Il problema di Crew Scheduling*

Appunti per il corso di
Analisi e Ottimizzazione dei Processi di Produzione
Università degli Studi "Tor Vergata.

2 ottobre 2003

1 Soluzione di problemi di PL per generazione di colonne

In molti problemi di interesse applicativo che possono essere modellati (almeno in prima approssimazione) con la Programmazione Lineare accade che le matrici dei vincoli degli associati modelli lineari contengano moltissime colonne, a fronte di un numero assai più limitato di righe (ad esempio, in alcuni problemi il numero di colonne è dell'ordine di un esponenziale nel numero delle righe). Per fissare le idee, consideriamo un tale modello lineare P, scritto in forma standard. Sia A la matrice dei vincoli, N l'insieme degli indici delle colonne di A ($|N| = n$), m il numero di righe di A ($n >> m$), c il vettore dei costi e b il vettore dei termini noti. Il problema P è quindi formulato come segue (eqno13.2):

$$\begin{array}{ll} \min & c^T x \\ & \sum_{j \in N} a^{(j)} x_j = b \\ & x \geq \mathbf{0}_n, \end{array} \tag{1}$$

dove si è indicato con $a^{(j)}$ la generica colonna della matrice A.

Si osservi che, nonostante la dimensione della matrice dei vincoli ($m \times n$) possa essere molto elevata, la dimensione di ognuna delle sue basi ($m \times m$, se il problema è ammissibile e nessun vincolo è ridondante) è sufficientemente piccola.

Per la soluzione di P è conveniente utilizzare un approccio di tipo iterativo, denominato "metodo della generazione di colonne," la cui caratteristica fondamentale è quella di operare, ad ogni iterazione, solo su un sottoinsieme delle colonne di A, "generando" nuove colonne solo quando necessario. Più formalmente si può descrivere il metodo come segue:

- 1. Viene selezionato un opportuno sottoinsieme delle colonne di A; sia $N^{(1)}$ l'insieme degli indici delle colonne scelte;
- 2. viene risolto il problema di programmazione lineare $P^{(1)}$ (*"problema principale"*) associato alla sottomatrice $A^{(1)}$ di A costituita dalle colonne indicizzate da $N^{(1)}$; sia x^* la soluzione ottima del problema $P^{(1)}$ e $\bar{x}$ il vettore a n componenti definito da

$$\bar{x}_j = \begin{array}{ll} x_j^* & \text{se } j \in N^{(1)}; \\ 0 & \text{altrimenti;} \end{array}$$

- 3. Viene invocato un opportuno "oracolo di generazione di colonne" (*"sottoproblema"*). Tale oracolo verifica se $\bar{x}$ è soluzione ottima di P; se è questo il caso si termina l'esecuzione;

*Queste note sono tratte dal testo "APPUNTI SUL PROBLEMA DI SET-COVERING di P. Nobili, IASI CNR, Roma, 5 Dicembre 1996.

- 4. altrimenti, l'oracolo fornisce ("genera") una colonna $a^{(j')}$ di A (con j' in $N - N^{(1)}$) candidata ad appartenere alla base ottima di A. L'indice j' viene aggiunto all'insieme $N^{(1)}$ e si torna al passo 2.

Specifichiamo ora i vari passi dell'algoritmo. Innanzitutto occorre definire il criterio di scelta dell'insieme iniziale $N^{(1)}$. Affinché il problema principale $P^{(1)}$ sia ammissibile, occorre che $N^{(1)}$ contenga almeno una base ammissibile di A. Per garantire ciò, si possono applicare criteri analoghi a quelli adottati nella fase 1 del metodo del simplesso. In particolare, se per porre P in forma standard è stato necessario aggiungere variabili di slack / surplus (è il caso, ad esempio, dei rilassamenti lineari dei problemi di set-covering) è possibile scegliere $N^{(1)}$ in modo da contenere gli indici di tali variabili. Altrimenti, può essere necessario aggiungere al problema variabili artificiali, con costo opportunamente elevato.

Consideriamo ora l'oracolo di generazione delle colonne. Assumiamo che x^* sia una soluzione di base e che B (sottomatrice di $A^{(1)}$) sia la associata matrice di base ($x^* = B^{-1}b$). Sia inoltre $u^* = (B^{-1})^T c_B$ la soluzione ottima del duale di P, avendo indicato con c_B il vettore costituito dalle componenti di c associate alle colonne di B. Si osservi che $\bar{x}$ costituisce una soluzione di base ammissibile per il problema P, essendo ancora B (considerata ora come sottomatrice di A) la associata matrice di base. Dalla teoria della Programmazione Lineare, sappiamo che $\bar{x}$ è soluzione ottima di P se i costi ridotti $\bar{c}_j = c_j - (u^*)^T a^{(j)}$ sono non negativi per ogni $j \in N$. Possiamo assumere che $\bar{c}_j \geq 0$ per $j \in N^{(1)}$. Sappiamo inoltre che, se per qualche indice j il costo ridotto $\bar{c}_j$ è negativo, la corrispondente colonna $a^{(j)}$ è candidata ad appartenere alla base ottima. Queste osservazioni suggeriscono il seguente algoritmo per l'oracolo di generazione di colonne:

- (i) Si risolva il problema di ottimizzazione (sottoproblema)

$$z = \min_{j \in N} \{\bar{c}_j = c_j - (u^*)^T a^{(j)}\};$$

- (ii) se z è non-negativo, si dichiari che la soluzione $\bar{x}$ è ottima per P. Infatti, in tal caso non esistono variabili con costo ridotto negativo;

- (iii) altrimenti, sia j^* l'indice in corrispondenza al quale si ottiene il minimo; si fornisca la colonna a_{j^*}.

Nelle prossime sezioni si discuterà in dettaglio la realizzazione di un oracolo di generazione di colonne per un problema di grande interesse applicativo.

2 Il problema dei Turni del Personale

Sia data una lista di "servizi," ciascuno caratterizzato da un istante di inizio e un istante di fine e un insieme di "serventi," equivalenti tra loro, ciascuno in grado di espletare qualunque servizio della lista. Sia data, inoltre, una relazione binaria di "incompatibilità" tra servizi: in particolare, due servizi siano detti *incompatibili* se non possono essere espletati dallo stesso servente (sono incompatibili, ad esempio, due servizi i cui tempi di esecuzione si sovrappongono, in quanto i serventi possono espletare un solo servizio alla volta).

