

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

SEDE DI CESENA

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI

Corso di Laurea in Scienze e Tecnologie Informatiche

Metodi Euristici per il Problema dei Trasporti con Costi Fissi

Relazione finale in

RICERCA OPERATIVA

Relatore:

Prof. Aristide Mingozzi

Presentata da:

Emanuele Reggi

Correlatore:

Dott. Roberto Roberti

Sessione 2

Anno Accademico 2012 · 2013

Indice

| | |
|---|-----------|
| 1. Introduzione..... | 7 |
| 2. Il Problema dei Trasporti con Costi Fissi | 9 |
| 2.1 Applicazioni a problemi reali | 10 |
| 2.1.1 L'assegnamento dei veicoli di lancio alle missioni spaziali | 10 |
| 2.1.2 L'assegnamento dei docenti..... | 11 |
| 2.2 Problema della rete di flusso minimo a costo concavo | 12 |
| 2.2.1 Complessità di un MCNFP non capacitato | 12 |
| 2.3 Formulazione matematica classica | 16 |
| 2.3.1 Rilassamento lineare standard | 17 |
| 2.4 Formulazione matematica a tre indici | 17 |
| Proposizione 2.1..... | 18 |
| Proposizione 2.2..... | 19 |
| 2.5 Una nuova formulazione matematica intera..... | 20 |
| Proposizione 2.3..... | 21 |
| Proposizione 2.4..... | 22 |
| 3. Metodi di risoluzione esatti | 25 |
| 3.1 Il metodo di Barr | 25 |
| 3.1.1 Test di pre-processing..... | 26 |
| 3.1.2 Local star..... | 26 |
| 3.1.3 Strategie di separazione e branching..... | 27 |
| 3.2 Penalties di Driebeek e di Cabot e Erenguc | 27 |
| 3.3 Penalties di Palekar, Karwan e Zionts | 29 |
| 3.4 Penalties di Lamar e Wallace | 30 |
| 3.5 Il metodo di Agarwal e Aneja | 31 |
| 3.5.1 La formulazione di Agarwal e Aneja..... | 31 |

| | |
|---|-----------|
| 3.5.2 Facets, euristici e metodo | 33 |
| 3.5.2.1 Facets a copertura d'insieme | 33 |
| 3.5.2.2 Facets del secondo ordine e facets estese | 35 |
| 3.5.2.3 Separazione delle disequazioni..... | 35 |
| 3.6 Il metodo esatto di Roberti, Bartolini e Mingozzi (2013)..... | 36 |
| 3.6.1 Disequazioni valide | 37 |
| 3.6.1.1 Disequazioni Set Covering (SC)..... | 37 |
| 3.6.1.2 Disequazioni Upper Bound Cover estese (EGUBC)..... | 38 |
| 3.6.1.3 Disequazioni di ammissibilità (FSB)..... | 39 |
| 3.6.1.4 Disequazioni Couple (CPL)..... | 40 |
| 3.6.1.5 Disequazioni di Chvátal-Gomory (CGD-CGU)..... | 41 |
| 3.6.1.6 Lifting delle disequazioni di Chvátal-Gomory (LCGD) | 42 |
| 3.6.2 Risoluzione del problema di pricing..... | 43 |
| 3.6.3 Un esatto per la risoluzione del FCTP..... | 43 |
| 3.6.3.1 Procedura di bounding..... | 43 |
| 3.6.3.2 Strategia di branching..... | 44 |
| 4. Metodi di risoluzione euristici | 47 |
| 4.1 Il metodo di Cooper | 48 |
| Teorema 4.1 | 49 |
| 4.1.1 L'algoritmo..... | 50 |
| 4.2 Il metodo di Adlakha e Kowalski | 51 |
| 4.2.1 Fase 1: soluzione ammissibile di partenza | 51 |
| 4.2.2 Fase 2: miglioramento e verifica dell'ottimo | 52 |
| 4.3 Il metodo di Aguado | 52 |
| 4.3.1 Rilassamento e Decomposizione Lagrangiana..... | 53 |
| 4.3.2 Generazione dei sotto problemi..... | 56 |
| 4.3.3 Euristico per la risoluzione dei sotto problemi..... | 58 |

| | |
|--|-----------|
| 4.4 Il metodo di Glover, Amini e Kochenberger..... | 58 |
| 4.4.1 Step 0 | 59 |
| 4.4.2 Step 1 | 60 |
| 4.4.3 Step 2: Fase I..... | 60 |
| 4.4.4 Step 2: Fase II | 61 |
| 4.4.5 Step 3 | 62 |
| 4.5 Il metodo euristico di Buson, Roberti e Toth (2013)..... | 63 |
| 4.5.1 Disequazioni valide per il problema | 63 |
| 4.5.2 Procedura di Lower Bounding..... | 65 |
| 4.5.3 Local Search | 66 |
| 4.5.4 Restart Phase..... | 67 |
| 4.5.5 L'euristico | 68 |
| 4.5.6 Risultati computazionali | 68 |
| 5. Tabu Search..... | 71 |
| 5.1 Concetti fondamentali | 71 |
| 5.1.1 Mossa | 71 |
| 5.1.2 Soluzioni adiacenti e pricing..... | 72 |
| 5.1.3 Condizioni tabu e livello di aspirazione | 73 |
| 5.1.4 Intensificazione, diversificazione e criterio di arresto | 74 |
| 5.1.5 Soluzione base ammissibile di partenza | 75 |
| 5.2 L'algoritmo..... | 75 |
| 5.2.1 Inizializzazione | 75 |
| 5.2.2 Processo di memoria a breve termine | 76 |
| 5.2.3 Processo di memoria a medio termine 1 | 76 |
| 5.2.4 Processo di memoria a medio termine 2..... | 76 |
| 5.2.5 Processo di memoria a lungo termine 1 | 77 |
| 5.2.6 Processo di memoria a lungo termine 2..... | 77 |

Indice

| | |
|--|-----------|
| 5.3 Implementazione e risultati computazionali | 78 |
| 6. Conclusioni | 81 |
| Bibliografia | 83 |

1. Introduzione

In questa tesi viene considerato il *Problema dei Trasporti con Costi Fissi* (FCTP) che, assieme al *Traveling Salesman Problem* (TSP), è uno dei problemi nobili dell'ottimizzazione combinatoria. Esso generalizza il ben noto *Problema dei Trasporti* (TP) imponendo che il costo per spedire prodotti da un'origine ad una destinazione sia composto da un costo fisso ed un costo proporzionale alla quantità spedita.

Molti problemi pratici di trasporto e distribuzione possono essere modellati come FCTP. Per esempio, in un sistema di trasporto, si può incorrere in un costo fisso per ogni spedizione tra una data origine e una data destinazione; in un problema di ubicazione dei complessi industriali, ciò può presentarsi come investimento fisso per la creazione o espansione di uno stabilimento o di un magazzino. Inoltre, molti problemi pratici, non concernenti il trasporto di prodotti, possono essere formulati come FCTP.

