PROBLEMA DEL COMMESSO VIAGGIATORE:

METODI RISOLUTIVI e APPLICAZIONE

Descrizione del problema

Un rappresentante operante nel settore del legno si occupa di fornire utensili e materiali agli artigiani e a piccole industrie della zona. Gli indirizzi dei clienti sono stati usati per calcolare, attraverso un software di navigazione satellitare, le distanze effettive presenti tra essi.

La matrice dei costi (distanze) riportata non è simmetrica, questo dipende dal fatto che non sempre sono consentiti entrambi i sensi di marcia nelle strade che collegano due località.

	Ancona	Potenza Picena	Osimo Stazione	Civitanova Marche	Castelfidardo	Camerano
Ancona	-	33,6	14,0	40,9	14,5	11,5
Potenza Picena	34,7	-	21,7	13,0	20,2	23,4
Osimo Stazione	14,8	21,5	-	29,3	2,0	3,9
Civitanova Marche	41,7	13,1	29,4	-	27,6	30,3
Castelfidardo	15,0	20,2	2,0	27,5	-	3,9
Camerano	12,0	22,8	2,0	30,1	4,0	-

Nella tabella sono riportati i Km necessari per raggiungere una destinazione (in alto nella tabella) a partire da un'origine (a sinistra).

Si vuole minimizzare la distanza (o il tempo) totale che occorre per effettuare un giro di visita a tutti i clienti che si intendono visitare, senza passare due volte per la stessa località, e ritornando nella località di partenza.

Nota: è possibile costruire un *grafo* associato al problema in cui ogni nodo rappresenta una località e ogni arco rappresenta la distanza (coefficienti c_{ij}) che le separa. Il grafo relativo al problema è *orientato*, questo deriva dal fatto che, avendo una tabella dei costi non simmetrica, la distanza tra due *nodi* qualsiasi *i*, *j* è diversa della distanza tra *j* ed *i*. Le soluzioni ammissibili del problema sono quindi i possibili cicli hamiltoniani orientati sul grafo stesso. Infatti un *ciclo* corrisponde ad un cammino dove i nodi di inizio e di fine coincidono, in particolare i *cicli hamiltoniani* sono quei cicli che passano una ed una sola volta per ogni nodo. Lo scopo sarà quindi quello di determinare il ciclo hamiltoniano al quale corrisponde il costo minimo.

La formulazione matematica del problema del commesso viaggiatore è quindi la seguente:

$$1) \quad \min z = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} \cdot x_{ij}$$

$$2) \quad \sum_{i=1}^{n} x_{ij} = 1 \qquad\qquad j = 1, \dots, n$$

$$3) \quad \sum_{j=1}^{n} x_{ij} = 1 \qquad\qquad i = 1, \dots, n$$

$$4) \quad \sum_{i,j \in S} x_{ij} \leq |S| - 1 \qquad\qquad S \subset N; 2 \leq |S| \leq n - 2$$

$$5) \quad x_{ij} = \begin{cases} 1 \text{ se l'arco } (i, j) \text{ è scelto} \\ 0 \text{ se l'arco } (i, j) \text{ non è scelto} \end{cases}$$

in cui S è un qualsiasi sottoinsieme di nodi di cardinalità compresa tra *2* e *N-2*. l vincoli 2 significano che una destinazione può essere raggiunta da una sola origine mentre i vincoli 3 servono ad evitare che da una stessa origine si parta per raggiungere più destinazioni. I vincoli 4 sono definiti *vincoli di eliminazione dei sottocicli*, questi, insieme ai vincoli 2 e 3 eliminano tutti i cicli con lunghezza minore di *n*. Si noti che il numero di vincoli di tipo 4 è *esponenziale*.

APPROCCI RISOLUTIVI

Il numero esponenziale di vincoli di tipo 4 rende il problema difficilmente risolvibile con metodi diretti. Si propongono i seguenti approcci risolutivi alternativi:

1. Branch & Bound
2. Generazione di vincoli (constraint generation).

Branch & Bound

Qualora non si considerino i vincoli 4 il problema si trasforma in un *problema di assegnamento* e può essere facilmente risolto con il metodo del simplesso (con garanzia di soluzioni intere anche senza imporre i vincoli 5) oppure con il *metodo ungherese* (versione specializzata del metodo del simplesso, in modo analogo al metodo di Dantzig per il problema dei trasporti).

Risolvendo il *problema di assegnamento* associato, però, di solito compaiono nella soluzione dei sottocicli, il che fa sì che la soluzione ottenuta sia non ammissibile per il problema del commesso viaggiatore. Tuttavia, tale soluzione rappresenta un limite inferiore (lower bound) per il problema in esame (si tratta del problema di partenza in cui sono stati eliminati dei vincoli) e si può applicare lo schema del *"Branch & Bound"*: tra tutti i problemi rimasti da valutare si sceglie quello che ha il più piccolo valore del limite inferiore del problema di assegnamento associato (politica di esplorazione *best bound first*). Se il valore della soluzione del problema scelto è maggiore od uguale di quello della miglior soluzione nota, la ricerca termina perché la soluzioni di tutti i problemi rimasti non migliorerebbero il suo valore.

Se invece la soluzione del problema scelto costituisce un ciclo hamiltoniano, il problema viene eliminato (nodo chiuso per ammissibilità del rilassamento), la migliore soluzione nota viene aggiornata e si considera il problema successivo.

Quando le verifiche precedenti hanno esito negativo, si esegue un’operazione di *branch*: il problema scelto viene rimpiazzato da due o più problemi ottenuti imponendo il soddisfacimento di uno dei vincoli 4 violati e si passa alla scelta del problema seguente. Come si vedrà nell’esempio che segue, vi sono vari modi di eseguire il branching. In ogni caso, si cercherà di preservare la struttura di assegnamento associata al problema da risolvere ad ogni nodo, per sfruttare le caratteristiche di totale unimodularità della relativa matrice dei vincoli.

Generazione di vincoli

Siccome i vincoli 4 sono molto numerosi, in particolar modo quando si ha a che fare con un numero elevato di località da visitare, la loro applicazione in fase di risoluzione non è conveniente. Si procede quindi come segue:

1. Si considera come problema corrente il problema di assegnamento ottenuto eliminando tutti i vincoli di tipo 4
2. Si risolve il problema corrente
3. Se la soluzione non presenta sotto-cicli, la soluzione corrente è la soluzione ottima
4. Se ci sono dei sottocicli, aggiungere al problema corrente i vincoli di eliminazione di uno (o più) dei sottocicli stessi
5. tornare al passo 2

Si noti come i vincoli di eliminazione di sottocicli si introducano in modo incrementale. In questo modo una soluzione ammissibile potrà essere trovata prima di aggiungere tutti i vincoli di tipo 4 (generalmente se ne introducono solo una piccola percentuale). Tale soluzione sarà anche ottima, visto che, oltre ad essere ammissibile, risolve un problema rilassato (con meno vincoli) rappresenta una stima ottimistica del valore ottimo della funzione obiettivo.