Il problema dei turni del personale consiste nell'assegnazione ottima dei servizi ai serventi, compilando per ciascuno di questi un programma dei servizi da compiere (turno) in modo da garantire che: (i) ciascun servizio venga svolto da un servente; (ii) ciascun turno sia composto di servizi non incompatibili e rispetti opportune regole di ammissibilità. Per fissare le idee, si descriverà ora una istanza del problema dei turni del personale che si presenta nella compilazione dei turni degli autisti in una azienda di trasporto pubblico urbano. L'esempio è liberamente tratto dal lavoro *"A Column Generation Approach to the Urban Transit Crew Scheduling Problem"*, M. Desrochers e F. Soumis, Transportation Science, 23 (1989).

Nel caso in esame, il problema consiste nel formare turni giornalieri. Ogni giorno, debbono essere effettuate un certo numero di "corse" per ciascuna linea. Per ogni corsa è stabilita una tabella oraria che

specifica l'ora di partenza dal primo capolinea, l'ora di arrivo al secondo capolinea, l'ora di passaggio in ciascuna delle fermate intermedie. Sono individuate le fermate (tra cui i capolinea) in cui è consentito effettuare un cambio di autista ("fermate utili"). La tratta di corsa compresa tra due fermate utili deve essere percorsa ininterrottamente dallo stesso autista. Il compito di guidare un autobus lungo una di tali tratte costituisce quindi il "servizio elementare" r_i $(i = 1, \ldots, m)$. Per ogni servizio elementare r_i è nota l'ora di inizio s_i, l'ora di fine e_i e le relative fermate utili di inizio e fine. Una successione di uno o più servizi consecutivi appartenenti alla stessa corsa costituisce un sotto-turno (denominato, in inglese, *"piece of work"*). Si noti che un sotto-turno è completamente specificato dal primo e ultimo servizio della sequenza. Si indichi con L_{ij} il sotto-turno avente r_i e r_j, rispettivamente, come primo e ultimo servizio.

Un sotto-turno L_{ij} si dice ammissibile se la sua durata $e_j - s_i$ è compresa in un opportuno intervallo. Un turno, in generale, sarà composto da diversi sotto-turni separati da pause di riposo. Si indicherà con B_{ji} una pausa di riposo compresa tra la fine del servizio r_j e l'inizio del servizio r_i (servizi appartenenti alla stessa o a diverse corse).

La pausa di riposo B_{ji} è ammissibile se la fermata finale del servizio r_j coincide con la fermata iniziale del servizio r_i e se la durata della pausa $s_i - e_j$ è compresa in un opportuno intervallo.

Per comodità di notazione, si introduce un'attività fittizia di "inizio-turno" P_i assegnata ad ogni turno il cui primo servizio è r_i $(i = 1, \ldots, m)$ e un'attività fittizia di "fine-turno" Q_j assegnata ad ogni turno il cui ultimo servizio è r_j $(j = 1, \ldots, m)$. L'inizio-turno P_i si dice ammissibile se r_i può effettivamente costituire il primo servizio di un turno (ad esempio, la sua ora di inizio s_i è precedente ad un opportuno istante di soglia).

Analogamente, il fine-turno Q_j si dice ammissibile se r_j può costituire l'ultimo servizio di un turno (ad esempio, la sua ora di fine e_j è successiva ad un opportuno istante di soglia). Con le posizioni fatte, un turno T_t è una sequenza $\{P_{i_1}, L_{i_1 j_1}, B_{j_1 i_2}, \ldots, L_{i_k j_k}, Q_{j_k}\}$ composta da un inizio-turno, una serie alternata di sotto-turni e pause di riposo, un ultimo sotto-turno e un fine-turno. Se l'inizio-turno, il fine-turno e tutti i sotto-turni e le pause di riposo della sequenza sono ammissibili, diremo che il turno è "corretto."

Infine, un turno corretto T_t si dice "ammissibile" se le seguenti proprietà sono soddisfatte:

- (1) il numero k di sotto-turni in T_t è minore di un massimo N_M;
- (2) la durata complessiva del turno T_t (denominata, in inglese, *"spread"*), $e_{j_k} - s_{i_1}$ è minore di un massimo S_M;
- (3) il carico complessivo di lavoro del turno T_t, dato da

$$\sum_{L_{ij} \in T_t} (e_j - s_i),$$

  è minore di un massimo W_M.

Definiamo ora il costo c_t di un turno T_t. Esso è costituito da diverse componenti:

- (1) Ad ogni inizio-turno ammissibile P_i è associato un costo $c_i^{(1)}$;
- (2) ad ogni fine-turno ammissibile Q_j è associato un costo $c_j^{(2)}$;
- (3) ad ogni sotto-turno ammissibile L_{ij} è associato un costo $c_{ij}^{(3)} = \mathrm{HR}\ (e_j - s_i)$, avendo indicato con HR il costo orario del lavoro (paga oraria);
- (4) ad ogni pausa di riposo ammissibile B_{ji} è associato un costo $c_{ji}^{(4)} = \mathrm{BR}\ (s_i - e_j)$, avendo indicato con BR il costo orario del riposo (remunerazione oraria della pausa).

Con tali posizioni, il costo del turno T_t è dato da

$$c_t = c^{(1)}_{i_1} + c^{(2)}_{j_k} + \sum_{L_{ij} \in T_t} c^{(3)}_{ij} + \sum_{B_{ji} \in T_t} c^{(4)}_{ji}.$$

Supponiamo ora di aver generato tutti i turni ammissibili, e sia A la matrice di incidenza servizi-turni. In altri termini, le righe di A (siano esse m) sono associate ai servizi r_i e le colonne (siano esse n) sono associate ai turni T_t. L'elemento a_{it} della matrice A, posto in riga i e colonna t è uguale a 1 se il servizio r_i è assicurato ("coperto") dal turno T_t, 0 altrimenti. Possiamo quindi formulare il problema dei turni del personale (*"Crew Scheduling Problem"*) come problema di set-covering:

$$(CSP) \qquad \begin{array}{ll} \min & c^T x \\ & Ax \geq \mathbf{1}_m \\ & x \in \{0,1\}^n. \end{array}$$

La soluzione ottima x^* di CSP è il vettore di incidenza di un sottoinsieme C dei turni ammissibili tale da garantire, a costo minimo, l'espletamento (copertura) di tutti i servizi. Si noti che è possibile che un servizio r_i sia coperto contemporaneamente da più turni in C. L'interpretazione di una tale situazione è la seguente: dei diversi autisti a cui è assegnato il servizio r_i uno solo lo esegue effettivamente, conducendo l'autobus; gli altri viaggiano come passeggeri. Si osservi, infatti, che può essere necessario ricorrere a tale evenienza per consentire agli autisti di riprendere il servizio da fermate diverse da quelle in cui lo avessero precedentemente interrotto.