Il FCTP è stato formulato per la prima volta in un articolo di Hirsch e Dantzig (1968) ed è stato da allora oggetto di studio per la ricerca di nuovi e sempre migliori algoritmi di risoluzione. Nessuno dei metodi esatti fin ora pubblicati è in grado di risolvere istanze con al più 15 origini e 15 destinazioni. Solo recentemente, Roberti et al. (2013), in un paper in corso di pubblicazione, hanno presentato un metodo esatto basato su una nuova formulazione matematica del problema, il quale è in grado di risolvere istanze di FCTP con 70 origini e 70 destinazioni.

La crescita esponenziale dello sforzo computazionale richiesto dai metodi esatti ne ha confinato l'applicazione a problemi di dimensioni ridotte.

Tali limitazioni hanno portato allo studio e alla ricerca di approcci approssimativi, euristici e metaeuristici i quali sfruttano varie strategie di *local search*. Fra i molteplici metodi euristici presentati in letteratura, meritano particolare attenzione quelli di Sun et al. (1998) e Glover et al. (2005). Recentemente, Buson et al. (2013) hanno presentato un nuovo euristico che domina tutti i precedenti sui problemi test proposti in letteratura.

Capitolo 1. Introduzione

In questa tesi viene presentato un approccio *Tabu Search* che migliora il metodo originalmente proposto da Sun et al. (1998). I risultati computazionali ottenuti con un codice prototipale indicano che l'algoritmo sviluppato è migliore del metodo originario di Sun et al. (1998) e competitivo con il più recente metodo proposto da Buson et al. (2013).

La tesi si articola come segue. Nel capitolo 2 viene descritto il FCTP, vengono presentate alcune applicazioni pratiche e le formulazioni matematiche più rilevanti. Nel capitolo 3 viene presentata una rassegna dei metodi esatti proposti in letteratura. Nel capitolo 4 vengono descritti i principali metodi euristici. Nel capitolo 5 viene descritto il metaeuristico di Sun et al. (1998) ed i miglioramenti apportati in questa tesi a tale metodo. Inoltre, vengono presentati i risultati computazionali ottenuti su un insieme di problemi test. Infine, nel capitolo 6, vengono riassunte le nostre conclusioni.

2. Il Problema dei Trasporti con Costi Fissi

Il Problema dei Trasporti con Costi Fissi (FCTP) è una nota generalizzazione del Problema dei Trasporti (TP) ove il costo per spedire un flusso da un'origine ad una destinazione è uguale ad un costo variabile proporzionale al flusso più un costo fisso.

Il FCTP è definito come segue. Sono date n origini ed m destinazioni. Ad ogni origine i è disponibile una quantità a_i di prodotto, mentre ogni destinazione j richiede un quantitativo b_j del prodotto disponibile alle origini. Senza perdita di generalità, si assume che il problema sia *bilanciato*, ovvero che la quantità di prodotto complessivamente disponibile alle origini sia pari alla quantità complessiva richiesta dalle destinazioni (i.e. $\sum_{i=1}^n a_i = \sum_{j=1}^m b_j$). Inoltre, si assume che a_i e b_j siano numeri interi.

Il costo per trasportare un quantitativo non nullo q dall'origine i alla destinazione j è pari a $f_{ij} + c_{ij}q$, dove f_{ij} rappresenta il costo fisso per inviare un qualsiasi quantitativo dall'origine i alla destinazione j , mentre c_{ij} rappresenta il costo per spedire un'unità di prodotto.

Il FCTP consiste nel determinare il piano di trasporto di costo minimo per inviare alle destinazioni tutto il prodotto disponibile alle origini.

Il FCTP è un caso speciale del problema del flusso massimo di un unico prodotto a costo minimo in un grafo orientato $G = (V, E)$ dove V contiene n vertici origine, m vertici destinazione e $|V| - n - m$ vertici di transito. Ad ogni arco $(i, j) \in E$ è associato sia un costo fisso f_{ij} che un costo unitario c_{ij} . Questo problema più generale si riduce al FCTP nel caso in cui non esistano vertici di transito e il grafo sia bipartito come $V = (S, T)$, dove S è l'insieme delle n origini, T l'insieme delle m destinazioni e $E \subseteq \{(i, j): i \in S, j \in T\}$.

Il FCTP rientra nell'ordine di complessità dei problemi NP-Hard (Guisewite e Pardalos (1990)), ossia esiste un problema NP-completo Turing-riducibile in tempo polinomiale al FCTP. Nel loro paper Guisewite e Pardalos si riferiscono, più in

generale, ai problemi di reti di flusso minimo a costo concavo (MCNFP), dei quali il FCTP costituisce un caso speciale.

Nel paragrafo 2.2 verrà descritto il MCNFP ed enunciato il teorema che mostra come il MCNFP sia *NP-Hard*.

2.1 Applicazioni a problemi reali

Il FCTP sopradescritto rappresenta un valido modello di molteplici problemi applicativi nel settore dei sistemi di distribuzione e trasporti, ove i costi fissi rappresentano, ad esempio, pedaggi autostradali, tasse doganali e/o aereoportuali di atterraggio, costi di avviamento dei sistemi di produzione, costi per costruire strade o costi di noleggio dei mezzi di trasporto.

Oltre alle applicazioni ai sistemi di distribuzione e trasporti (Palekar et al. (1990), Adlakha e Kowalsky (2003)), sono molteplici le applicazioni del FCTP a problemi applicativi, quali ad esempio, l'assegnamento dei veicoli di lancio alle missioni spaziali Stroup (1966), la selezione dei processi (Hirsch e Dantzig (1968)), l'ubicazione di conti bancari (Cornuejols et al. (1977)), del centro di distribuzione (Nozick e Turnquist (1998), Nozick e Turnquist (1998)) e delle centrali nucleari (Dutton et al. (1974)), la gestione dei rifiuti solidi (Walker (1976)), i sistemi di acque reflue (Jarvis et al. (1978)), l'assegnamento dei docenti (Hultberg e Cardoso (1997)), i sistemi di tubature di gas naturale (Rothfarb et al. (1970)), le piattaforme di trivellazione in mare aperto (Balas e Padberg (1976)) e la commutazione delle reti di telecomunicazione (Luna et al. (1987)).

2.1.1 L'assegnamento dei veicoli di lancio alle missioni spaziali

John William Stroup (1966), ha proposto un'applicazione del FCTP al problema dell'assegnamento dei veicoli di lancio alle missioni spaziali. Il problema consiste nel selezionare quali progetti di veicoli debbano essere realizzati e, successivamente, nell'assegnare gli stessi, nonché i veicoli già esistenti, alle future missioni spaziali, al fine di minimizzare il costo complessivo di queste ultime.

