

Carte prepagate.

Un'azienda manda i propri dipendenti in giro per la città, per visitare alcuni clienti. La sequenza delle visite per ogni dipendente è data. Ogni dipendente si sposta nella città con la metropolitana, usando una carta prepagata virtuale messa a disposizione dall'azienda che gestisce il trasporto urbano nella città. Tutte le carte hanno un uguale massimo importo e non possono essere ricaricate. Il costo di ogni corsa è noto, è un numero intero positivo ed in generale dipende dalla distanza da percorrere. Il credito sulla carta viene consumato all'uscita dalla metropolitana, dopo ogni corsa. E' possibile iniziare una corsa con una carta se e solo se il credito residuo sulla carta è positivo. La corsa che esaurisce il credito residuo della carta può costare anche di più del credito residuo. Un dipendente può trasferire ad un collega la sua carta virtuale in qualsiasi momento e gratuitamente. Nuove carte possono essere acquistate in qualsiasi momento da qualunque dei dipendenti dell'azienda.

Si vuole anzitutto minimizzare il numero di carte necessarie e secondariamente si vuole minimizzare il numero di trasferimenti di carta da un dipendente all'altro.

Formulare il problema e classificarlo. Risolvere l'esempio descritto nel file `CartePrepagate.txt`.

Esempio.

Il numero di dipendenti è 3, indicati da A, B e C.

	1	2	3	4
A	10	9	8	25
B	30	30	10	.
C	15	4	14	3

Tabella 1: Costi delle corse di ciascun dipendente (euro).

La carica iniziale di ogni carta è pari a 25 euro.

Soluzione.

I vincoli di capacità, definiscono una variante particolare del problema di bin packing. Tuttavia, poiché è essenziale identificare l'ultimo elemento che entra in ogni bin, e poiché inoltre esistono vincoli di precedenza dovuti a sequenze prefissate di corse per ogni persona, non bastano i vincoli di assegnamento e di capacità come nel problema di bin packing, ma è necessario anche definire sequenze di corse. Non è strettamente necessario un ordinamento totale di tutte le corse, ma per lo meno un ordinamento delle corse per ogni carta.

Dati. Sia D l'insieme indicizzato dei dipendenti e sia C_d l'insieme (indicizzato) delle corse per ogni dipendente. Sia a_{dc} il costo di ogni corsa $c \in C_d, d \in D$. Sia B il valore iniziale della carica su ogni carta. Sia K l'insieme (indicizzato) delle carte. Per stimare la sua cardinalità si può assumere come upper bound, ad esempio, il numero totale di corse.

Variabili. Come nel problema di bin packing, serve una variabile binaria associata ad ogni carta per indicare se la carta è usata (1) o no (0). Indichiamo tale variabile con $z_k \forall k \in K$.

Come nel problema di bin packing serve definire gli assegnamenti delle corse alle carte. Usiamo quindi variabili binarie $y_{dck} \forall d \in D, \forall c \in C_d, \forall k \in K$.

Per rappresentare la sequenza delle corse per ogni carta, si possono utilizzare variabili binarie con 5 indici: $x_{d_1 c_1 d_2 c_2 k} \forall d_1 \in D, c_1 \in C_{d_1}, d_2 \in D, c_2 \in C_{d_2}, k \in K$ che indicano che la carta $k \in K$ viene usata consecutivamente dal dipendente d_1 per la sua corsa c_1 e poi dal dipendente d_2 per la sua corsa c_2 . I dipendenti d_1 e d_2 possono essere uguali o diversi. Queste variabili consentono di definire cammini, uno per ogni carta utilizzata, sul grafo in cui le corse sono i nodi.

Per ciascuno di questi cammini è necessario identificare l'ultimo nodo, cioè l'ultima corsa eseguita con la corrispondente carta. Per identificarla introduciamo una apposita variabile binaria $u_{dck} \forall d \in D, \forall c \in C_d, \forall k \in K$. Per simmetria, può essere utile introdurre anche una variabile che identifica la prima corsa effettuata con ogni carta. $p_{dck} \forall d \in D, \forall c \in C_d, \forall k \in K$.