La risoluzione di CSP è complicata dal fatto che la matrice A dei coefficienti è composta da moltissime colonne (tante quanti sono i turni ammissibili) e che non è sempre possibile generarla completamente a priori. Se si adotta un metodo di soluzione che prevede, come passo intermedio, di risolvere il rilassamento lineare di CSP (ad esempio il metodo del Branch & Bound) è conveniente utilizzare la tecnica di generazione di colonne introdotta nella precedente sezione. Occorre in questo caso impostare il "sottoproblema" (oracolo di generazione di colonne) consistente nella generazione di un turno ammissibile (colonna della matrice A) a "costo ridotto minimo." Nella prossima sezione si discuterà un algoritmo per la soluzione di tale problema.

3 Il sottoproblema : generazione di un turno ammissibile

Vogliamo ora applicare il metodo di generazione di colonne descritto in precedenza alla soluzione del rilassamento lineare del problema dei turni del personale CSP. Sia u^* il vettore della soluzione duale ottima del problema principale $P^{(1)}$, ottenuto al termine del passo 2 del metodo. Le componenti del vettore u^* sono associate ai servizi (righe di A). Consideriamo un turno ammissibile T_t, come descritto in sezione 14. Il suo costo ridotto, relativo a u^*, è dato dalla seguente espressione:

$$\begin{aligned} \bar{c}_t &= c_t - \textstyle\sum_{r_h \in T_t} u^*_h = c^{(1)}_{i_1} + c^{(2)}_{j_k} + \sum_{L_{ij} \in T_t} c^{(3)}_{ij} + \sum_{B_{ji} \in T_t} c^{(4)}_{ji} - \sum_{L_{ij} \in T_t} \sum_{r_h \in L_{ij}} u^*_h = \\ &= c^{(1)}_{i_1} + c^{(2)}_{j_k} + \textstyle\sum_{L_{ij} \in T_t} \left(c^{(3)}_{ij} - \sum_{r_h \in L_{ij}} u^*_h \right) + \sum_{B_{ji} \in T_t} c^{(4)}_{ji}. \end{aligned} \tag{2}$$

Formuleremo il problema di generare il turno ammissibile T_t avente minimo costo ridotto come un problema di cammino minimo con vincoli aggiuntivi su un opportuno grafo orientato. Il grafo orientato $G = (N, A)$ con insieme di nodi N e insieme di archi orientati A è definito come segue. L'insieme dei nodi $N = \{p, q\} \cup U \cup V$ è costituito da:

- un nodo "sorgente" p, rappresentante l'istante di inizio giornata;
- un nodo "pozzo" q, rappresentante l'istante di fine giornata;
- due famiglie U e V di nodi; ad ogni servizio r_i è associato un nodo $m_i \in U$ (rappresentante l'istante di inizio del servizio r_i) e un nodo $n_i \in V$ (rappresentante l'istante di fine del servizio r_i).

L'insieme degli archi orientati A è costituito da:

- un arco "iniziale" f_{pm_i} dal nodo p al nodo m_i per ogni inizio-turno ammissibile P_i;
- un arco "finale" f_{n_jq} dal nodo n_j al nodo q per ogni fine-turno ammissibile Q_j;
- un arco "in avanti" $f_{m_in_j}$ dal nodo m_i al nodo n_j per ogni sotto-turno ammissibile L_{ij};
- un arco "all'indietro" $f_{n_jm_i}$ dal nodo n_j al nodo m_i per ogni pausa di riposo ammissibile B_{ji}.

Ad ogni arco orientato di G associamo una "lunghezza" come segue:

- ad un arco "iniziale" f_{pm_i} attribuiamo la lunghezza $l_{pm_i} = c_i^{(1)}$;
- ad un arco "finale" f_{n_jq} attribuiamo la lunghezza $l_{n_jq} = c_j^{(2)}$;
- ad un arco "in avanti" $f_{m_in_j}$ attribuiamo la lunghezza $l_{m_in_j} = c_{ij}^{(3)} - \sum_{r_h \in L_{ij}} u_h^*$;
- ad un arco "all'indietro" $f_{n_jm_i}$ attribuiamo la lunghezza $l_{n_jm_i} = c_{ji}^{(4)}$.

Si osserva facilmente che il grafo G definito come sopra non contiene cicli orientati: infatti, ad ogni nodo del grafo è associato un istante di tempo e se esiste un arco dal nodo v_1 al nodo v_2 allora l'istante di tempo associato a v_1 è anteriore all'istante di tempo associato a v_2. Non esistendo cicli orientati, ogni cammino orientato sul grafo G ha lunghezza finita.