Si considerino n missioni che debbano essere eseguite da, al massimo, m tipi di veicoli di lancio. L' i -esimo tipo di veicolo ha associato un costo di sviluppo, non ricorrente, D_i . Ogni veicolo dell' i -esimo tipo costa una somma a_i e ha una

affidabilità r_i . Associato con la j -esima missione, vi è un costo b_j . Infine, un costo e_{ij} rappresenta il costo delle modifiche da effettuare su un veicolo dell' i -esimo tipo perché possa effettuare la j -esima missione.

Definiamo il costo $c_{ij} = \frac{a_i + b_j + e_{ij}}{r_i}$ come il costo atteso per effettuare la j -esima missione con un veicolo dell' i -esimo tipo e sia x_{ij} la variabile che assume valore 1 se il veicolo di tipo i è assegnato alla missione j , 0 altrimenti. Il problema diviene, quindi, un FCTP ove i costi c_{ij} sopra definiti rappresentano i costi unitari di trasporto mentre i costi fissi sono dati da $f_{ij} = D_i, j = 1, \dots, m$.

2.1.2 L'assegnamento dei docenti

Tim Hultberg e Domingos Cardoso (1994) hanno formulato il problema dell'assegnamento di un insieme J di n docenti a un insieme I di m materie come un caso speciale del FCTP.

Ogni soluzione ammissibile di questo problema deve soddisfare i seguenti vincoli.

$$\sum_{j \in J} x_{ij} = h_i \quad \forall i \in I \quad (2.1)$$

$$\sum_{i \in I} x_{ij} + s_j = b_j \quad \forall j \in J \quad (2.2)$$

dove $x_{ij} \geq 0$ è il numero di ore della materia i insegnata dal docente j , h_i il numero totale di ore di lezione della materia i (il prodotto tra ore settimanali della materia ed il numero di classi richieste per la materia), e b_j il numero massimo di ore di lezione che il docente j può tenere. Per convenienza sono state introdotte delle variabili di scarto $s_j \geq 0$ ($j \in J$) nell'equazione (2.2) per rappresentare le ore lavorative non sfruttate dal docente j . Per contare il numero di materie distinte, introduciamo delle variabili binarie d_{ij} che indicano se il docente j stia insegnando la materia i o meno. Questo problema equivale dunque ad un caso speciale del Pure Fixed-Charge Transportation Problem (PFCTP), un particolare tipo di FCTP in cui tutti i costi unitari sono nulli, nel quale tutti i costi fissi sono uguali a 1.

2.2 Problema della rete di flusso minimo a costo concavo

Il problema della rete di flusso minimo a costo concavo (MCNFP) può essere definito formalmente come segue:

Dato un grafo direzionato $G = (N, A)$ consistente di un insieme N di n nodi ed un insieme A di m coppie ordinate di nodi distinti detti archi, accoppiati con un n -vettore $d = (d_i)$ (vettore delle richieste) ed una funzione concava di costo per ogni arco, $c_{ij}(x_{ij})$, allora si risolve

$$\begin{aligned} \text{Min} \quad & \sum_{(i,j) \in A} c_{ij}(x_{ij}) \\ \text{s. t.} \quad & \sum_{(i,k) \in A} x_{ik} - \sum_{(k,i) \in A} x_{ki} = d_i \quad \forall i \in N \quad (2.3) \\ & 0 \leq a_{ij} \leq x_{ij} \leq b_{ij} \quad \{i,j\} \in A \quad (2.4) \end{aligned}$$

Si suppone che tutti i vincoli e le richieste siano interi. Un sistema consistente soddisfa il vincolo $\sum_{i=1}^n d_i = 0$, che afferma che la disponibilità totale equivale alla richiesta totale. I vincoli in (2.3) sono detti di conservazione del flusso, mentre quelli in (2.4) sono detti vincoli di capacità sui flussi degli archi. Ai nodi con $d_i > 0$ corrispondono sorgenti, mentre a quelli con $d_i < 0$ corrispondono destinazioni. Un problema di questo genere è detto non capacitato se $a_{ij} = 0$ e $b_{ij} = \infty, \forall (i,j) \in A$.

2.2.1 Complessità di un MCNFP non capacitato

La seguente trasformazione dimostra come il MCNFP sia NP-Hard per casi riguardanti strettamente funzioni obiettivo concave.

Si consideri il problema noto come *Subset Sum Problem* (SSP), il quale è noto essere NP-completo. Siano dati un insieme finito D , una dimensione $s(d) \in Z^+$, per ogni $d \in D$ ed un intero positivo B .

Il SSP consiste nel verificare se esiste un sottoinsieme $D' \subseteq D$ tale che $\sum_{d \in D'} s(d) = B$.

Costruiamo il seguente problema di flusso con $N = |D|$:

- (1) Creiamo N vertici sorgente, V_i , con flusso uguale a $s(i)$.

2.2 Problema della rete di flusso minimo a costo concavo

- (2) Creiamo due vertici destinazione S_B e $S_{\bar{B}}$, con flusso negativo B e $\sum_{d \in D} s(d) - B$.
- (3) Creiamo un arco da ogni sorgente ad ogni destinazione, (V_d, S_B) e $(V_d, S_{\bar{B}})$, per ogni $d \in D$.
- (4) Gli archi sono non capacitati.

La rete risultante è mostrata in figura 2.1. Consideriamo le seguenti funzioni di flusso:

- (1) Costi di messa in opera (assumiamo che i nodi siano etichettati $1, 2, \dots, N + 2$): questo caso include
 - (a) $c_{ij}(0) = 0, c_{ij}(x) = a, a > 0$
 - (b) $c_{ij}(0) = 0, c_{ij}(x) = a + bx, 0 < b < a$
- (2) Funzioni di costo che soddisfano: $c_{ij}(0) = 0, c_{ij}(x + y) < c_{ij}(x) + c_{ij}(y), x, y > 0$. Questo caso include tutte le funzioni strettamente concave, un numero di non concave, funzioni non convesse e tutte le funzioni relative ai costi fissi.

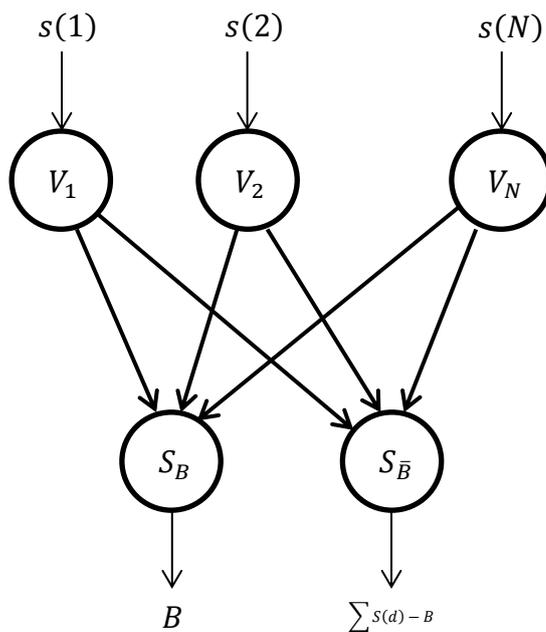


Figura 2.1: flusso risultante dalla trasformazione.