Applicazione del Branch & Bound

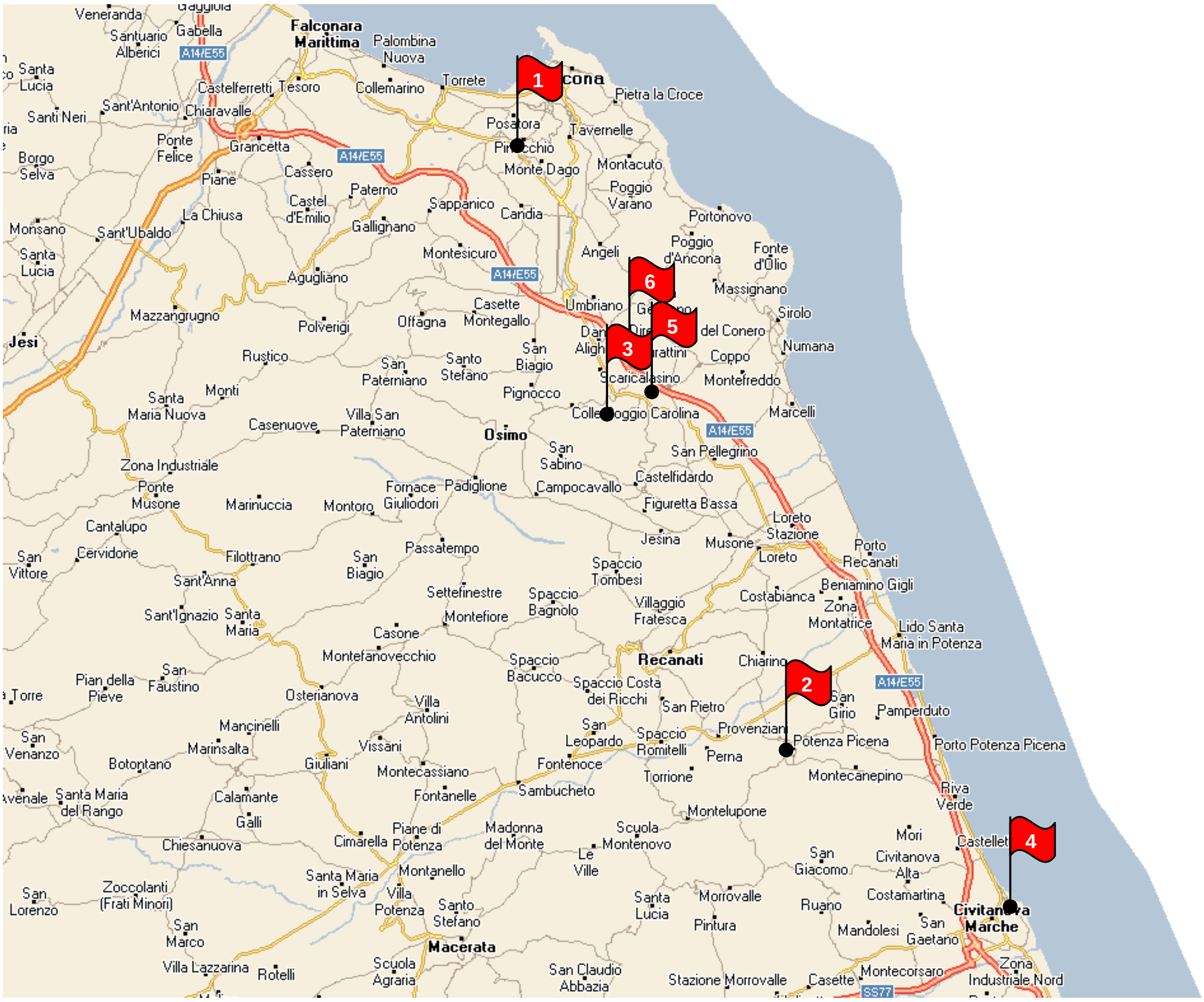
Nell'esercizio che stiamo per affrontare si impiegherà il software LINDO *(Linear Interactive aNd Discrete Optimizer)* per determinare le soluzioni del *problema di assegnamento* associato (P0) e di quelli successivi che si genereranno per stringere la regione di ammissibilità. La tabella dei costi (Km) che prendiamo in considerazione è la seguente:

	Ancona	Potenza Picena	Osimo Stazione	Civitanova Marche	Castelfidardo	Camerano
Ancona	-	33,6	14,0	40,9	14,5	11,5
Potenza Picena	34,7	-	21,7	13,0	20,2	23,4
Osimo Stazione	14,8	21,5	-	29,3	2,0	3,9
Civitanova Marche	41,7	13,1	29,4	-	27,6	30,3
Castelfidardo	15,0	20,2	2,0	27,5	-	3,9
Camerano	12,0	22,8	2,0	30,1	4,0	-

Per facilitare la lettura delle soluzioni associamo un numero ad ogni località:

Località	***Numero***
Ancona	1
Potenza Picena	2
Osimo Stazione	3
Civitanova Marche	4
Castelfidardo	5
Camerano	6

Obiettivo: Trovare il giro di visite che passa una e una sola volta per ognuna delle sei località e che minimizza la distanza totale percorsa.


Veneranda
Santuario Alberici
Gabella
A14/E55
Falconara Marittima
Palombina Nuova
Santa Lucia
Castelferretti
Tesoro
Collemarino
Torrete
1
Pietra la Croce
Sant'Antonio
Chiaravalle
Santi Neri
Posatora
Tavernelle
Ponte Felice
Grancetta
Borgo Selva
Piane
Cassero
Montacuto
Monte Dago
Paterno
Poggio Varano
Castel d'Emilio
Sappanico
Candia
La Chiusa
Portonovo
Monsano
Sant'Ubaldo
Gallignano
Santa Lucia
Montesicuro
Angeli
Poggio d'Ancona
Fonte d'Olio
Agugliano
Massignano
6
Casette Montegallo
Umbriano
Sirolo
Mazzangrugno
5
Polverigi
Offagna
del Conero
Jesi
3
Numana
Rustico
San Biagio
San Paterniano
Santo Stefano
Coppo
Montefreddo
Monti
Pignocco
Santa Maria Nuova
Poggio
Carolina
Marcelli
Villa San Paterniano
Casenuove
Osimo
San Sabino
San Pellegrino
Zona Industriale
Castelfidardo
Ponte Musone
Fornace Giuliodori
Padiglione
Campocavallo
Marinuccia
Montoro
Figuretta Bassa
Cantalupo
Loreto Stazione
Jesina
Musone
Cervidone
Passatempo
Loreto
Porto Recanati
San Vittore
Filottrano
San Biagio
Spaccio Tombesi
Sant'Anna
Settefinestre
Spaccio Bagnolo
Beniamino Gigli
Costabianca
Zona Montatrice
Sant'Ignazio
Santa Maria
Montefiore
Villaggio Fratesca
Casone
Lido Santa Maria in Potenza
Montefanovecchio
Spaccio Bacucco
Recanati
Chiarino
San Faustino
Pian della Pieve
Spaccio Costa dei Ricchi
2
A14/E55
Osterianova
Villa Antolini
San Pietro
San Girio
Pamperduto
San Venanzo
Mancinelli
San Leopardo
Spaccio Romitelli
Provenziani
Potenza Picena
Marinsalta
Vissani
Porto Potenza Picena
Botontano
Giuliani
Montecassiano
Fontenoce
Perna
Torrione
Montecanepino
Santa Maria del Rango
Calamante
Fontanelle
Sambucheto
Riva Verde
Galli
Montelupone
Piane di Potenza
Madonna del Monte
Scuola Montenovo
Mori
Castellet
4
Chiesanuova
Cimarella
Le Ville
San Giacomo
Civitanova Alta
Santa Maria in Selva
Montanello
Costamartina
San Lorenzo
Zoccolanti (Frati Minori)
Villa Potenza
Santo Stefano
Santa Lucia
Morrovalle
Ruano
Civitanova Marche
San Marco
Macerata
Pintura
Mandolesi
San Gaetano
Villa Lazzarina
Rotelli
Scuola Agraria
San Claudio Abbazia
Stazione Morrovalle
Casette
Montecorsaro
Zona Industriale Nord
SS77

Problema P0

Il problema P0 scritto nel linguaggio utilizzato dal LINDO è il seguente:

```
min 33.6 x12 + 14.0 x13 + 40.9 x14 + 14.5 x15 + 11.5 x16
    34.7 x21 + 21.7 x23 + 13.0 x24 + 23.5 x25 + 23.4 x26
    14.8 x31 + 21.5 x32 + 29.3 x34 +  2.0 x35 +  3.9 x36
    41.7 x41 + 13.1 x42 + 29.4 x43 + 27.6 x45 + 30.3 x46
    15.0 x51 + 20.2 x52 +  2.0 x53 + 27.5 x54 +  3.9 x56
    12.0 x61 + 22.8 x62 +  2.0 x63 + 30.1 x64 +  4.0 x65

subject to
      riga1)  x12 + x13 + x14 + x15 + x16=1
      riga2)  x21 + x23 + x24 + x25 + x26=1
      riga3)  x31 + x32 + x34 + x35 + x36=1
      riga4)  x41 + x42 + x43 + x45 + x46=1
      riga5)  x51 + x52 + x53 + x54 + x56=1
      riga6)  x61 + x62 + x63 + x64 + x65=1
      col1)   x21 + x31 + x41 + x51 + x61=1
      col2)   x12 + x32 + x42 + x52 + x62=1
      col3)   x13 + x23 + x43 + x53 + x63=1
      col4)   x14 + x24 + x34 + x54 + x64=1
      col5)   x15 + x25 + x35 + x45 + x65=1
      col6)   x16 + x26 + x36 + x46 + x56=1
end
```