Dato il grafo sopra definito, possiamo immediatamente formulare il problema (rilassato) di determinare un turno *corretto* ottimo. Si osservi, infatti, che un cammino orientato P_t da p a q corrisponde univocamente ad un turno corretto T_t: infatti, esso è composto da un arco "iniziale" (associato ad un inizio-turno ammissibile), seguito da una sequenza alternata di archi "in avanti" (associati a sotto-turni ammissibili) e archi "all'indietro" (associati a pause di riposo ammissibili) e infine da un arco "finale" (associato ad un fine-turno ammissibile). Inoltre, la lunghezza del cammino è pari al costo ridotto del turno T_t, come è facile verificare. Si ha quindi che il problema di determinare un turno corretto a costo ridotto minimo equivale a quello di trovare il cammino orientato da p a q di lunghezza minima. Tale problema può essere formulato associando a ciascun arco f_{uv} una variabile binaria y_{uv} con l'interpretazione che l'arco appartiene al cammino minimo se e solo se la variabile ha valore 1 nella soluzione ottima. La formulazione di programmazione lineare intera che si ottiene è la seguente:

$$\begin{aligned} \min \quad & \sum_{f_{uv} \in A} l_{uv} y_{uv} \\ & \sum_{v:f_{uv}\in A} y_{uv} - \sum_{v:f_{vu}\in A} y_{vu} = \begin{cases} 1 & \text{se } u = p; \\ 0 & \text{se } u \in N - \{p,q\}; \\ -1 & \text{se } u = q; \end{cases} \quad u \in N \\ & y_{uv} \in \{0,1\}, \quad f_{uv} \in A. \end{aligned} \tag{3}$$

Per ottenere una formulazione del problema di trovare un turno *ammissibile* ottimo, occorre imporre ulteriori vincoli. Per individuarli, ragioniamo come segue. Consideriamo ciascuna delle tre grandezze da controllare (numero dei sotto-turni ammissibili contenuti nel turno, durata complessiva del turno, carico lavorativo del turno) come una "risorsa" $\mathcal{T}^{(l)}$ $(l = 1, 2, 3)$ che viene accumulata durante l'espletamento del turno. Così, per essere precisi, la quantità di risorsa $\mathcal{T}^{(1)}$ (numero dei sotto-turni ammissibili) è zero all'inizio del turno e si incrementa di una unità ogni volta che viene completato un sotto-turno ammissibile; la quantità di risorsa $\mathcal{T}^{(2)}$ (durata complessiva del turno) si incrementa, ogni volta che si completa un sotto-turno ammissibile o una pausa di riposo ammissibile, della relativa durata; la quantità di risorsa $\mathcal{T}^{(3)}$ (carico lavorativo del turno) si incrementa della relativa durata ogni volta che si completa un sotto-turno ammissibile. Un turno corretto è anche ammissibile se le quantità delle tre risorse accumulate a fine turno sono minori dei massimi stabiliti.

Dato un generico cammino $\mathcal{P}_t$ da p a q sul grafo G, corrispondente ad un turno corretto T_t, possiamo immaginare che le quantità delle tre risorse accumulate negli istanti significativi del turno (inizio o fine di attività) siano associate ai corrispondenti nodi del cammino.

Le precedenti osservazioni ci suggeriscono, per ottenere la formulazione desiderata, di procedere come segue.

Associamo a ciascun nodo v del grafo tre variabili $T_v^{(l)}$ $(l = 1, 2, 3)$. Per una soluzione ammissibile del modello, corrispondente ad un cammino da p a q associato ad un turno corretto, i valori assunti da tali variabili rappresenteranno le quantità delle tre risorse $\mathcal{T}^{(l)}$ accumulate nell'istante associato al nodo v, se esso appartiene al cammino; avranno significato indefinito se v non appartiene al cammino.

Inoltre, ad ogni arco f_{uv} del grafo G associamo tre quantità (costanti) $d_{uv}^{(l)}$ $(l = 1, 2, 3)$ con la seguente interpretazione: $d_{uv}^{(l)}$ rappresenta l'incremento della risorsa $\mathcal{T}^{(l)}$ nel passare dall'istante associato al noto u all'istante associato al nodo v, sotto l'ipotesi che l'arco f_{uv} appartenga al cammino ottimo. Le quantità $d_{uv}^{(l)}$ $(l = 1, 2, 3)$ sono definite come segue:

- se $f_{uv} \equiv f_{pm_i}$ è un arco "iniziale," $d_{uv}^{(l)} = 0$ $(l = 1, 2, 3)$;
- se $f_{uv} \equiv f_{n_j q}$ è un arco "finale," $d_{uv}^{(l)} = 0$ $(l = 1, 2, 3)$;
- se $f_{uv} \equiv f_{m_i n_j}$ è un arco "in avanti," $d_{uv}^{(1)} = 1$, $d_{uv}^{(2)} = d_{uv}^{(3)} = (e_j - s_i)$;
- se $f_{uv} \equiv f_{n_j m_i}$ è un arco "all'indietro," $d_{uv}^{(1)} = d_{uv}^{(3)} = 0$, $d_{uv}^{(2)} = (s_i - e_j)$.

Dobbiamo, infine, imporre i vincoli sulle variabili $T_v^{(l)}$:

- condizioni iniziali delle risorse,
$$T_p^{(1)} = T_p^{(2)} = T_p^{(3)} = 0;$$
- ammissibilità del turno,
$$T_q^{(l)} \leq b^{(l)},$$
avendo posto
$$\begin{aligned} b^{(1)} &= N_M, \\ b^{(2)} &= S_M, \\ b^{(3)} &= W_M; \end{aligned} \tag{4}$$
- incremento delle quantità di risorse accumulate lungo gli archi appartenenti al cammino ottimo,
$$(y_{uv} = 0) \bigvee (T_v^{(l)} = T_u^{(l)} + d_{uv}^{(l)}) \qquad l = 1, 2, 3; \quad f_{uv} \in A.$$

Si osservi che i vincoli introdotti al terzo punto impongono, per ogni arco $f_{uv} \in A$, che sia verificata una delle seguenti condizioni: o l'arco non appartiene al cammino ottenuto come soluzione del modello $(y_{uv} = 0)$ oppure la risorsa $\mathcal{T}^{(l)}$ $(l = 1, 2, 3)$ si incrementa nel passare dall'istante associato al nodo u all'istante associato al nodo v esattamente della quantità $d_{uv}^{(l)}$. Vincoli che, come in questo caso, impongono il verificarsi di una di due condizioni vengono detti *disgiuntivi*.