Per il caso 1, consideriamo quando tutti gli archi hanno costo 1 per flussi non nulli. Allora il costo ottimo del problema è N se la risposta alla domanda del SSP è sì. In generale, se il costo su ogni arco è $c_{ij}(x) = a + bx$, allora il costo ottimo del problema è $(N \cdot a) + (b \cdot \sum_{i=1}^N s(i))$ se e solo se la risposta alla domanda del SSP è sì. In entrambi i casi il risultato segue immediatamente dal notare che se qualsiasi flusso è spezzato, allora incorriamo in costi di messa in opera aggiuntivi.

Per il caso 2, consideriamo quando i costi di (V_i, S_B) e $(V_i, S_{\bar{B}})$ sono identici e soddisfano i vincoli specificati. Qui troviamo che il costo ottimo del problema è $\sum_{i=1}^N c_{V_i, S_B}(s(i))$ se e solo se la risposta alla domanda del SSP è sì. Ciò segue dall'osservazione che se non si verifica alcuno spezzamento dei flussi, il costo è come specificato sopra e gli archi del flusso costituiscono una soluzione al problema del sottoinsieme somma. Se qualsiasi spezzamento si verifica, allora il costo aumenta.

2.2 Problema della rete di flusso minimo a costo concavo

Per il caso con flusso non spezzato:

$$\begin{aligned}
 Costo &= \sum_{i=1}^N c_{V_i, S_B}(x_{i, S_B}) + \sum_{i=1}^N c_{V_i, S_{\bar{B}}}(x_{i, S_{\bar{B}}}) = \\
 &= \sum_{i=1}^N c_{V_i, S_B}(s(i)) = \\
 &= \sum_{i=1}^N c_{V_i, S_{\bar{B}}}(s(i)).
 \end{aligned}$$

Per il caso con il flusso diviso a V_m :

$$\begin{aligned}
 Costo &= \sum_{i=1}^N c_{V_i, S_B}(x_{i, S_B}) + \sum_{i=1}^N c_{V_i, S_{\bar{B}}}(x_{i, S_{\bar{B}}}) = \\
 &= \left(\sum_{i=1, i \neq m}^N c_{V_i, S_B}(s(i)) \right) + c_{V_m, S_B}(x_{m, S_B}) + c_{V_m, S_{\bar{B}}}(x_{m, S_{\bar{B}}}) > \\
 &> \sum_{i=1}^N c_{V_i, S_B}(s(i)).
 \end{aligned}$$

Lo studio del suddetto problema rivela che quando tutti gli archi hanno costo pari a 1 per flussi non nulli, è triviale trovare un flusso con costo uno più del costo di flusso ottimo (i.e. forzare i flussi da V_i a S_B fino a raggiungere tale flusso). Ciò porta un semplice esempio in cui tecniche iterative di ricerca, come branch-and-bound, risultano inefficienti anche se è possibile ottenere un buon upper bound iniziale per il costo del flusso. In aggiunta, una qualsiasi soluzione ottima a tale problema implica un flusso ammissibile con un numero minimo di archi attivi (flusso non nullo). Ciò indica come il problema di calcolare un flusso ammissibile con tale proprietà sia NP-Hard.

Inoltre, la rete sottesa nella trasformazione di cui sopra è bipartita. Ciò indica che il problema generale delle reti con flusso a costo concavo è NP-Hard quando ristretto a reti bipartite, anche per quei casi che non coinvolgano costi fissi sui flussi sugli archi.

2.3 Formulazione matematica classica

Il FCTP può essere definito formalmente come segue. Siano dati un insieme $S = \{1, 2, \dots, n\}$ di n origini ed un insieme $T = \{1, 2, \dots, m\}$ di m destinazioni. Ogni origine $i \in S$ ha una quantità positiva disponibile $a_i \in N$ di prodotto e ogni destinazione $j \in T$ una quantità positiva richiesta $b_j \in N$ – assumiamo, senza perdita di generalità, che $\sum_{i \in S} a_i = \sum_{j \in T} b_j$, ossia che la somma delle quantità disponibili alle origini sia uguale alla somma delle quantità richieste dalle destinazioni. È dato, inoltre, un grafo bipartito completo $G = (S, T, E)$, dove E è definito come $E = \{\{i, j\}: i \in S, j \in T\}$. Un costo unitario variabile non negativo c_{ij} per trasportare un'unità di prodotto dall'origine $i \in S$ alla destinazione $j \in T$ ed un costo fisso non negativo f_{ij} per l'utilizzo dell'arco $\{i, j\}$ sono associati con l'arco $\{i, j\} \in E$. Il FCTP richiede di minimizzare il costo totale per trasportare tutti i prodotti dalle origini alle destinazioni.

Per ogni arco $\{i, j\} \in E$, definiamo una variabile non negativa x_{ij} per denotare la quantità di beni (flusso) trasportati attraverso l'arco ed una variabile binaria y_{ij} uguale a 1 se e solo se la variabile x_{ij} corrispondente è positiva (0 altrimenti). Sia m_{ij} il flusso massimo attraverso l'arco $\{i, j\} \in E$ definito come $m_{ij} = \min\{a_i, b_j\}$. Il FCTP può essere formulato come il seguente MILP (Mixed Integer Linear Problem)

$$\begin{cases}
 z = \text{Min} & \sum_{\{i,j\} \in E} c_{ij}x_{ij} + \sum_{\{i,j\} \in E} f_{ij}y_{ij} & (2.5) \\
 \text{s. t.} & \sum_{j \in T} x_{ij} = a_i & i \in S & (2.6) \\
 & \sum_{i \in S} x_{ij} = b_j & j \in T & (2.7) \\
 & x_{ij} \leq m_{ij}y_{ij} & \{i, j\} \in E & (2.8) \\
 & x_{ij} \geq 0 & \{i, j\} \in E & (2.9) \\
 & y_{ij} \in \{0, 1\} & \{i, j\} \in E & (2.10)
 \end{cases}$$

La funzione obiettivo (2.5) richiede di minimizzare il costo totale (fisso e variabile). I vincoli (2.6) richiedono che tutti i beni disponibili ad ogni origine $i \in S$ siano spediti. I vincoli (2.7) obbligano, in qualsiasi soluzione ammissibile, a spedire esattamente b_j unità di beni ad ogni destinazione $j \in T$. I vincoli (2.8)

accoppiano le variabili x_{ij} e y_{ij} . Infine, i vincoli (2.9) – (2.10) definiscono gli intervalli di validità per le variabili x_{ij} e y_{ij} , rispettivamente.