L'output che si ottiene in seguito al comando *Solve* è il seguente:

```
 LP OPTIMUM FOUND AT STEP      9

        OBJECTIVE FUNCTION VALUE

        1)      53.60000

  VARIABLE        VALUE          REDUCED COST
       X12         0.000000         20.499998
       X13         0.000000          4.400000
       X14         0.000000         27.900002
       X15         0.000000          2.900000
       X16         1.000000          0.000000
       X21         0.000000         15.100000
       X23         0.000000         12.100000
       X24         1.000000          0.000000
       X25         0.000000         11.900000
       X26         0.000000         11.900000
       X31         0.000000          4.800000
       X32         0.000000         18.000000
       X34         0.000000         25.900000
       X35         1.000000          0.000000
       X36         0.000000          2.000000
       X41         0.000000         22.100000
       X42         1.000000          0.000000
       X43         0.000000         19.799999
       X45         0.000000         16.000000
       X46         0.000000         18.799999
       X51         0.000000          3.000000
       X52         0.000000         14.700001
       X53         1.000000          0.000000
       X54         0.000000         22.100000
       X56         0.000000          0.000000
       X61         1.000000          0.000000
       X62         0.000000         17.299999
       X63         0.000000          0.000000
       X64         0.000000         24.700001
       X65         0.000000          0.000000

       ROW   SLACK OR SURPLUS     DUAL PRICES
    RIGA1)         0.000000          0.000000
    RIGA2)         0.000000          0.000000
    RIGA3)         0.000000          9.600000
    RIGA4)         0.000000          0.000000
```

```
      RIGA5)        0.000000          7.600000
      RIGA6)        0.000000          7.600000
       COL1)        0.000000        -19.600000
       COL2)        0.000000        -13.100000
       COL3)        0.000000         -9.600000
       COL4)        0.000000        -13.000000
       COL5)        0.000000        -11.600000
       COL6)        0.000000        -11.500000

 NO. ITERATIONS=       9
```

La soluzione del problema P0 trovata vale *53,6* ed è la seguente:

1-6-1, 2-4-2, 3-5-3

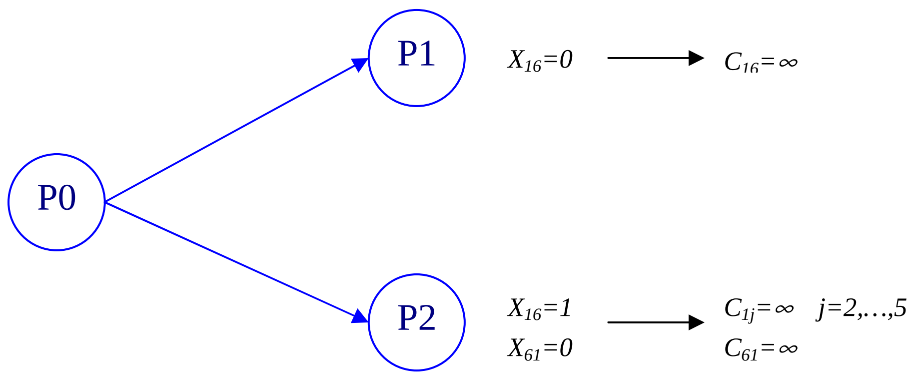
Tale soluzione tuttavia non è ammissibile in quanto i cicli non sono hamiltoniani.
Si procederà in modo che uno dei tre sottocicli non faccia parte della soluzione. In altre parole, attraverso la modifica dei coefficienti c_{ij}, realizzeremo il vincolo:

$$x_{16} + x_{61} \le 1$$

Ciò significa scegliere una delle variabili nel ciclo (ad esempio x_{16}) e porre $x_{16}=0$, tale caso infatti è compatibile con il vincolo e permette che si verifichi anche il caso <; oppure porre $x_{16}=1$ e allo stesso tempo $x_{61}=0$ in tal caso il vincolo è rispettato con il segno di uguaglianza.
Attenzione: l'aggiunta del vincolo direttamente nel modello comporterebbe la perdita delle proprietà della matrice dei vincoli per i problemi di assegnamento (totale unimodularità) e pertanto non si potrebbe utilizzare il semplice simplesso per la soluzione del nuovo problema generato: occorrerebbe mantenere il vincolo sull'interezza delle variabili decisionali!
Per evitare ciò, si può formulare il problema figlio sempre come problema di assegnamento osservando quanto segue. Imponendo ad alcuni coefficienti c_{ij} della tabella dei costi valori molto alti (nel nostro caso 1000 è sufficiente) è come se si vincolasse a non scegliere alcuni percorsi e a preferire quello il cui coefficiente c_{ij} è rimasto invariato. Per esempio se si pone $c_{hk}=\infty$ è come se si inserisse il vincolo $x_{hk}=0$. Inoltre, per imporre $x_{hk}=1$, basta mettere a ∞ tutti i costi c_{hk}

Questa tecnica permette di risolvere sempre *problemi di assegnamento* consente quindi l'utilizzo del metodo del simplesso (*ungherese*).

Ciò può essere rappresentato nella seguente maniera:



Si noti che, imponendo x_{16} a 0, il vincolo di eliminazione dei sottocicli in esame è automaticamente rispettato. Quando invece si impone x_{16} a 1, bisogna imporre almeno una alla volta le altre variabili a 0. Se k è il numero di archi nel sottociclo in esame, per non perdere soluzioni ammissibili, si introducono (oltre al nodo con $x_{16} = 0$), altri $k - 1$ nodi con vincoli $x_{16} = 1$, $x_{pq} = 0$, dove (p,q) è uno degli altri $k - 1$ archi. Si noti che gli insiemi di soluzioni rappresentati da ciascuno dei k figli non è disgiunto. Ciò, come sappiamo, non inficia la validità del branch & bound, che impone solo di non perdere nessuna soluzione ammissibile passando dal nodo padre all'unione dei nodi figli.

Un modo alternativo per effettuare il branching, ottenendo sempre k nodi figli, è quello di imporre solo il vincolo sulla variabile da mettere a 0, come segue:

1. si sceglie un sottociclo
2. si genera un figlio per ogni arco nel sottociclo imponendo come vincolo di branching che la corrispondente variabile sia 0.

Si noti come, anche in questo caso, il principio di intersezione vuota tra tutti i sottoproblemi generati non sia rispettato, ma ciò, come sappiamo, non inficia la validità del branch & bound, che impone solo di non perdere nessuna soluzione ammissibile passando dal nodo padre all'unione dei nodi figli.

Ponendo a infinito il costo della variabile da imporre a 0, di volta in volta, i sottoproblemi sono valutabili come problemi di assegnamento, secondo lo stesso metodo visto finora.

Si può notare come il livello si sovrapposizione degli insiemi di soluzioni associate ad ogni nodo è più elevato in questo secondo caso, rispetto al primo. Sono possibili tecniche che riducano ulteriormente il livello di sovrapposizione, ma ciò comporterebbe un numero molto elevato di nodi da generare ad ogni branching.

Nel prosieguo si considererà la prima tecnica di branching, considerata un buon compromesso tra i livelli di sovrapposizione e il numero di nodi di branching.

Problema P1 (ottenuto da P0)

Come appena visto dal diagramma ad albero porre $x_{16}=0$ è equivalente a porre il coefficiente $c_{16}=\infty$. La tabella dei costi del *problema di assegnamento* P1è dunque la seguente:

	Ancona	Potenza Picena	Osimo Stazione	Civitanova Marche	Castelfidardo	Camerano
Ancona	-	33,6	14,0	40,9	14,5	∞
Potenza Picena	34,7	-	21,7	13,0	20,2	23,4
Osimo Stazione	14,8	21,5	-	29,3	2,0	3,9
Civitanova Marche	41,7	13,1	29,4	-	27,6	30,3
Castelfidardo	15,0	20,2	2,0	27,5	-	3,9
Camerano	12,0	22,8	2,0	30,1	4,0	-