Siamo ora in grado di formulare il problema di determinare il turno ammissibile di costo ridotto minimo:

$$\begin{array}{ll} \min & \sum_{f_{uv} \in A} l_{uv} y_{uv} \\ & \sum_{v:f_{uv} \in A} y_{uv} - \sum_{v:f_{vu} \in A} y_{vu} = \begin{cases} 1 & \text{se } u = p; \\ 0 & \text{se } u \in N - \{p, q\}; \\ -1 & \text{se } u = q; \end{cases} \quad u \in N \end{array}$$

$$
\begin{gathered}
y_{uv} \in \{0,1\}, \quad f_{uv} \in A; \\
T_p^{(1)} = T_p^{(2)} = T_p^{(3)} = 0; \\
T_q^{(l)} \leq b^{(l)}; \\
(y_{uv} = 0) \bigvee (T_v^{(l)} = T_u^{(l)} + d_{uv}^{(l)}) \qquad l = 1,2,3; \quad f_{uv} \in A.
\end{gathered} \tag{5}
$$

Una generica soluzione ammissibile del sistema di vincoli dato corrisponde ad un cammino $P = (v_1 \equiv p, v_2, \ldots, v_{t-1}, v_t \equiv q)$ e ad una associata assegnazione di valori per le variabili $T_v^{(l)}$ che soddisfa, per ogni nodo v_k appartenente al cammino, alla relazione

$$
T_{v_k}^{(l)} = \sum_{i=2}^{k} d_{v_{i-1}v_i}^{(l)} \quad l = 1,2,3.
$$

In generale, ad ogni cammino $P_k = (v_1 \equiv p, v_2, \ldots, v_k)$ con estremi p e v_k possiamo associare la grandezza $T_{v_k}^{(l)}$ $(l = 1,2,3)$ definita come sopra. Essa corrisponde alla quantità di risorsa $\mathcal{T}^{(l)}$ accumulata fino all'istante corrispondente al nodo v_k dal turno corretto (parziale) rappresentato da P_k.

Nel seguito, per brevità, diremo semplicemente che $T_{v_k}^{(l)}$ è la quantità di risorsa $\mathcal{T}^{(l)}$ accumulata dal cammino P_k.

4 Soluzione del modello

Il modello di programmazione lineare intera disgiuntiva introdotto nella precedente sezione è, in generale, di difficile soluzione. Tuttavia, in pratica, la presenza di vincoli piuttosto stringenti sui cammini da p a q accettabili limita fortemente il numero di soluzioni ammissibili del modello, rendendo sufficientemente efficienti i metodi di enumerazione implicita. Se ne descriverà ora uno basato sulla programmazione dinamica.

Dato il grafo G descritto nella precedente sezione, per ogni nodo v di G si definisca come segue una funzione

$$
g_v : \mathbb{R}^3 \to \mathbb{R} :
$$

$g_v(T^{(1)}, T^{(2)}, T^{(3)})$ *sia la lunghezza del cammino più corto tra* p *e* v *che accumula quantità delle risorse* $\mathcal{T}^{(1)}$, $\mathcal{T}^{(2)}$, $\mathcal{T}^{(3)}$ *limitate superiormente rispettivamente da* $T^{(1)}$, $T^{(2)}$, $T^{(3)}$; $g_v(T^{(1)}, T^{(2)}, T^{(3)}) = \infty$ *se non esiste un tale cammino.*

Naturalmente, $g_p(T^{(1)}, T^{(2)}, T^{(3)}) = 0$ per qualunque tripla di valori non-negativi delle quantità $T^{(l)}$ $(l = 1,2,3)$. Inoltre $g_v(T^{(1)}, T^{(2)}, T^{(3)}) = \infty$ per ogni v se almeno una delle tre quantità $T^{(l)}$ $(l = 1,2,3)$ è negativa. Come è facile osservare, vale la seguente relazione di ricorrenza:

$$
g_v(T^{(1)}, T^{(2)}, T^{(3)}) = \min_{f_{uv} \in A} \{ g_u(T^{(1)} - d_{uv}^{(1)}, T^{(2)} - d_{uv}^{(2)}, T^{(3)} - d_{uv}^{(3)}) + l_{uv} \}.
$$

Supponiamo, per il momento, di voler solo determinare la lunghezza del cammino ottimo da p a q che soddisfa i vincoli del modello. Evidentemente, tale lunghezza è data da $g_q(N_M, S_M, W_M)$, e quindi il suo valore può essere calcolato usando la data relazione di ricorrenza. Si osservi che, sebbene le funzioni g_v siano definite su tutto $\mathbb{R}^3$, in pratica sarà necessario valutarle solo in un insieme discreto di punti. In particolare, supponendo per semplicità che le quantità N_M, S_M, W_M, $d_{uv}^{(l)}$ $(l = 1,2,3,\ f_{uv} \in A)$ siano tutte numeri interi, sarà necessario valutare le funzioni g_v solo per valori interi dei parametri. Inoltre, poiché l'insieme di interesse per i valori dei parametri è limitato superiormente e per parametri negativi le funzioni assumono valore infinito, sarà necessario valutare le funzioni g_v solo su di un insieme finito di valori.

Descriveremo ora un semplice algoritmo ricorsivo per calcolare il valore di $g_q(N_M, S_M, W_M)$. L'algoritmo farà uso di una tabella ausiliaria (TAB) a quattro indici destinata a contenere i valori assunti dalle funzioni $g_v(T^{(1)}, T^{(2)}, T^{(3)})$ $(v \in N)$ in corrispondenza ai diversi possibili valori dei parametri $T^{(1)}$, $T^{(2)}$, $T^{(3)}$. In

particolare, la posizione della tabella individuata dagli indici v, $T^{(1)}$, $T^{(2)}$, $T^{(3)}$ sarà associata al valore $g_v(T^{(1)}, T^{(2)}, T^{(3)})$. Inoltre, si utilizzerà il simbolo $\emptyset$ per marcare quelle locazioni della tabella corrispondenti a valori delle funzioni non ancora calcolati.