2.3.1 Rilassamento lineare standard

Denotiamo con P' il rilassamento lineare del problema P , ove i vincoli (2.10) sono sostituiti da $y_{ij} \geq 0, \forall \{i, j\} \in E$, e con z' il costo della sua soluzione ottima. I vincoli $y_{ij} \leq 1, \forall \{i, j\} \in E$ possono essere ignorati. In qualsiasi soluzione ottima di P' , le variabili $x_{ij} > 0$ corrispondono ad una soluzione base ammissibile (BFS) dei vincoli (2.6) – (2.7) e le variabili y_{ij} sono uguali a $y_{ij} = \frac{x_{ij}}{m_{ij}}, \forall \{i, j\} \in E$ (Hirsch e Dantzig (1954)). Allora z' è uguale al costo della soluzione ottima del seguente problema (Balinsky (1961))

$$P' \left\{ \begin{array}{l} z' = \text{Min} \quad \sum_{\{i,j\} \in E} \left(c_{ij} + \frac{f_{ij}}{m_{ij}} \right) x_{ij} \quad (2.11) \\ \text{s. t.} \quad \sum_{j \in T} x_{ij} = a_i \quad i \in S \quad (2.12) \\ \quad \quad \sum_{i \in S} x_{ij} = b_j \quad j \in T \quad (2.13) \\ \quad \quad x_{ij} \geq 0 \quad \{i, j\} \in E \quad (2.14) \end{array} \right.$$

2.4 Formulazione matematica a tre indici

Recentemente è stata presentata una nuova formulazione matematica a tre indici del problema (Buson et al. (2013)). Nel presentarla, gli autori mostrano come, sebbene il rilassamento lineare di tale formulazione fornisca lo stesso bound del problema P' , sia possibile ottenere velocemente buoni lower bound aggiungendo, con metodi cutting plane, un numero pseudopolinomiale di disequazioni valide. Il lower bound finale risulta competitivo con il lower bound raggiunto al nodo radice dell'albero di ricerca dal solver MILP Cplex 12.5, quando applicato alla formulazione P .

Sia ξ_{ij}^q una variabile binaria uguale a 1 se e solo se un flusso pari a $q \in \mathbb{N}$ unità ($1 \leq q \leq m_{ij}$) è spedito dall'origine $i \in S$ alla destinazione $j \in T$ (0 altrimenti). Un costo d_{ij}^q definito come $d_{ij}^q = f_{ij} + qc_{ij}$ è associato ad ogni variabile ξ_{ij}^q . Una formulazione a tre indici del FCTP è la seguente

$$P_3 \left\{ \begin{array}{l} z_3 = \text{Min} \quad \sum_{\{i,j\} \in E} \sum_{q=1}^{m_{ij}} d_{ij}^q \xi_{ij}^q \quad (2.15) \\ s. t. \quad \sum_{j \in T} \sum_{q=1}^{m_{ij}} q \xi_{ij}^q = a_i \quad i \in S \quad (2.16) \\ \sum_{i \in S} \sum_{q=1}^{m_{ij}} q \xi_{ij}^q = b_j \quad j \in T \quad (2.17) \\ \xi_{ij}^q \in \{0,1\} \quad \{i,j\} \in E, q = 1, \dots, m_{ij} \quad (2.18) \end{array} \right.$$

La funzione obiettivo (2.15) richiede di minimizzare il costo totale associato con le variabili ξ_{ij}^q . I vincoli (2.16) impongono che il flusso uscente dall'origine $i \in S$ sia uguale a a_i . Allo stesso modo, i vincoli (2.17) impongono che il flusso entrante nell'origine $j \in T$ sia uguale a b_j . Infine, i vincoli (2.18) definiscono le variabili ξ_{ij}^q come binarie.

Denotiamo con P'_3 il rilassamento lineare del problema P_3 , ottenuto sostituendo ai vincoli (2.18) i vincoli $\xi_{ij}^q \geq 0$, e con z'_3 il costo della sua soluzione ottima. I vincoli $\xi_{ij}^q \leq 1$ possono essere ignorati poiché non incrementano il lower bound z'_3 , come mostrato di seguito.

Proposizione 2.1: Sia $\tilde{\xi}$ una soluzione ammissibile di P'_3 di costo $d(\tilde{\xi})$ tale che $\tilde{\xi}_{ij}^q \leq 1, \forall \{i,j\} \in E, q = 1, \dots, m_{ij}$. Possiamo sempre ricavare, da $\tilde{\xi}$, una soluzione ammissibile $\hat{\xi}$ di P'_3 di costo $d(\hat{\xi})$ tale che $\hat{\xi}_{ij}^q \leq 1, \forall \{i,j\} \in E, q = 1, \dots, m_{ij}$, e che $d(\hat{\xi}) \leq d(\tilde{\xi})$.

Dimostrazione. Poiché $\tilde{\xi}$ è soluzione ammissibile di P'_3 , la seguente relazione vale per ogni arco $\{i,j\} \in E$

$$m_{ij} \geq \sum_{q=1}^{m_{ij}} q \tilde{\xi}_{ij}^q = m_{ij} \sum_{q=1}^{m_{ij}} \frac{q}{m_{ij}} \tilde{\xi}_{ij}^q \quad (2.19)$$

Dalla (2.19), deduciamo che

$$\sum_{q=1}^{m_{ij}} \frac{q}{m_{ij}} \tilde{\xi}_{ij}^q \leq 1 \quad \{i,j\} \in E$$

Definiamo la soluzione $\hat{\xi}$ tramite le seguenti espressioni

$$\hat{\xi}_{ij}^q = \begin{cases} \sum_{q=1}^{m_{ij}} \frac{q}{m_{ij}} \xi_{ij}^q & \text{se } q = m_{ij} \\ 0 & \text{altrimenti} \end{cases} \quad \{i, j\} \in E, q = 1, \dots, m_{ij} \quad (2.20)$$

Possiamo osservare come la soluzione $\hat{\xi}$, definita conformemente con le espressioni (2.20), sia una soluzione ammissibile per P'_3 tale che $\hat{\xi}_{ij}^q \leq 1, \forall \{i, j\} \in E, q = 1, \dots, m_{ij}$.

Dimostriamo ora che $d(\hat{\xi}) \leq d(\tilde{\xi})$

$$\begin{aligned} d(\hat{\xi}) &= \sum_{\{i, j\} \in E} \sum_{q=1}^{m_{ij}} d_{ij}^q \hat{\xi}_{ij}^q = \sum_{\{i, j\} \in E} d_{ij}^{m_{ij}} \sum_{q=1}^{m_{ij}} \frac{q}{m_{ij}} \xi_{ij}^q \\ &= \sum_{\{i, j\} \in E} \sum_{q=1}^{m_{ij}} (f_{ij} + m_{ij} c_{ij}) \frac{q}{m_{ij}} \xi_{ij}^q = \sum_{\{i, j\} \in E} \sum_{q=1}^{m_{ij}} \left(\frac{q f_{ij}}{m_{ij}} + q c_{ij} \right) \xi_{ij}^q \\ &\leq \sum_{\{i, j\} \in E} \sum_{q=1}^{m_{ij}} (f_{ij} + q c_{ij}) \xi_{ij}^q = \sum_{\{i, j\} \in E} \sum_{q=1}^{m_{ij}} d_{ij}^q \xi_{ij}^q = d(\tilde{\xi}) \end{aligned}$$

Proposizione 2.2: $z'_3 = z'$.