Con il LINDO il problema può essere implementato come segue:

```
min   33.6 x12 + 14.0 x13 + 40.9 x14 + 14.5 x15 + 1000 x16
      34.7 x21 + 21.7 x23 + 13.0 x24 + 23.5 x25 + 23.4 x26
      14.8 x31 + 21.5 x32 + 29.3 x34 +  2.0 x35 +  3.9 x36
      41.7 x41 + 13.1 x42 + 29.4 x43 + 27.6 x45 + 30.3 x46
      15.0 x51 + 20.2 x52 +  2.0 x53 + 27.5 x54 +  3.9 x56
      12.0 x61 + 22.8 x62 +  2.0 x63 + 30.1 x64 +  4.0 x65

subject to
      riga1) x12 + x13 + x14 + x15 + x16=1
      riga2) x21 + x23 + x24 + x25 + x26=1
      riga3) x31 + x32 + x34 + x35 + x36=1
      riga4) x41 + x42 + x43 + x45 + x46=1
      riga5) x51 + x52 + x53 + x54 + x56=1
      riga6) x61 + x62 + x63 + x64 + x65=1
      col1)  x21 + x31 + x41 + x51 + x61=1
      col2)  x12 + x32 + x42 + x52 + x62=1
      col3)  x13 + x23 + x43 + x53 + x63=1
      col4)  x14 + x24 + x34 + x54 + x64=1
      col5)  x15 + x25 + x35 + x45 + x65=1
      col6)  x16 + x26 + x36 + x46 + x56=1
end
```

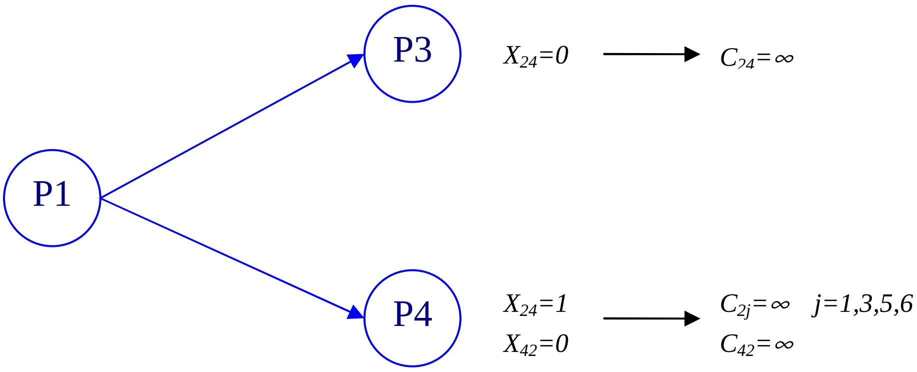
La soluzione trovata in questo caso è, ovviamente, maggiore in quanto si è aumentato il numero dei vincoli e vale *58*

```
LP OPTIMUM FOUND AT STEP     11

        OBJECTIVE FUNCTION VALUE

        1)      58.00000

  VARIABLE        VALUE          REDUCED COST
       X12         0.000000         20.499998
       X13         1.000000          0.000000
       X14         0.000000         27.900002
       X15         0.000000          0.500000
       X16         0.000000       9984.099609
       X21         0.000000         10.700001
       X23         0.000000          7.700001
       X24         1.000000          0.000000
       X25         0.000000          9.500000
       X26         0.000000          7.500000
       X31         0.000000          2.800000
       X32         0.000000         20.400000
       X34         0.000000         28.299999
       X35         1.000000          0.000000
       X36         0.000000          0.000000
       X41         0.000000         17.700001
       X42         1.000000          0.000000
       X43         0.000000         15.400000
       X45         0.000000         13.600000
       X46         0.000000         14.400000
       X51         0.000000          3.000000
       X52         0.000000         19.100000
       X53         0.000000          0.000000
       X54         0.000000         26.500000
       X56         1.000000          0.000000
       X61         1.000000          0.000000
       X62         0.000000         21.699999
       X63         0.000000          0.000000
       X64         0.000000         29.100000
       X65         0.000000          2.000000
```

I nuovi sottocicli sono:

1-3-5-6-1, 2-4-2

Ancora una volta la soluzione non è ammissibile perché non formata da un unico ciclo hamiltoniano. Si sceglie, arbitrariamente, il sottociclo *2-4-2* (perché composto fa un numero inferiore di archi, quindi genererà meno nodi figli. Am al scelta è arbitraria!) Verranno quindi generati ulteriori due sottoproblemi P3 e P4.



Problema P2 (ottenuto da P0)

Come indicato dal diagramma ad albero porre $x_{16}=1$ e $x_{61}=0$ corrisponde a porre a porre il coefficiente $c_{1j}=\infty$ con $j=2,\ldots,5$ e porre $c_{61}=\infty$. La tabella dei costi del *problema di assegnamento* P1è dunque la seguente:

	Ancona	Potenza Picena	Osimo Stazione	Civitanova Marche	Castelfidardo	Camerano
Ancona	-	∞	∞	∞	∞	11,5
Potenza Picena	34,7	-	21,7	13,0	20,2	23,4
Osimo Stazione	14,8	21,5	-	29,3	2,0	3,9
Civitanova Marche	41,7	13,1	29,4	-	27,6	30,3
Castelfidardo	15,0	20,2	2,0	27,5	-	3,9
Camerano	∞	22,8	2,0	30,1	4,0	-

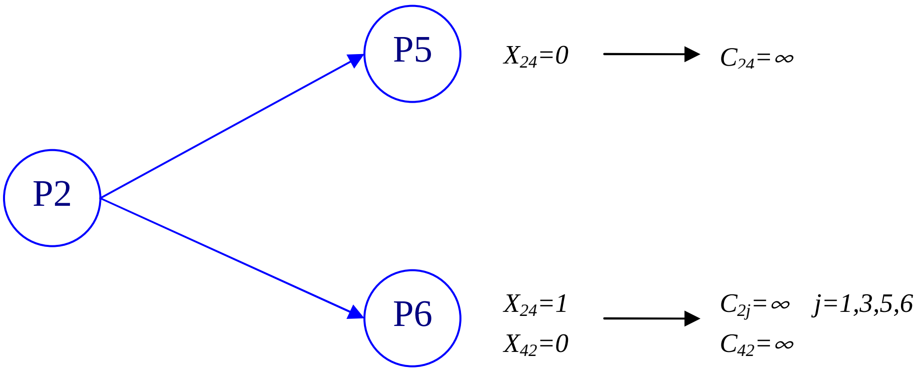
Ovvero:

```
min   1000 x12 + 1000 x13 + 1000 x14 + 1000 x15 + 11.5 x16
      34.7 x21 + 21.7 x23 + 13.0 x24 + 23.5 x25 + 23.4 x26
      14.8 x31 + 21.5 x32 + 29.3 x34 +  2.0 x35 +  3.9 x36
      41.7 x41 + 13.1 x42 + 29.4 x43 + 27.6 x45 + 30.3 x46
      15.0 x51 + 20.2 x52 +  2.0 x53 + 27.5 x54 +  3.9 x56
      1000 x61 + 22.8 x62 +  2.0 x63 + 30.1 x64 +  4.0 x65
```

La soluzione ottenuta in questo caso vale *56,6* ed i sottocicli trovati sono:

$$1\text{-}6\text{-}3\text{-}5\text{-}1,\ 2\text{-}4\text{-}2$$

Essendo il valore della soluzione inferiore a quella del problema P1 esploreremo prima i nodi figlio di P2 e poi quelli di P1.



Problema P5 (ottenuto da P2)

La tabella dei costi alla luce dei nuovi vincoli sarà:

	Ancona	Potenza Picena	Osimo Stazione	Civitanova Marche	Castelfidardo	Camerano
Ancona	-	∞	∞	∞	∞	11,5
Potenza Picena	34,7	-	21,7	∞	20,2	23,4
Osimo Stazione	14,8	21,5	-	29,3	2,0	3,9
Civitanova Marche	41,7	13,1	29,4	-	27,6	30,3
Castelfidardo	15,0	20,2	2,0	27,5	-	3,9
Camerano	∞	22,8	2,0	30,1	4,0	-

Che in LINDO diventa:

```
min   1000 x12 + 1000 x13 + 1000 x14 + 1000 x15 + 11.5 x16
      34.7 x21 + 21.7 x23 + 1000 x24 + 23.5 x25 + 23.4 x26
      14.8 x31 + 21.5 x32 + 29.3 x34 +  2.0 x35 +  3.9 x36
      41.7 x41 + 13.1 x42 + 29.4 x43 + 27.6 x45 + 30.3 x46
      15.0 x51 + 20.2 x52 +  2.0 x53 + 27.5 x54 +  3.9 x56
      1000 x61 + 22.8 x62 +  2.0 x63 + 30.1 x64 +  4.0 x65
```

La funzione obiettivo assume il valore *90,8* ed è ammissibile in quanto corrisponde al ciclo hamiltoniano:

1-6-3-5-4-2-1

Il problema P2, avendo soluzione ammissibile, sarà una foglia dell'albero dei sottoproblemi e non avrà ulteriori nodi figlio.