Il cuore dell'algoritmo è costituito dalla seguente funzione ricorsiva:

- **G** $(v, T^{(1)}, T^{(2)}, T^{(3)})$
  - *Input:* un nodo v e tre parametri interi, $T^{(1)}, T^{(2)}, T^{(3)}$.
  - *Output:* il valore di $g_v(T^{(1)}, T^{(2)}, T^{(3)})$.
- Se uno dei tre parametri $T^{(1)}, T^{(2)}, T^{(3)})$ è negativo, si termini l'esecuzione, fornendo in uscita il valore ∞.
- Altrimenti, per ogni nodo u per cui esiste l'arco $f_{uv} \in A$:
  1. (i) si ponga $\hat{T}^{(1)} \leftarrow T^{(1)} - d^{(1)}_{uv}$, $\hat{T}^{(2)} \leftarrow T^{(2)} - d^{(2)}_{uv}$, $\hat{T}^{(3)} \leftarrow T^{(3)} - d^{(3)}_{uv}$;
  2. (ii) si valuti la funzione $g_u(\hat{T}^{(1)}, \hat{T}^{(2)}, \hat{T}^{(3)})$. Tale valore è dato da TAB $[u, \hat{T}^{(1)}, \hat{T}^{(2)}, \hat{T}^{(3)}]$, se tale posizione è diversa da $\emptyset$, altrimenti sia ottenuto richiamando ricorsivamente la funzione G con parametri $(u, \hat{T}^{(1)}, \hat{T}^{(2)}, \hat{T}^{(3)})$;
  3. (iii) si ponga $h_{uv} \leftarrow g_u(\hat{T}^{(1)}, \hat{T}^{(2)}, \hat{T}^{(3)}) + l_{uv}$.
- Sia $\bar{u}$ il nodo per cui $h_{\bar{u}v}$ è minimo. Si ponga TAB $[v, T^{(1)}, T^{(2)}, T^{(3)}] \leftarrow h_{\bar{u}v}$ e si ritorni $g_v(T^{(1)}, T^{(2)}, T^{(3)}) \leftarrow h_{\bar{u}v}$.

Si può ora descrivere molto semplicemente l'algoritmo:

- (1) **Inizializzazione.** Per $v \in N - \{p\}$, $0 \le T^{(1)} \le N_M$, $0 \le T^{(2)} \le S_M$, $0 \le T^{(3)} \le W_M$, si ponga TAB $[v, T^{(1)}, T^{(2)}, T^{(3)}] \leftarrow \emptyset$. Per $0 \le T^{(1)} \le N_M$, $0 \le T^{(2)} \le S_M$, $0 \le T^{(3)} \le W_M$, si ponga TAB $[p, T^{(1)}, T^{(2)}, T^{(3)}] \leftarrow 0$.
- (2) **Passo ricorsivo.** Si valuti $g_q(N_M, S_M, W_M) \leftarrow$ G (q, N_M, S_M, W_M).
  Una semplice modifica dell'algoritmo descritto consente di ricavare il cammino ottimo da p a q. A tal fine si utilizzerà un vettore P indicizzato dai nodi del grafo e, come ultimo passo nell'esecuzione della funzione ricorsiva G, si aggiungerà l'assegnazione

$$P[v] \leftarrow \bar{u}.$$

  Il cammino ottimo sarà quindi alla fine ricostruito nel modo seguente:

- (3) si ponga $i \leftarrow 1$, $p_1 \leftarrow q$;
- (4) se $p_i = p$ si ponga $n \leftarrow i$ e si termini l'esecuzione;
- (5) altrimenti si ponga $u \leftarrow P[p_i]$, $i \leftarrow i + 1$, $p_i \leftarrow u$ e si torni al passo (4).

È facile convincersi che, alla fine dell'esecuzione dell'algoritmo, i nodi da p_1 a p_n rappresenteranno, in ordine inverso, i nodi del cammino ottimo da p a q.

Nota su un algoritmo di branch and bound per il Crew Scheduling Problem

Dispensa per il modulo di
"Analisi e Ottimizzazione dei Processi di Produzione"
Università di Roma "Tor Vergata"

a cura di Andrea Pacifici

A.A. 2004-05

Introduzione

In questa nota è illustrata una procedura di enumerazione implicita, basata sul branch and bound, per il problema di *crew scheduling* come introdotto da Nobili in [1]. La formulazione ivi proposta consiste in un particolare problema di *set covering*:

$$\min \left\{ c^T x : \sum_{j \in \mathcal{T}} A_j x_j \geq \mathbf{1}_m;\ x_j \in \{0,1\}\ \forall j \in \mathcal{T} \right\} \tag{1}$$

dove $\mathbf{1}_m$ indica il vettore ad m componenti tutte pari a 1 e $\mathcal{T}$ è l' insieme dei *turni ammissibili*. Nella maggioranza delle applicazioni reali, la cardinalità di $\mathcal{T}$ è tale da richiedere il ricorso ad una tecnica di generazione colonne per risolvere il rilassamento lineare del Problema (1):

$$\min \left\{ c^T x : \sum_{j \in \mathcal{T}} A_j x_j \geq \mathbf{1}_m;\ 0 \leq x_j \leq 1\ \forall j \in \mathcal{T} \right\} \tag{2}$$

Per la soluzione del Problema di programmazione lineare (2) ci riferiamo, nel seguito, alla tecnica di generazione colonne riportata in [1]: tale tecnica si basa sul calcolo di un cammino di costo minimo su una rete G opportunamente definita.

Branch and bound

Ciascun nodo dell'albero di enumerazione è associato ad un particolare sottoproblema di PLI, in cui un certo sottoinsieme $\bar{\mathcal{T}}$ di turni è stato selezionato per la soluzione e un altro sottoinsieme $\mathcal{S}$ è stato scartato da essa.

Il sotto-problema si può scrivere:

$$\min \left\{ c^T x : \sum_{j \in \mathcal{T} \setminus \mathcal{S}} A_j x_j \geq \mathbf{b}(\bar{\mathcal{T}});\ x_j \in \{0,1\}\ \forall j \in \mathcal{T} \setminus \mathcal{S} \right\} \tag{3}$$

dove $\mathbf{b}(\bar{\mathcal{T}})$ è un vettore di $\{0,1\}^m$ la cui componente i-ma è pari a 0 o 1 a seconda che il servizio corrispondente r_i sia, rispettivamente, coperto o meno nell'insieme dei turni selezionati $\bar{\mathcal{T}}$. A esempio, il (sotto-)problema al "nodo radice" corrisponde al problema originale (Problema (1)) dove $\bar{\mathcal{T}} = \mathcal{S} = \emptyset$. Il rilassamento lineare del Problema (3) è chiaramente ottenuto rilassando i vincoli di interezza della variabile x_j e ponendo $0 \leq x_j \leq 1$ per ogni $j \in \mathcal{T} \setminus \mathcal{S}$.