Dimostrazione. Siano $\mathbf{u} \in \mathbb{R}^n$ e $\mathbf{v} \in \mathbb{R}^m$ le variabili duali associate con i vincoli (2.16) e (2.17), rispettivamente, del problema P'_3 . Il duale D'_3 del problema P'_3 è formulato come segue

$$\max \sum_{i \in S} a_i u_i + \sum_{j \in T} b_j v_j \quad (2.21)$$

$$s. t. \quad q u_i + q v_j \leq f_{ij} + q c_{ij} \quad \{i, j\} \in E, q = 1, \dots, m_{ij} \quad (2.22)$$

$$u_i \in \mathbb{R} \quad i \in S \quad (2.23)$$

$$v_j \in \mathbb{R} \quad j \in T \quad (2.24)$$

Possiamo osservare che, per un dato arco $\{i, j\} \in E$, tutti i vincoli (2.22) per cui $q < m_{ij}$ sono ridondanti. Dunque, il problema D'_3 può anche essere scritto come

$$\max \sum_{i \in S} a_i u_i + \sum_{j \in T} b_j v_j \quad (2.25)$$

$$\text{s. t. } u_i + v_j \leq \frac{f_{ij}}{m_{ij}} + c_{ij} \quad \{i, j\} \in E \quad (2.26)$$

$$u_i \in \mathbb{R} \quad i \in S \quad (2.27)$$

$$v_j \in \mathbb{R} \quad j \in T \quad (2.28)$$

Il problema (2.25) – (2.28) corrisponde anche al duale del problema (2.11) – (2.14). Dunque, $z'_3 = z'$.

2.5 Una nuova formulazione matematica intera

Una nuova formulazione intera del problema è stata recentemente proposta (Roberti et al. (2013)). Tale formulazione, che prevede un numero esponenziale di variabili binarie, si basa sull'osservazione che una soluzione ottima del FCTP corrisponda ad una soluzione base ammissibile delle equazioni (2.6) e (2.7) e che, assumendo che $a_i \in \mathbb{Z}_+, i \in S$ e $b_j \in \mathbb{Z}_+, j \in T$, una qualsiasi soluzione base ammissibile sia anche intera (Murty (1968)).

Per ogni origine $i \in S$, sia $W_i = \{\mathbf{w} \in \mathbb{Z}_+^n : \sum_{j \in T} w_j = a_i, w_j \leq m_{ij}, j \in T\}$. Siano i vettori dell'insieme W_i , $i \in S$, indicizzati in maniera tale che i vettori $\{w^1, w^2, \dots, w^{|W_1|}\}$ corrispondano a W_1 , $\{w^{|W_1|+1}, \dots, w^{|W_1|+|W_2|}\}$ a W_2 e così via. Sia \mathcal{W}_i l'insieme degli indici di tutti i vettori dell'insieme W_i (i.e., $\mathcal{W}_i = \{\ell \in \mathbb{Z} : \sum_{s=1}^{i-1} |W_s| + 1 \leq \ell \leq \sum_{s=1}^i |W_s|\}$) e sia $\mathcal{W} = \cup_{i \in S} \mathcal{W}_i$. In quanto segue ci riferiremo a qualsiasi vettore \mathbf{w}^ℓ , $\ell \in \mathcal{W}_i$, come ad un pattern di origini $i \in S$.

Sia d_ℓ il costo del pattern $\ell \in \mathcal{W}_i$, $i \in S$, definito come $d_\ell = \sum_{j \in T} c_{ij} w_j^\ell + \sum_{j \in T: w_j^\ell > 0} f_{ij}$, e sia ξ_ℓ una variabile binaria uguale a 1 se e solo se il pattern $\ell \in \mathcal{W}$ è nella soluzione (0 altrimenti). La nuova formulazione del FCTP, detta P_N , è la seguente

$$P_N \left\{ \begin{array}{l} z_N = \text{Min} \sum_{\ell \in \mathcal{W}} d_\ell \xi_\ell \quad (2.29) \\ \text{s. t. } \sum_{\ell \in \mathcal{W}} w_j^\ell \xi_\ell = b_j \quad j \in T \quad (2.30) \\ \sum_{\ell \in \mathcal{W}} \xi_\ell = 1 \quad i \in S \quad (2.31) \\ \xi_\ell \in \{0,1\} \quad \ell \in \mathcal{W} \quad (2.32) \end{array} \right.$$

La funzione obiettivo (2.29) richiede di minimizzare il costo totale dei pattern scelti. I vincoli (2.30) e (2.31) richiedono che tutte le richieste delle destinazioni siano soddisfatte e che sia scelto esattamente un pattern per ogni sorgente, rispettivamente. I vincoli (2.32) impongono che le variabili siano intere.

Da qui in avanti, denotiamo con P'_N il rilassamento lineare della formulazione P_N e con z'_N il costo della sua soluzione ottima.

Le seguenti due proposizioni affermano la relazione tra i problemi P'_N e P' .

Proposizione 2.3: *Qualsiasi soluzione $\bar{\xi}$ di P_N , di costo \bar{z}'_N , può essere trasformata in una soluzione base ammissibile di P' , (\bar{x}, \bar{y}) , di costo $\bar{z}' = \bar{z}'_N$ definendo*

$$\bar{x}_{ij} = \sum_{\ell \in \mathcal{W}_i} w_j^\ell \bar{\xi}_\ell \quad \text{e} \quad \bar{y}_{ij} = \sum_{\ell \in \mathcal{W}_i: w_j^\ell > 0} \bar{\xi}_\ell \quad (i, j) \in A \quad (2.33)$$

Dimostrazione. Innanzitutto, dimostriamo che il vettore \bar{x} definito dalle espressioni (2.33) soddisfa i vincoli (2.6) e (2.7). Come conseguenza della definizione degli insiemi \mathcal{W}_i e come conseguenza dei vincoli (2.31), per ogni origine $i \in S$ abbiamo che

$$\sum_{j \in T} \bar{x}_{ij} = \sum_{\ell \in \mathcal{W}_i} \sum_{j \in T} w_j^\ell \bar{\xi}_\ell = \sum_{\ell \in \mathcal{W}_i} a_i \bar{\xi}_\ell = a_i \sum_{\ell \in \mathcal{W}_i} \bar{\xi}_\ell = a_i \quad (2.34)$$

Inoltre, dalle equazioni (2.30) e (2.33) relative alla destinazione $j \in T$, abbiamo che

$$\sum_{i \in S} \bar{x}_{ij} = \sum_{i \in S} \sum_{\ell \in \mathcal{W}_i} w_j^\ell \bar{\xi}_\ell = \sum_{\ell \in \mathcal{W}} w_j^\ell \bar{\xi}_\ell = b_j \quad (2.35)$$