Problema P6 (ottenuto da P2)

La tabella dei costi è:

	Ancona	Potenza Picena	Osimo Stazione	Civitanova Marche	Castelfidardo	Camerano
Ancona	-	∞	∞	∞	∞	11,5
Potenza Picena	∞	-	∞	13,0	∞	∞
Osimo Stazione	14,8	∞	-	29,3	2,0	3,9
Civitanova Marche	41,7	13,1	29,4	-	27,6	30,3
Castelfidardo	15,0	20,2	2,0	27,5	-	3,9
Camerano	∞	22,8	2,0	30,1	4,0	-

che in LINDO corrisponde a:

```
min   1000 x12 + 1000 x13 + 1000 x14 + 1000 x15 + 11.5 x16
      1000 x21 + 1000 x23 + 13.0 x24 + 1000 x25 + 1000 x26
      14.8 x31 + 1000 x32 + 29.3 x34 +  2.0 x35 +  3.9 x36
      41.7 x41 + 1000 x42 + 29.4 x43 + 27.6 x45 + 30.3 x46
      15.0 x51 + 20.2 x52 +  2.0 x53 + 27.5 x54 +  3.9 x56
      1000 x61 + 22.8 x62 +  2.0 x63 + 30.1 x64 +  4.0 x65
```

Risolvendo il problema si ottiene un valore della funzione obiettivo pari a *89,1*. Tale soluzione tuttavia generando i seguenti sottocicli non risulta ammissibile:

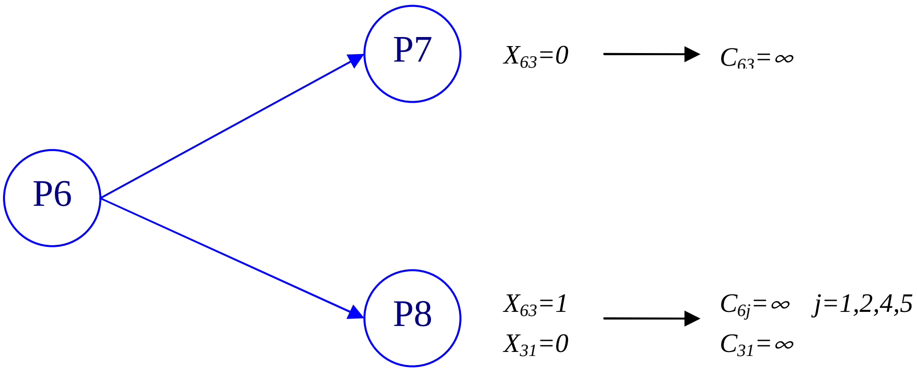
1-6-3-1, 2-4-5-2

Alla luce del fatto che P6 ha come valore della funzione obiettivo un valore minore di *90,8* (valore della più piccola soluzione ammissibile trovata), dovremo considerare i sottoproblemi di P6 per vedere se danno origine a soluzioni ammissibili migliori di quella appena trovata.

Questa volta ci troviamo di fronte ad un caso differente da quelli precedenti, infatti dobbiamo porre dei vincoli in modo da eliminare un ciclo che attraversa 3 nodi del grafo. Il vincolo 4 che dovremo applicare sarà quindi il seguente:

$$x_{16} + x_{63} + x_{31} \leq 2 \qquad (1)$$

Considerando che la variabile x_{16}=1 in seguito all'assegnazione effettuata in P2 (dal quale si è ottenuto P6) i casi possibili sono i seguenti:



Il problema P7 contempla anche il caso di < nel vincolo (1). Si noti che, se il nodo P6 non avesse ereditato il vincolo x_{16}=1, avremmo dovuto generare anche un terzo nodo figlio x_{63}=1, x_{16}=0 (nel caso in esame, invece, tale nodo è subito riconoscibile come non ammissibile, vista la contraddizione tra i vincoli di branching, e non viene generato, anziché essere generato e venire immediatamente chiuso).

Prima di analizzare i sottoproblemi di P6 procederemo con l'analisi dei sottoproblemi di P1 in quanto il valore ottimo della sua funzione obiettivo era inferiore.

Problema P3 (ottenuto da P1)

La tabella dei costi derivante dai nuovi vincoli è:

	Ancona	Potenza Picena	Osimo Stazione	Civitanova Marche	Castelfidardo	Camerano
Ancona	-	33,6	14,0	40,9	14,5	∞
Potenza Picena	34,7	-	21,7	∞	20,2	23,4
Osimo Stazione	14,8	21,5	-	29,3	2,0	3,9
Civitanova Marche	41,7	13,1	29,4	-	27,6	30,3
Castelfidardo	15,0	20,2	2,0	27,5	-	3,9
Camerano	12,0	22,8	2,0	30,1	4,0	-

Che implementato al LINDO diventa:

```
min   33.6 x12 + 14.0 x13 + 40.9 x14 + 14.5 x15 + 1000 x16
      34.7 x21 + 21.7 x23 + 1000 x24 + 23.5 x25 + 23.4 x26
      14.8 x31 + 21.5 x32 + 29.3 x34 +  2.0 x35 +  3.9 x36
      41.7 x41 + 13.1 x42 + 29.4 x43 + 27.6 x45 + 30.3 x46
      15.0 x51 + 20.2 x52 +  2.0 x53 + 27.5 x54 +  3.9 x56
      12.0 x61 + 22.8 x62 +  2.0 x63 + 30.1 x64 +  4.0 x65
```

Il valore assunto dalla funzione obiettivo in questo caso è *92,0* che risulta ammissibile dal momento che si ottiene il seguente ciclo hamiltoniano:

1-3-5-4-2-6-1

Tale soluzione è maggiore di quella ottenuta in P5 pertanto continueremo a considerare come valore candidato all'ottimo *90,8*. Anche in questo caso, come in P5, non ci saranno nodi figli di questo sottoproblema.

Problema P4 (ottenuto da P1)

La tabella dei costi è la seguente:

	Ancona	Potenza Picena	Osimo Stazione	Civitanova Marche	Castelfidardo	Camerano
Ancona	-	33,6	14,0	40,9	14,5	∞
Potenza Picena	∞	-	∞	13,0	∞	∞
Osimo Stazione	14,8	21,5	-	29,3	2,0	3,9
Civitanova Marche	41,7	∞	29,4	-	27,6	30,3
Castelfidardo	15,0	20,2	2,0	27,5	-	3,9
Camerano	12,0	22,8	2,0	30,1	4,0	-

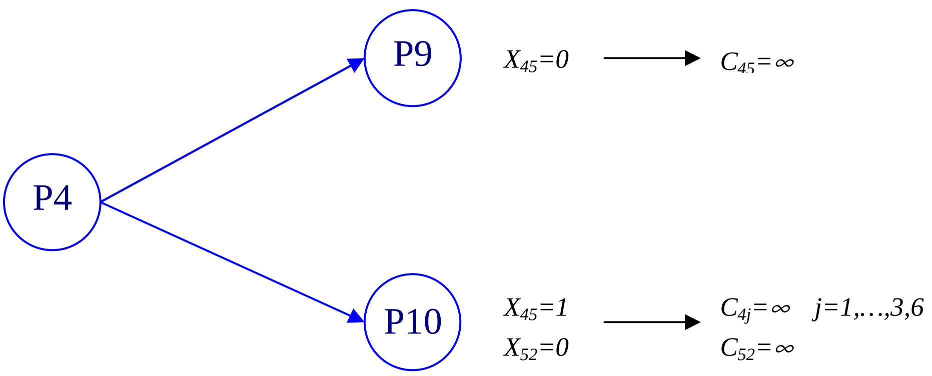
ovvero:

```
min   33.6 x12 + 14.0 x13 + 40.9 x14 + 14.5 x15 + 1000 x16
      1000 x21 + 1000 x23 + 13.0 x24 + 1000 x25 + 1000 x26
      14.8 x31 + 21.5 x32 + 29.3 x34 +  2.0 x35 +  3.9 x36
      41.7 x41 + 1000 x42 + 29.4 x43 + 27.6 x45 + 30.3 x46
      15.0 x51 + 20.2 x52 +  2.0 x53 + 27.5 x54 +  3.9 x56
      12.0 x61 + 22.8 x62 +  2.0 x63 + 30.1 x64 +  4.0 x65
```

In questo caso si ottiene *90,7* e i seguenti cicli:

1-3-6-1, 2-4-5-2

Essendo il valore della funzione obiettivo inferiore al minimo valore delle soluzioni ammissibili trovane fino ad ora, sarà necessario, analizzare i seguenti sottoproblemi:



Prima di procedere con l'analisi di questi sottoproblemi analizzeremo quelli ottenuti da P6 in quanto il valore della sua funzione obiettivo è minore.