Per l'esecuzione del branch and bound è necessario, per ogni sotto-problema (ovvero ad ogni nodo dell'albero di enumerazione), determinare un *lower bound* attraverso la soluzione del corrispondente rilassamento lineare. Anche per un sotto-problema, a causa del numero molto elevato di variabili, la soluzione del corrispondente rilassamento lineare richiede di ricorrere ad un metodo del *simplesso dinamico*. A tale scopo ci si chiede se è possibile, e se sì in che modo, utilizzare l'oracolo di generazione colonne illustrato in [1] per il problema al "nodo radice". Nel paragrafo che segue vediamo come questo sia possibile utilizzando una particolare tecnica di *branching*, vale a dire di decomposizione dei sottoproblemi di tipo (3).

Un modo naturale di effettuare branching nell'albero di enumerazione,(il cosiddetto *branching binario* descritto nel paragrafo che segue) non rende agevole utilizzare l'oracolo di generazione colonne basato sul calcolo di un cammino minimo su una rete G opportunamente definita (si veda [1]). Nella sezione ancora successiva descriviamo una tecnica alternativa di branching (proposta in [2]) e le sue conseguenze nell'applicazione del metodo di generazione colonne.

Branching binario

Supponiamo che la variabile x_t ($t \in \mathcal{T}$) assuma valore frazionario nella soluzione del rilassamento lineare di un dato sottoproblema P_ℓ corrispondente al nodo i dell'albero di enumerazione. In tal caso, se il nodo non è sondato (*fathomed*) per altri motivi, si può decomporre il problema P_ℓ in due ulteriori sottoproblemi $P_{\ell 0}$ e $P_{\ell 1}$ in cui la variabile x_t viene fissata a 0 e 1, rispettivamente in $P_{\ell 0}$ e $P_{\ell 1}$. Nell'albero di enumerazione questo corrisponde a generare due ulteriori nodi $\ell 0$ e $\ell 1$ figli del nodo ℓ. In questo modo $\bar{\mathcal{T}}$ e $\mathcal{S}$ sono determinati dai sotto-insiemi delle variabili x_j che sono state fissate a 1 e a 0 rispettivamente. ribadiamo che tale tecnica rende difficile utilizzare l'oracolo di generazione colonne basato sul calcolo del cammino minimo: si tratterebbe infatti di inibire la soluzione (vale a dire un cammino sulla rete G) dall'assumere valori corrispondenti a turni dell'insieme $\bar{\mathcal{T}} \cup \mathcal{S}$.

Branching e oracolo di generazione colonne

Lo schema di branching è stato proposto originariamente da Ryan e Foster [3] e adattato da Desrochers e Soumis (come descritto in [2]).

Definiamo $T(u,v) \subseteq \mathcal{T}$ l'insieme dei turni in cui il servizio elementare r_u è eseguito immediatamente dopo r_v (vale a dire senza ulteriori servizi elementari in mezzo). Data x, soluzione del rilassamento lineare del generico (sotto-)problema P_ℓ, scegliamo una coppia

di servizi elementari r_u e r_v tali che

$$0 < \sum_{\ell \in T(u,v)} x_\ell < 1.$$

L'originale problema P_ℓ viene decomposto in due sottoproblemi $P_{\ell 0}$ in cui $\sum_{\ell \in T(u,v)} x_\ell = 0$ e $P_{\ell 1}$ in cui $\sum_{\ell \in T(u,v)} x_\ell \geq 1$. Nel problema $P_{\ell 0}$ *si proibisce* che r_u e r_v vengano eseguiti consecutivamente e, nella rete G, impiegata nell'oracolo di generazione colonne, questo corrisponde a rimuovere tutti gli archi in *avanti* (i, j) corrispondenti a sottoturni L_{ij} in cui r_u e r_v sono coperti. Nel problema $P_{\ell 1}$ *si impone* che r_u e r_v vengano eseguiti consecutivamente, vale a dire fare in modo che ogni sottoturno che preveda di coprire r_u debba necessariamente coprire anche r_v. Questo corrisponde, nella rete G, a "condensare in un unico servizio elementare r_u e r_v e, conseguentemente rappresentarlo su G utilizzando due soli nodi anziché quattro.

In definitiva, utilizzando la tecnica di branching su esposta, siamo in grado, attraverso opportune modifiche della rete G, di utilizzare, anche per la soluzione del rilassamento lineare dei sottoproblemi $P_{\ell 0}$ e $P_{\ell 1}$, l'oracolo di generazione colonne basato sul calcolo di un cammino minimo.

Pubblicazioni

[1] P. Nobili, Appunti sul problema di Set Covering, IASI CNR, Roma 5 dicembre 1996. Riportato parzialmente in `www.disp.uniroma2.it/users/pacifici/teaching/crewsched.pdf`

[2] Desrochers, M. and F. Soumis, A Column Generation Approach to the Urban Transit Scheduling Problem, *Transp. Sc.*, **23**, 1, 1–13, 1989.

[3] Ryan, D.M. and B.A. Foster, An Integer Programming Approach to Scheduling, in *Computer Scheduling of Public Transport Urban Passenger Vehicle and Crew Scheduling*, edito da A. Wren, Elsevier, Amsterdam, 1981.

23, 1, 1–13, 1989.

IL PROBLEMA DEL CREW SCHEDULING

Un esempio di applicazione della procedura di programmazione dinamica per la determinazione del cammino di costo minimo con vincoli sulle risorse.

Si supponga dato il grafo $G = (N, A)$ di figura, per il problema di generazione di un turno ammissibile di costo ridotto associato ottimo. Sugli archi sono riportati i costi associati.