Appare ovvio dalle espressioni (2.33) che ogni $\bar{y} \geq 0$ poiché $\bar{\xi} \geq 0$. Per dimostrare che $\bar{y} \leq 1$, è sufficiente osservare che

$$\bar{y}_{ij} = \sum_{\ell \in \mathcal{W}_i: w_j^\ell > 0} \bar{\xi}_\ell \leq \sum_{\ell \in \mathcal{W}_i} \bar{\xi}_\ell = 1 \quad (i, j) \in A$$

Capitolo 2. Il Problema dei Trasporti con Costi Fissi

Per provare che (\bar{x}, \bar{y}) soddisfa i vincoli (2.8), osserviamo che per una data destinazione $j \in T$ abbiamo $\sum_{\ell \in \mathcal{W}_i} w_j^\ell \bar{\xi}_\ell = \sum_{\ell \in \mathcal{W}_i: w_j^\ell > 0} w_j^\ell \bar{\xi}_\ell$. Dunque, poiché $w_j^\ell \leq m_{ij}$, $\ell \in \mathcal{W}_i, j \in T$, dalle espressioni (2.33) otteniamo

$$\bar{x}_{ij} = \sum_{\ell \in \mathcal{W}_i} w_j^\ell \bar{\xi}_\ell = \sum_{\ell \in \mathcal{W}_i: w_j^\ell > 0} w_j^\ell \bar{\xi}_\ell \leq m_{ij} \sum_{\ell \in \mathcal{W}_i: w_j^\ell > 0} \bar{\xi}_\ell = m_{ij} \bar{y}_{ij} \quad (i, j) \in A \quad (2.36)$$

Infine, dimostriamo che $\bar{z}' = \bar{z}'_N$. Utilizzando la definizione dei costi d_ℓ , il costo \bar{z}'_N della soluzione $\bar{\xi}$ può essere scritto come

$$\bar{z}'_N = \sum_{\ell \in \mathcal{W}} d_\ell \bar{\xi}_\ell = \sum_{i \in S} \sum_{\ell \in \mathcal{W}_i} d_\ell \bar{\xi}_\ell = \sum_{i \in S} \sum_{\ell \in \mathcal{W}_i} \sum_{j \in T} c_{ij} w_j^\ell \bar{\xi}_\ell + \sum_{i \in S} \sum_{\ell \in \mathcal{W}_i} \sum_{j \in T: w_j^\ell > 0} f_{ij} \bar{\xi}_\ell \quad (2.37)$$

Dalle equazioni (2.33), otteniamo

$$\sum_{i \in S} \sum_{\ell \in \mathcal{W}_i} \sum_{j \in T} c_{ij} w_j^\ell \bar{\xi}_\ell = \sum_{i \in S} \sum_{j \in T} c_{ij} \sum_{\ell \in \mathcal{W}_i} w_j^\ell \bar{\xi}_\ell = \sum_{i \in S} \sum_{j \in T} c_{ij} \bar{x}_{ij} \quad (2.38)$$

e

$$\sum_{i \in S} \sum_{\ell \in \mathcal{W}_i} \sum_{j \in T: w_j^\ell > 0} f_{ij} \bar{\xi}_\ell = \sum_{i \in S} \sum_{j \in T} f_{ij} \sum_{\ell \in \mathcal{W}_i: w_j^\ell > 0} \bar{\xi}_\ell = \sum_{i \in S} \sum_{j \in T} f_{ij} \bar{y}_{ij} \quad (2.39)$$

Infine, dalle espressioni (2.37) – (2.38) abbiamo che

$$\bar{z}'_N = \sum_{i \in S} \sum_{j \in T} c_{ij} \bar{x}_{ij} + \sum_{i \in S} \sum_{j \in T} f_{ij} \bar{y}_{ij} = \bar{z}'$$

Proposizione 2.4: Siano z' e z'_N i costi di una soluzione ottima di P' e P'_N , rispettivamente. Allora, vale la disuguaglianza $z' \leq z'_N$ e tale disuguaglianza può essere verificata in senso stretto.

Dimostrazione. Sia $\bar{\xi}$ una soluzione ottima di P'_N e sia (\bar{x}, \bar{y}) la soluzione di P' di costo z' derivata da $\bar{\xi}$ in accordo con le espressioni (2.33). Notiamo che se avessimo $0 < w_j^\ell < m_{ij}$ per un qualche arco $(i, j) \in A$ e qualche $\ell \in \mathcal{W}_i$ tale che $\bar{\xi}_\ell > 0$, allora, grazie alle equazioni (2.33), avremmo

$$\bar{x}_{ij} = \sum_{\ell \in \mathcal{W}_i} w_j^\ell \bar{\xi}_\ell = \sum_{\ell \in \mathcal{W}_i: w_j^\ell > 0} w_j^\ell \bar{\xi}_\ell < m_{ij} \sum_{\ell \in \mathcal{W}_i: w_j^\ell > 0} \bar{\xi}_\ell = m_{ij} \bar{y}_{ij}.$$

Dunque, $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ non può essere soluzione ottima di P' in quanto possiamo derivare da $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ una soluzione ammissibile $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ di P' di costo $\hat{z}' < z'$ impostando $\hat{\mathbf{x}} = \bar{\mathbf{x}}$ e $\hat{y}_{ij} = \hat{x}_{ij}/m_{ij}, (i, j) \in A$. La soluzione $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ non è necessariamente ottima, ma satura comunque i vincoli (2.8) come una qualsiasi soluzione ottima di P' farebbe. In conclusione, abbiamo che $z'_N > \hat{z}' \geq z'$.

3. Metodi di risoluzione esatti

Il FCTP è stato formulato per la prima volta da Hirsch e Dantzig (1954), i quali osservarono come la soluzione ottima del problema si raggiunga ad un punto estremo della regione ammissibile delle equazioni (2.6) – (2.7) e delle disequazioni (2.9). I primi metodi di risoluzione esatti sono stati presentati, tra gli altri, da Murty (1968), Gray (1971) e Kennington e Unger (1976).

Barr, Glover e Klingman hanno descritto un algoritmo branch-and-bound per le reti di trasporto sparse (Barr et al. (1980)).

Cabot e Erenguc (1986), hanno presentato un algoritmo branch-and-bound basato su penalties condizionali per valutare il costo addizionale derivato dall'impostare a 0 o 1 una delle variabili binarie frazionali nella soluzione ottima P' .

Palekar, Karwan e Zionts hanno introdotto nuovi vincoli basati su penalties più forti delle precedenti proposte da Cabot e Erenguc (Palekar et al. (1990)). Lamar e Wallace (1997), hanno dimostrato che le penalties modificate, sviluppate da Palekar, non sono valide e hanno indicato le modifiche necessarie per renderle nuovamente valide.

In Göthe-Lundgren e Larsson (1994), viene descritto un algoritmo esatto per il PFTCP basato su una nuova formulazione del problema.

3.1 Il metodo di Barr

Nel 1980, Barr, Glover e Klingman hanno presentato un nuovo algoritmo branch-and-bound, il quale utilizza la regola LIFO per la costruzione dell'albero di enumerazione, principalmente per due motivi:

- facile utilizzo,
- semplificazione la riottimizzazione dei problemi candidati dalla precedente soluzione.