Vincoli. Come nel problema di bin packing, i vincoli di assegnamento impongono che ogni corsa sia effettuata con una carta.

$$\sum_{k \in K} y_{dck} = 1 \quad \forall d \in D, c \in C_d.$$

Come nel problema di bin packing bisogna anche imporre un vincolo di capacità su ogni carta. In questa variante del problema, l'ultima corsa consuma un solo euro.

$$\sum_{d \in D, c \in C_d} a_{dc}(y_{dck} - u_{dck}) \leq (B - 1)z_k \quad \forall k \in K.$$

Ogni carta utilizzata deve avere una prima e un'ultima corsa; le carte non utilizzate, invece, no.

$$\sum_{d \in D, c \in C_d} p_{dck} = \sum_{d \in D, c \in C_d} u_{dck} = z_k \quad \forall k \in K.$$

La prima e l'ultima corsa della carta $k \in K$ devono essere tra le corse assegnate ad essa:

$$p_{dck} \leq y_{dck} \quad \forall d \in D, \forall c \in C_d, \forall k \in K$$

$$u_{dck} \leq y_{dck} \quad \forall d \in D, \forall c \in C_d, \forall k \in K.$$

Per definire la prima e l'ultima corsa e le rispettive sequenze per ogni carta, si usano i vincoli di flusso come nel problema di cammino su grafo. Il grado entrante in ogni nodo deve essere uguale al grado uscente ad eccezione del primo nodo che non ha grado entrante e dell'ultimo nodo che non ha grado uscente.

$$\sum_{d_1 \in D, c_1 \in C_{d_1}: d_1 \neq d \vee c_1 \leq c} x_{d_1 c_1 d c k} = y_{dck} - p_{dck} \quad \forall d \in D, \forall c \in C_d, \forall k \in K.$$

$$\sum_{d_2 \in D, c_2 \in C_{d_2}: d_2 \neq d \vee c_2 \geq c} x_{d c d_2 c_2 k} = y_{dck} - u_{dck} \quad \forall d \in D, \forall c \in C_d, \forall k \in K.$$

Per completare il modello è necessario vietare i sotto-cicli, che potrebbero formarsi a causa delle sequenze date, dal momento che non ci sono vincoli che colleghino i cammini relativi a carte diverse.

Un possibile modo di impedire i sottocicli senza dover definire un ordinamento totale su tutte le corse è ricorrere ai vincoli della formulazione MTZ del TSP. Introduciamo variabili intere v_{dc} che indicano la posizione di ogni corsa nella sua sequenza. I valori di queste variabili sono vincolati tra 1 e M , dove $M = \sum_{d \in D} |C_d|$. I sottocicli vengono quindi impediti imponendo

$$v_{d_1 c_1} + 1 \leq v_{d_2 c_2} + M(1 - \sum_{k \in K} x_{d_1 c_1 d_2 c_2 k}) \quad \forall d_1 \in D, c_1 \in C_{d_1}, d_2 \in D, c_2 \in C_{d_2} : d_1 \neq d_2 \vee c_1 \neq c_2$$

per corse consecutive effettuate da dipendenti diversi e

$$v_{dc-1} + 1 \leq v_{dc} \quad \forall d \in D, c \in C_d : c > 1$$

per corse consecutive effettuate dallo stesso dipendente.

Un tocco di classe finale consiste nel ridurre la simmetria presente nel modello, introducendo ad esempio questi vincoli

$$z_k \leq z_{k-1} \quad \forall k \in K : k > 1.$$

Obiettivi. Il primo obiettivo è la minimizzazione del numero di carte utilizzate, come nel bin packing problem:

$$\text{minimize } w_1 = \sum_{k \in K} z_k.$$

Nell'esempio dato, il valore ottimo è $w_1^* = 4$.

Il secondo obiettivo è la minimizzazione del numero di trasferimenti di carte tra dipendenti:

$$\text{minimize } w_2 = \sum_{d_1 \in D, c_1 \in C_{d_1}, d_2 \in D, c_2 \in C_{d_2}, k \in K: d_1 \neq d_2} x_{d_1 c_1 d_2 c_2 k}$$

e deve essere ottimizzato col vincolo di non peggiorare il primo obiettivo:

$$\sum_{k \in K} z_k \leq w_1^*.$$

Nell'esempio dato, il valore ottimo è $w_2^* = 3$.

Classificazione. Il modello risultante è di PLI, con variabili sia intere che binarie. La soluzione calcolata dai solutori è quindi garantita essere ottima; non è garantita l'unicità.