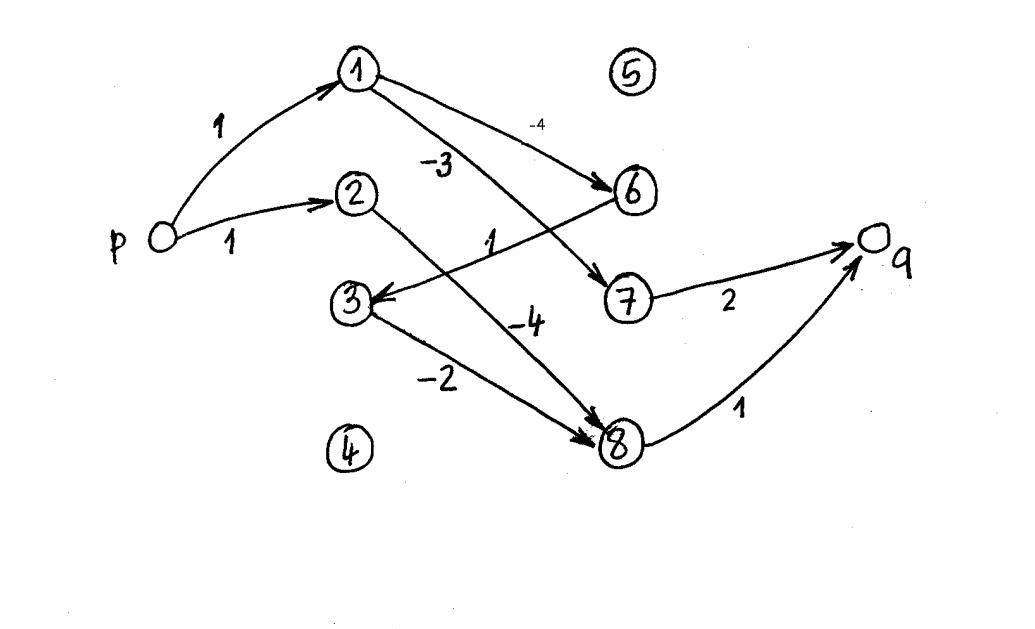

Figura 1.

In Tabella 1 sono riportati i costi relativi agli archi A, insieme con i dati relativi al consumo di risorse (numero di sottoturni, span, workload, rispettivamente.)

	Archi								
	p,1	p,2	1,6	1,7	2,8	3,8	6,3	7,q	8,q
costi	1	1	-4	-3	-4	-2	1	2	1
	iniziali		Sottoturni				pause	finali	
$d_{uv}^{(N)}$	0	0	1	1	1	1	0	0	0
$d_{uv}^{(S)}$	0	0	3	4	3	3	2	0	0
$d_{uv}^{(W)}$	0	0	3	4	3	3	0	0	0

Tabella 1.

Siano inoltre dati i valori massimi utilizzabili per le tre "risorse": $N_M = 2$, $S_M = 5$ e $W_M = 4$.

Procediamo dunque con il calcolo della funzione: $g_q(2,5,4)$. Come è noto (si veda la dispensa sul problema di turnazione del personale), la funzione viene calcolata in modo ricorsivo utilizzando la relazione di ricorrenza seguente:

$$g_v(x, y, z) = \min\{ g_v(x - d_{uv}^{(N)}, y - d_{uv}^{(S)}, z - d_{uv}^{(W)}) + l_{uv} \text{ con } uv \text{ arco di } A\}$$

e le condizioni iniziali seguenti:

$g_v(x, y, z) = +\infty$ se $\min\{ x, y, z \} < 0$ e per ogni v nodo di G.
$g_p(x, y, z) = 0$ se $\min\{ x, y, z \} \geq 0$

Per semplicità di scrittura poniamo $[v,x,y,z] := g_v(x, y, z)$. Si ha che:

1. $[q,2,5,4] = \min\{[7,2,5,4]+2; [8,2,5,4]+1\}$
2. Si procede con il calcolo di $[7,2,5,4] = [1,2 -1,5 -4,4 -4] -3 = [1,1,1,0] - 3$.
3. Procedendo *in profondità*, calcoliamo $[1,1,1,0] := [p,1,0,0] + 1 = 1$. Per cui si ha che (vedi passo 2) $[7,2,5,4] = -2$.
4. Continuiamo calcolando, $[8,2,5,4] = \min\{[2,2 -1,5 -3,4 -3] -4; [3, 2 -1,5 -3,4 -3] -2\} = \min\{[2,1,2,1] -4; [3, 1,2,1] -2\}$.
5. Siamo quindi costretti a calcolare ricorsivamente $[2,1,2,1]$ e $[3, 1,2,1]$. Si ha che $[2,1,2,1] = [p,1,2,1] +1 = 1$.
6. Si ha inoltre $[3,1,2,1] = [6,1 -0,2 -2,1 -0] +1 = [6,1,0,1] +1$; ma $[6,1,0,1] = [1,1 -1,0 -3,1 -3] -4 = +\infty$ per le condizioni iniziali.
7. Ma allora $[3, 1,2,1] = +\infty$ e quindi (vedi passo 4) $[8,2,5,4] = \min\{1 -4, +\infty\} = -3$.
8. Si ha quindi che (vedi passo 1) $[q,2,5,4] = \min\{-2 +2; -3 +1\} = -2$ che è il costo minimo di un cammino che non viola i vincoli sulle "risorse".

Come si vede anche in questo piccolo esempio, la soluzione di equazioni di ricorrenza del tipo in esame costringe a scrivere in (e a richiamare dalla) memoria un gran numero di dati (espressioni) parziali come, ad esempio, [8,2,5,4]. Sia la natura dei dati numerici del problema che le strategie con cui si sceglie di procedere al calcolo delle espressioni parziali possono determinare le prestazioni (in temini di velocità di calcolo) con cui si computa la soluzione ottima.

Per quanto riguarda il cammino di costo minimo, esso può essere ricostruito a ritroso a partire da q e tenendo conto dei minimi nelle varie relazioni ricorsive:

1. Da q, in base al minimo calcolato al passo 8, si ottiene che il suo predecessore è il nodo 8 (infatti il minimo è ottenuto in corrispondenza all'arco $(8,q)$.
2. Sulla base di analoghe considerazioni, si ha che il predecessore di 8 è (vedi passi 7 e 4) il nodo 2. Dal passo 5, si nota che il predecessore di 2 è necessariamente il nodo iniziale p.
3. In definitiva il cammino ammissibile di costo minimo è $(p,2,8,q)$ con costo pari a -2.

Si noti che il cammino $(p,1,6,3,8,q)$, di costo complessivo minore e pari a -3, non è ammissibile.