quanto tempo impiegano le guardie a caricare i valori. Il costo dovuto al tempo di viaggio è quindi esprimibile in modo semplice con una funzione lineare, poiché è il prodotto del tempo di viaggio dato per la variabile binaria corrispondente al furgone o alla guardia. Il costo dovuto al tempo di caricamento invece sarebbe espresso dal prodotto tra la variabile binaria ed il tempo di caricamento che però è variabile. Ciò darebbe luogo ad una funzione obiettivo non-lineare. Per evitare la non-linearità occorre quindi usare tante variabili quante le guardie e i furgoni, che rappresentano ciascuna il tempo impiegato da quella guardia e da quel furgone per le operazioni di caricamento. Indichiamo tali variabili con *tempof(f)* per ogni furgone *f=1..3* e con *tempog(g)* per ogni guardia *g=1..3*. Detto *time* il tempo complessivo impiegato dalle guardie e dai furgoni impiegati, bisogna imporre che *tempof(f)* e *tempog(g)* siano uguali a time per tutti i furgoni e le guardie effettivamente impiegati e siano invece 0 per gli altri. Questo effetto si ottiene con i vincoli seguenti:

$$tempof(f) \geq time - MaxTime(1 - y(f)) \; \forall f$$
$$tempog(g) \geq time - MaxTime(1 - x(g)) \; \forall g$$

Infatti, per *x=0* o per *y=0* il vincolo diventa ridondante (scegliendo un valore di *MaxTime* grande abbastanza) e la corrispondente variabile *tempof* (o *tempog*) resta vincolata inferiormente solo dalla condizione di non-negatività. Così si ottiene di consentire il valore 0 per il tempo di caricamento di furgoni e guardie non utilizzati. Quando invece *x=1* o *y=1*, allora il vincolo impone che *tempof* (o *tempog*) sia almeno pari a *time*.

Un valore "abbastanza grande" di *MaxTime* è dato dal rapporto tra la quantità da caricare e la minima velocità di caricamento delle guardie.

Per definire *time* occorre considerare la capacità di caricamento totale delle guardie. Indicando con *v(g)* la velocità di caricamento di ogni guardia *g=1..3* e con *Q* la quantità totale da caricare, si potrebbe definire direttamente

$$time * \sum_{g=1}^{3} (x(g) * v(g)) = Q$$

ma anche in questo caso otterremmo un vincolo non-lineare. Per evitarlo si può introdurre una variabile, che indichiamo con *n(g)*, che rappresenta la quantità caricata da ogni singola guardia, e poi imporre i vincoli "min-max":

$$time \geq \frac{n(g)}{v(g)} \; \forall g = 1..3$$

e

$$\sum_{g=1}^{3} n(g) = Q$$

L'effetto è quello di ripartire *Q* in tante quantità *n(g)* in modo da minimizzare il massimo rapporto *n(g)/v(g)*. Ovviamente la minimizzazione del massimo rapporto si ottiene ripartendo le quantità *n(g)* in modo che i rapporti *n(g)/v(g)* risultino tutti uguali (e tutti uguali a *time*). Il tutto si ottiene usando solo vincoli lineari.

Per evitare che sia assegnata una quantità *n(g)* positiva ad una guardia *g* non utilizzata, bisogna aggiungere vincoli del tipo:

$$n(g) \leq Q * x(g) \; \forall g = 1..3$$

Per completare il modello bisogna anche imporre che la capacità dei veicoli scelti sia sufficiente

$$\sum_{f=1}^{3} (cap(f) * y(f)) \geq Q$$

e che il numero di guardie, che devono guidare, non sia inferiore al numero di veicoli:

$$\sum_{g=1}^{3} x(g) \geq \sum_{f=1}^{3} y(f)$$

Il modello risultante è di PLI ed è riportato nel file Lingo PORTAVAL.LG4. La soluzione ottima è nel file PORTAVAL.LGR. Nel caso di formulazione non-lineare, non si avrebbe la garanzia di ottimalità del minimo locale calcolato.