Problema P7 (ottenuto da P6)

	Ancona	Potenza Picena	Osimo Stazione	Civitanova Marche	Castelfidardo	Camerano
Ancona	-	∞	∞	∞	∞	11,5
Potenza Picena	∞	-	∞	13,0	∞	∞
Osimo Stazione	14,8	∞	-	29,3	2,0	3,9
Civitanova Marche	41,7	13,1	29,4	-	27,6	30,3
Castelfidardo	15,0	20,2	2,0	27,5	-	3,9
Camerano	∞	22,8	∞	30,1	4,0	-

Che in LINDO equivale a:

```
min   1000 x12 + 1000 x13 + 1000 x14 + 1000 x15 + 11.5 x16
      1000 x21 + 1000 x23 + 13.0 x24 + 1000 x25 + 1000 x26
      14.8 x31 + 21.5 x32 + 29.3 x34 +  2.0 x35 +  3.9 x36
      41.7 x41 + 1000 x42 + 29.4 x43 + 27.6 x45 + 30.3 x46
      15.0 x51 + 20.2 x52 +  2.0 x53 + 27.5 x54 +  3.9 x56
      1000 x61 + 22.8 x62 + 1000 x63 + 30.1 x64 +  4.0 x65
```

La soluzione ottima per questi coefficienti vale *91,7* e corrisponde al ciclo hamiltoniano:

1-6-2-4-5-3-1

Continueremo comunque a tenere come valore ottimo di riferimento il *90,8* ottenuto risolvendo P6.

Problema P8 (ottenuto da P6)

	Ancona	Potenza Picena	Osimo Stazione	Civitanova Marche	Castelfidardo	Camerano
Ancona	-	∞	∞	∞	∞	11,5
Potenza Picena	∞	-	∞	13,0	∞	∞
Osimo Stazione	∞	∞	-	29,3	2,0	3,9
Civitanova Marche	41,7	13,1	29,4	-	27,6	30,3
Castelfidardo	15,0	20,2	2,0	27,5	-	3,9
Camerano	∞	∞	2,0	∞	∞	-

In LINDO si ottiene:

```
min   1000 x12 + 1000 x13 + 1000 x14 + 1000 x15 + 11.5 x16
      1000 x21 + 1000 x23 + 13.0 x24 + 1000 x25 + 1000 x26
      1000 x31 + 21.5 x32 + 29.3 x34 +  2.0 x35 +  3.9 x36
      41.7 x41 + 1000 x42 + 29.4 x43 + 27.6 x45 + 30.3 x46
```

```
15.0 x51 + 20.2 x52 +  2.0 x53 + 27.5 x54 +  3.9 x56
1000 x61 + 1000 x62 +  2.0 x63 + 1000 x64 + 1000 x65
```

In questo caso otteniamo che il valore della soluzione ottima vale *90,4* ed è inoltre ammissibile in quanto la si ottiene con il seguente percorso corrispondente al ciclo hamiltoniano:

1-6-3-5-2-4-1

Essendo il valore della soluzione ottima e ammissibile minore di tutti quelli trovati fino ad ora possiamo prendere questo valore come riferimento.

Problemi P9 e P10 (ottenuti da P4)

Non ha alcun senso studiare questi casi in quanto il problema P4 dai quali sono stati ottenuti aveva come valore della soluzione ottima *90,7* che è maggiore rispetto a *90,4* ottenuto risolvendo P8. Infatti i problemi P9 e P10, avendo ulteriori vincoli rispetto a P4) non possono avere altro che valori maggiori della soluzione ottima, essendo più piccola la regione di ammissibilità.

A scopo puramente didattico si sono comunque risolti tali problemi ottenendo i valori delle funzioni obiettivo *91,5* (P9) e *92,0* (P10), corrispondenti ai cicli hamiltoniani:

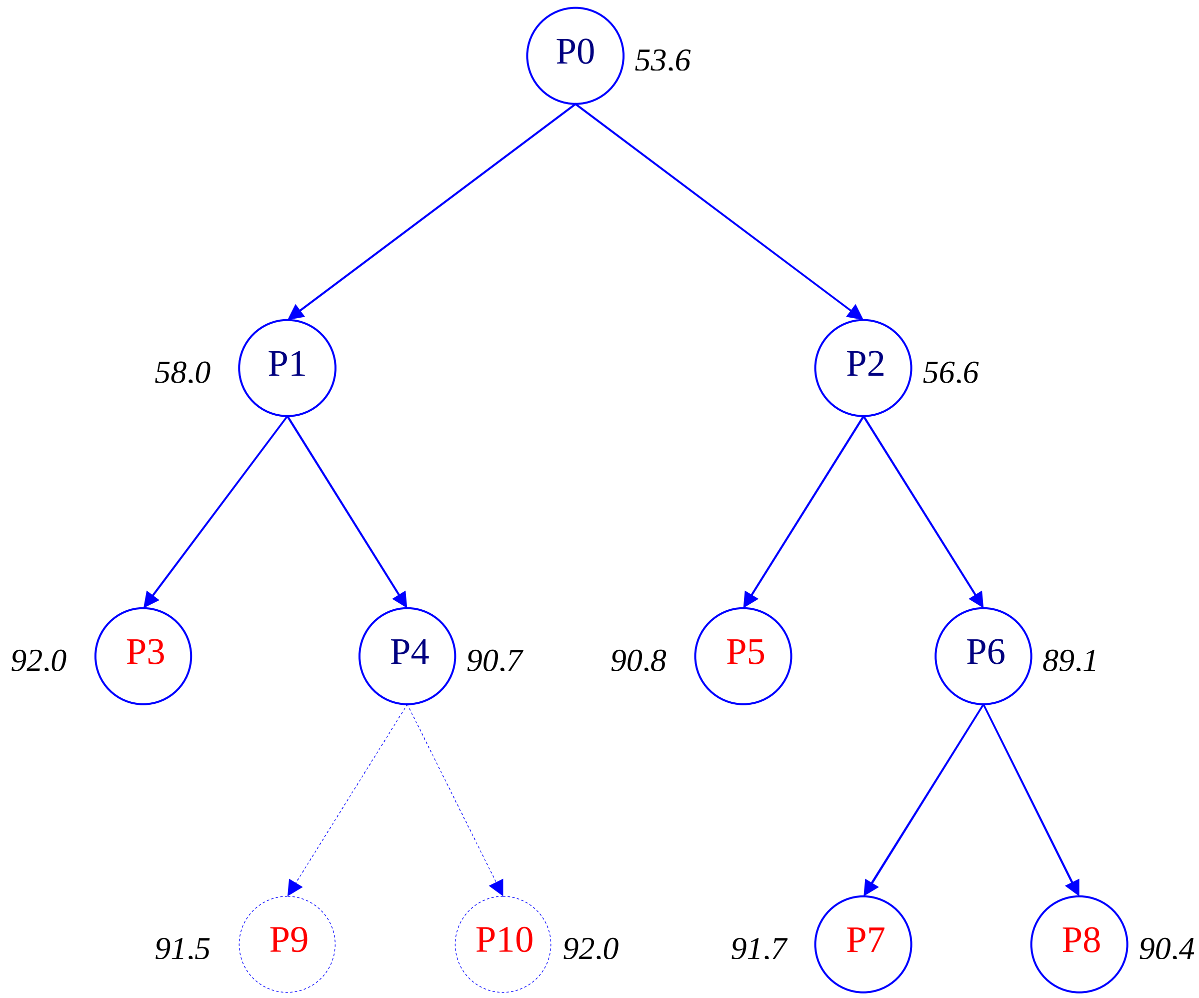
1-3-5-2-4-6-1   *(P9)*
1-3-2-4-5-6-1   *(P10)*

Scelta della soluzione ottima

In base all'analisi dei problemi precedenti la soluzione ammissibile con valore minore è quella del problema P8 e vale *90,4* ottenibile con il percorso:

1-6-3-5-2-4-1

L'utilizzo del metodo di variazione dei coefficienti c_{ij} ha comportato diverse iterazioni. Riporto ora l'albero completo che si è originato a partire da P0:



Legenda:

Indicati in rosso i problemi con soluzione ammissibile

Indicati con il tratteggio i problemi dei quali non è necessaria l'analisi

Applicazione della generazione di vincoli

In questo paragrafo ci utilizzeremo un approccio differente. Dopo aver trovato le soluzioni al *problema di assegnamento associato* P0 procederemo imponendo tutti i vincoli di tipo 4 violati. Non si modificheranno quindi i coefficienti di costo bensì i vincoli. Si potrà notare che sarà necessario studiare un numero molto minore di problemi, tuttavia questi richiederanno un utilizzo maggiore delle risorse di calcolo in quanto non si possono impiegare i metodi ottimizzati per i problemi di assegnamento. L'aggiunta dei vincoli, infatti, non può essere stavolta fatta con semplici modifiche della tabella dei costi e, pertanto, la matrice dei vincoli perde la proprietà di totale unimodularità e il simplesso applicato al rilassamento continuo non garantisce di trovare soluzioni intere. Pertanto bisogna lasciare i vincoli 5 (variabili decisionali binarie, e tale fatto costringe il computer di applicare l'algoritmo del *"Branch & Bound"* per la programmazione lineare mista intera ad ogni iterazione.

Problema P0

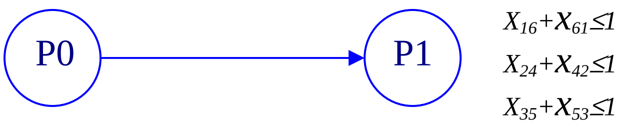
Il problema P0 fornisce si implementa come nel caso precedente. La soluzione ottima vale *53,6* e non è ammissibile.
I sottocicli che corrispondono a P0 sono:

1-6-1, 2-4-2, 3-5-3

Questa volta, a differenza di quanto visto nel paragrafo 2.1 il problema P1 sarà ottenuto con l'inserimento dei vincoli:

$$x_{16} + x_{61} \leq 1$$
$$x_{24} + x_{42} \leq 1$$
$$x_{35} + x_{53} \leq 1$$

La situazione può essere schematizzata con in seguente grafo:



Questa volta non ci saranno ramificazioni ma si procederà aggiungendo un sottoproblemi fino all'ottenimento della soluzione ottima ammissibile.

Problema P1 (ottenuto da P0)

In questo problema la tabella dei costi rimane costante, subiranno una variazione invece i vincoli del problema:

```
min   33.6 x12 + 14.0 x13 + 40.9 x14 + 14.5 x15 + 11.5 x16
      34.7 x21 + 21.7 x23 + 13.0 x24 + 23.5 x25 + 23.4 x26
      14.8 x31 + 21.5 x32 + 29.3 x34 +  2.0 x35 +  3.9 x36
      41.7 x41 + 13.1 x42 + 29.4 x43 + 27.6 x45 + 30.3 x46
      15.0 x51 + 20.2 x52 +  2.0 x53 + 27.5 x54 +  3.9 x56
      12.0 x61 + 22.8 x62 +  2.0 x63 + 30.1 x64 +  4.0 x65

subject to
      riga1)         x12 + x13 + x14 + x15 + x16=1
      riga2)         x21 + x23 + x24 + x25 + x26=1
      riga3)         x31 + x32 + x34 + x35 + x36=1
      riga4)         x41 + x42 + x43 + x45 + x46=1
      riga5)         x51 + x52 + x53 + x54 + x56=1
      riga6)         x61 + x62 + x63 + x64 + x65=1
      col1)          x21 + x31 + x41 + x51 + x61=1
      col2)          x12 + x32 + x42 + x52 + x62=1
      col3)          x13 + x23 + x43 + x53 + x63=1
      col4)          x14 + x24 + x34 + x54 + x64=1
      col5)          x15 + x25 + x35 + x45 + x65=1
      col6)          x16 + x26 + x36 + x46 + x56=1
      sciclo1)       x16+x61<=1
      sciclo2)       x24+x42<=1
      sciclo3)       x35+x53<=1
end

integer              x16
integer              x61
integer              x24
integer              x42
integer              x35
integer              x53
```

In seguito alla risoluzione del problema di ottiene l'output:

```
 LP OPTIMUM FOUND AT STEP     34
 OBJECTIVE VALUE =   89.0999985

 FIX ALL VARS.(    3)  WITH RC >  0.000000E+00

 NEW INTEGER SOLUTION OF    89.0999985     AT BRANCH      0
PIVOT      40
 BOUND ON OPTIMUM:  89.10000
 ENUMERATION COMPLETE. BRANCHES=     0 PIVOTS=      40

 LAST INTEGER SOLUTION IS THE BEST FOUND
 RE-INSTALLING BEST SOLUTION...

        OBJECTIVE FUNCTION VALUE

        1)      89.10000

  VARIABLE        VALUE          REDUCED COST
       X16         1.000000         -1.900000
       X61         0.000000         -4.400000
       X24         1.000000        -34.200001
       X42         0.000000        -33.799999
       X35         0.000000          1.300000
       X53         0.000000          1.400000
       X12         0.000000          0.000000
       X13         0.000000          0.000000
       X14         0.000000          0.000002
       X15         0.000000          0.200000
       X21         0.000000          0.000000
       X23         0.000000          1.400001
       X25         0.000000          2.900000
       X26         0.000000          3.700000
       X31         1.000000          0.000000
       X32         0.000000          1.500000
       X34         0.000000          1.999999
       X36         0.000000          4.100000
       X41         0.000000          0.000000
       X43         0.000000          2.100000
       X45         1.000000          0.000000
       X46         0.000000          3.599999
       X51         0.000000          0.000000
       X52         1.000000          0.000000
       X54         0.000000          0.000000
       X56         0.000000          3.900000
       X62         0.000000          1.199999
       X63         1.000000          0.000000
       X64         0.000000          1.200000
       X65         0.000000          1.700000


       ROW   SLACK OR SURPLUS     DUAL PRICES
    RIGA1)         0.000000          0.000000
    RIGA2)         0.000000         -6.300000
    RIGA3)         0.000000         13.600000
    RIGA4)         0.000000        -13.300000
    RIGA5)         0.000000         13.400000
    RIGA6)         0.000000         12.000000
     COL1)         0.000000        -28.400000
     COL2)         0.000000        -33.599998
     COL3)         0.000000        -14.000000
     COL4)         0.000000        -40.900002
     COL5)         0.000000        -14.300000
     COL6)         0.000000        -13.400000
  SCICLO1)         0.000000          0.000000
  SCICLO2)         0.000000          0.000000
  SCICLO3)         1.000000          0.000000

 NO. ITERATIONS=      50
 BRANCHES=    0 DETERM.=  1.000E    0
```

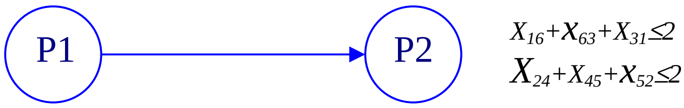
Il valore della soluzione ottima di P1 vale *89,1* ma non è ammissibile in quanto è composta dei cicli:

$$1\text{-}6\text{-}3\text{-}1,\ 2\text{-}4\text{-}5\text{-}2$$

I vincoli violati questa volta sono:

$$x_{16} + x_{63} + x_{31} \leq 2$$
$$x_{24} + x_{45} + x_{52} \leq 2$$

Il problema P2 sarà ottenuto a partire da P1 con l'aggiunta dei due vincoli sopra indicati:



Problema P2 (ottenuto da P1)

Implementando i vincoli al LINDO si ottiene:

```
min   33.6 x12 + 14.0 x13 + 40.9 x14 + 14.5 x15 + 11.5 x16
      34.7 x21 + 21.7 x23 + 13.0 x24 + 23.5 x25 + 23.4 x26
      14.8 x31 + 21.5 x32 + 29.3 x34 +  2.0 x35 +  3.9 x36
      41.7 x41 + 13.1 x42 + 29.4 x43 + 27.6 x45 + 30.3 x46
      15.0 x51 + 20.2 x52 +  2.0 x53 + 27.5 x54 +  3.9 x56
      12.0 x61 + 22.8 x62 +  2.0 x63 + 30.1 x64 +  4.0 x65

subject to
      riga1)         x12 + x13 + x14 + x15 + x16=1
      riga2)         x21 + x23 + x24 + x25 + x26=1
      riga3)         x31 + x32 + x34 + x35 + x36=1
      riga4)         x41 + x42 + x43 + x45 + x46=1
      riga5)         x51 + x52 + x53 + x54 + x56=1
      riga6)         x61 + x62 + x63 + x64 + x65=1
      col1)          x21 + x31 + x41 + x51 + x61=1
      col2)          x12 + x32 + x42 + x52 + x62=1
      col3)          x13 + x23 + x43 + x53 + x63=1
      col4)          x14 + x24 + x34 + x54 + x64=1
      col5)          x15 + x25 + x35 + x45 + x65=1
      col6)          x16 + x26 + x36 + x46 + x56=1
      sciclo1)       x16+x61<=1
      sciclo2)       x24+x42<=1
      sciclo3)       x35+x53<=1
      sciclo4)       x16+x63+x31<=2
      sciclo5)       x24+x45+x52<=2
end
```

```
integer             x16
integer             x61
integer             x24
integer             x42
integer             x35
integer             x53
integer             x63
integer             x31
integer             x45
integer             x52
```

Elaborando il listato si ottiene:

```
LP OPTIMUM FOUND AT STEP     40
 OBJECTIVE VALUE =   90.0222244

 SET       X24 TO <=     0 AT    1, BND=  -90.80     TWIN= -
90.40         45

 NEW INTEGER SOLUTION OF    90.8000031     AT BRANCH      1
PIVOT      45
 BOUND ON OPTIMUM:  90.40000
 FLIP       X24 TO >=         1 AT    1 WITH BND=   -
90.400002

 NEW INTEGER SOLUTION OF    90.4000015     AT BRANCH      1
PIVOT      45
 BOUND ON OPTIMUM:  90.40000
 DELETE      X24 AT LEVEL     1
 ENUMERATION COMPLETE. BRANCHES=     1 PIVOTS=      45

 LAST INTEGER SOLUTION IS THE BEST FOUND
 RE-INSTALLING BEST SOLUTION...

        OBJECTIVE FUNCTION VALUE

        1)      90.40000

  VARIABLE        VALUE          REDUCED COST
       X16         1.000000         -8.200000
       X61         0.000000        -11.900000
       X24         1.000000        -13.900000
       X42         0.000000        -27.500000
       X35         1.000000         51.400002
       X53         0.000000          7.700000
       X63         1.000000         -1.200000
       X31         0.000000         -4.100000
       X45         0.000000         68.199997
       X52         1.000000         20.300001
       X12         0.000000          0.000000
       X13         0.000000          0.000000
       X14         0.000000          0.000000
       X15         0.000000         48.099998
       X21         0.000000          0.000000
       X23         0.000000          7.700001
       X25         0.000000         57.099998
       X26         0.000000          3.700000
       X32         0.000000          3.700000
       X34         0.000000          4.199999
       X36         0.000000          0.000000
       X41         1.000000          0.000000
       X43         0.000000          8.400000
       X46         0.000000          3.599999
       X51         0.000000          0.000000
       X54         0.000000          6.300000
       X56         0.000000          3.900000
       X62         0.000000         -0.000001
       X64         0.000000          0.000000
       X65         0.000000         48.400002


       ROW   SLACK OR SURPLUS     DUAL PRICES
    RIGA1)         0.000000          0.000000
    RIGA2)         0.000000          0.000000
    RIGA3)         0.000000         15.800000
```

```
            RIGA4)         0.000000         -7.000000
            RIGA5)         0.000000         19.700001
            RIGA6)         0.000000         10.800000
             COL1)         0.000000        -34.700001
             COL2)         0.000000        -33.599998
             COL3)         0.000000        -14.000000
             COL4)         0.000000        -40.900002
             COL5)         0.000000         33.599998
             COL6)         0.000000        -19.700001
          SCICLO1)         0.000000          0.000000
          SCICLO2)         0.000000          0.000000
          SCICLO3)         0.000000          0.000000
          SCICLO4)         0.000000          0.000000
          SCICLO5)         0.000000         14.000000

 NO. ITERATIONS=      47
 BRANCHES=    1 DETERM.=  1.000E    0
```

Questa volta la funzione obiettivo vale *90,4* ed è ammissibile. Il ciclo hamiltoniano corrispondente è:

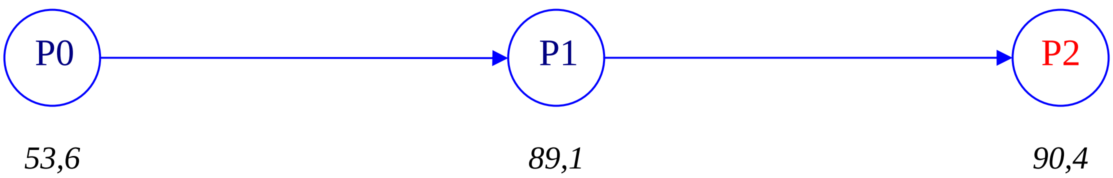
1-6-3-5-2-4-1

Scelta della soluzione ottima

La soluzione ammissibile con valore minore è *90,4* ottenibile con il percorso:

1-6-3-5-2-4-1

Questa volta si è proceduto in modo nettamente diverso: ogni problema infatti dava origine ad un unico sottoproblema del quale bisognava stabilire se la soluzione fosse ammissibile o no (a seconda che il percorso formasse un ciclo hamiltoniano o no). Infatti una volta trovata la prima soluzione ammissibile un ulteriore passaggio, con la corrispondente aggiunta di un vincolo, comporterebbe solo l'aumento della funzione obiettivo in quanto si restringe la regione di ammissibilità.



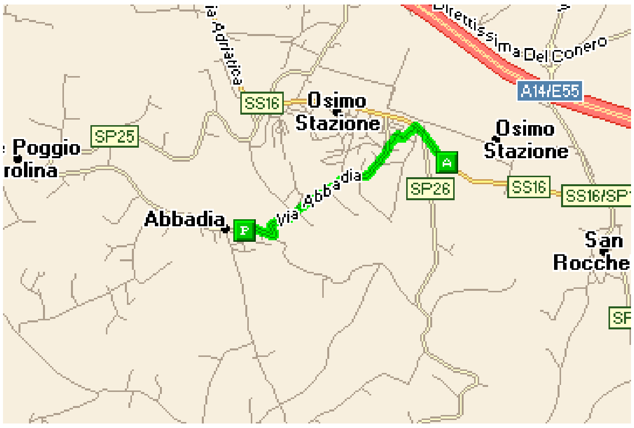
Legenda:

Indicato in rosso il problema con soluzione ammissibile

Paragrafo 2.3 Conclusioni

Entrambe i metodi proposti giungono allo stesso risultato sia in termini di valore che in termini di percorso ottimo.

Il valore della funzione obiettivo è 90,4 e si ottiene per:



Occorre ora effettuare alcune considerazioni utili a mettere in luce vantaggi e svantaggi di questi due approcci.

Il primo metodo richiede esigue risorse di calcolo per risolvere ciascun sottoproblema. Si risolvono infatti *problemi di assegnamento* per i quali esistono metodi specifici di risoluzione. Un altro vantaggio è che il numero di variabili e di vincoli è costante dal primo all'ultimo sottoproblema: i vincoli di eliminazione del percorso vengono aggiunti con lo stratagemma della modifica dei coefficienti c_{ij}. Come rovescio della medaglia si deve prestare attenzione alla scelta e alla gestione dei sottoproblemi: questi, specie nel caso di molte località, potrebbero diventare molto numerosi e potrebbe essere necessario uno studio di parecchi casi prima di determinare la soluzione ammissibile ed ottima. Un ulteriore inconveniente è rappresentato dal fatto che, anche dopo aver trovato la soluzione che in seguito si rivelerà ottima, potrei dover studiare altri sottoproblemi e verificare che hanno valori della funzione obiettivo peggiori.

Il metodo che invece utilizza i vincoli di eliminazione dei sottocicli presenta caratteristiche molto diverse. Esso richiede, ogni volta che si procede con un sottoproblema successivo, l'introduzione di almeno un vincolo, il quale potrebbe essere piuttosto complesso quando il numero di località del sottociclo trovato sono numerose. Le risorse di calcolo richieste per ogni iterazione in questo tipo di approccio sono elevate: basti pensare che non si può ricorrere a metodi per programmazione lineare, come nel caso precedente, visto che, non avendo più garanzie sull'interezza delle variabili, occorre imporre l'interezza delle variabili (in realtà basta imporre l'interezza per le sole variabili che entrano nei vincoli di eliminazione di sottocicli). Per questo motivo ogni sottoproblema successivo sarà un problema di *programmazione lineare misto* e richiederà l'utilizzo della tecnica del "*Branch & Bound*" da parte del calcolatore. I vantaggi derivano da un generalmente inferior numero di iterazioni da svolgere per arrivare ad una soluzione ammissibile (e quindi ottima).