Questo metodo utilizza il rilassamento lineare standard (Balinsky (1961)) secondo il quale il costo corrente relativo all'arco (i, j) ad un dato passo del processo di branch-and-bound è uguale a

Capitolo 3. Metodi di risoluzione esatti

- $d_{ij} = c_{ij} + \frac{f_{ij}}{m_{ij}}$, se x_{ij} è libera,
- M (costante positiva grande), se $x_{ij} = 0$,
- c_{ij} , se $x_{ij} > 0$.

Nell'ultimo caso, la costante f_{ij} va aggiunta al valore corrente della funzione obiettivo e x_{ij} può dunque ricevere un qualsiasi valore non negativo (anche 0).

3.1.1 Test di pre-processing

Vengono utilizzati tre test standard un dato sottoproblema:

- inammissibilità,
- integralità della soluzione,
- il valore della funzione obiettivo più le penalties eccede il valore della soluzione ottima in carica.

In aggiunta, come conseguenza di quanto dimostrato in Hirsch e Dantzig (1968), alle variabili x_{ij} vincolate a positivo viene richiesto che siano in base. Perciò, ogni volta che una variabile viene vincolata a positivo, una procedura verifica se si creino cicli relativi agli altri archi vincolati, o meno. Se ciò avviene, tale “mossa” è identificata come inammissibile. La virtù primaria di tale test è che può essere eseguito rapidamente, anche su reti molto grandi, con un impiego molto basso di risorse di memoria.

L'algoritmo, inoltre, utilizza le “up” e “down” penalties come formulate da Driebeek (1966), e migliorate da Tomlin (1971). Poiché il calcolo di tali valori risulta dispendioso, viene utilizzato un approccio ad etichettatura “a predecessore generalizzato”, che permette di determinare tutte le penalties con poco sforzo in più di quello richiesto per calcolarne una.

3.1.2 Local star

La complessità computazionale del processo di risoluzione diminuisce quando bound più stretti sono imposti sul valore della soluzione ottima. Pertanto, viene ricercato un upper bound più stretto sull'ottimo, tramite l'utilizzo iniziale di una procedura di pivoting proposta da Walker (1976), per isolare una “local star”.

3.1.3 Strategie di separazione e branching

Due regole per direzionare la ricerca tra i sotto problemi risultano cruciali per l'effettiva performance di un algoritmo: la regola per la selezione della variabile di separazione e la regola di decisione di branching (Geoffrion e Marsten (1972)).

Delle due strategie generali testate, la prima utilizza le up e down penalties per tutti gli archi a costo fisso in base tra il loro upper e lower bound. I criteri di separazione testati includono la scelta della variabile con la maggiore penalty individuale (SR-1) e quella con la maggior differenza assoluta tra le penalties (SR-2). Una terza regola di separazione considera la minor penalty per ogni variabile e sceglie quella variabile corrispondente alla maggiore di quelle penalties (SR-3). Le regole di branching testate forzano sempre la variabile di separazione al suo upper bound (BR-1), al suo lower bound (BR-2) e nella direzione della sua penalty minima o massima (BR-3 e BR-4).

La seconda strategia non richiede il calcolo delle penalties ma, al contrario, utilizza la funzione $g_{ij} = f_{ij} \left(1 - \left(\frac{x_{ij}}{m_{ij}} \right) \right)$, la deviazione del rilassamento lineare, d_{ij} , dalla funzione obiettivo reale, f_{ij} , al corrente livello del flusso.

La variabile scelta per la separazione è quella con la massima o minima deviazione (SR-5 e SR-6) e viene data precedenza a quelle variabili che si trovano a più di αm_{ij} unità da entrambi i bound (SR-7), ove $0 < \alpha < 1$. Il branching viene eseguito tramite le regole BR-1 e BR-2 sopra descritte.

3.2 Penalties di Driebeek e di Cabot e Erenguc

Consideriamo il rilassamento lineare P' , come visto nel paragrafo 2.3.1. La soluzione a tale problema, in genere, avrà alcune delle variabili y_{ij} con valore frazionario, cioè $0 < y_{ij} < 1$. Ciò nonostante, in qualsiasi soluzione al problema originale P ogni y_{ij} deve avere un valore di 0 o 1. Ciò comporta un costo aggiuntivo per ottenere una soluzione intera. Al fine rendere più stringente il lower bound sull'ottimo, si può calcolare un lower bound su tale costo aggiuntivo, detto penalty condizionale.

Allora, se forziamo $y_{kl} = 1$, otteniamo le seguenti up penalties per la variabile y_{kl} :

- *penalties di Driebeek*

$$UP_{kl}^D = \min \left\{ c^*, \frac{f_{kl}}{m_{kl}} \right\} (m_{kl} - \bar{x}_{kl}), \text{ ove}$$

$$c^* = \min_{x_{ij} \in N} \left\{ \left(\bar{c}_{ij} + \frac{f_{ij}}{m_{ij}} \right) \right\}, \text{ se } a_{ij}^{kl} = -1.$$

- *penalties di Cabot-Erenguc*

$$UP_{kl}^C = \min \{ UP_{kl}^1, UP_{kl}^2, UP_{kl}^3 \}, \text{ ove}$$

$$UP_{kl}^1 = \min_{x_{ij} \in N} \{ \bar{c}_{ij} (m_{kl} - \bar{x}_{kl}) + f_{ij} \}, \text{ se } a_{ij}^{kl} = -1 \text{ e } \bar{c}_{ij} \geq 0,$$

$$UP_{kl}^2 = \min_{x_{ij} \in N} \{ \bar{c}_{ij} m_{ij} + f_{ij} \}, \text{ se } a_{ij}^{kl} = -1 \text{ e } \bar{c}_{ij} < 0,$$

$$UP_{kl}^3 = \frac{f_{kl}}{m_{kl}} (m_{kl} - \bar{x}_{kl}).$$

Similmente, se forziamo $y_{kl} = 0$, otteniamo le seguenti down penalties:

- *penalties di Driebeek*

$$DN_{kl}^D = \min_{x_{ij} \in N} \left\{ \left(\bar{c}_{ij} + \frac{f_{ij}}{m_{ij}} \right) \bar{x}_{kl} \right\}, \text{ se } a_{ij}^{kl} = 1.$$

- *penalties di Cabot-Erenguc*

$$DN_{kl}^C = \min \{ DN_{kl}^1, DN_{kl}^2 \}, \text{ ove}$$

$$DN_{kl}^1 = \min_{x_{ij} \in N} \{ \bar{c}_{ij} \bar{x}_{kl} + f_{ij} \}, \text{ se } a_{ij}^{kl} = 1 \text{ e } \bar{c}_{ij} \geq 0,$$

$$DN_{kl}^2 = \min_{x_{ij} \in N} \{ \bar{c}_{ij} m_{ij} + f_{ij} \}, \text{ se } a_{ij}^{kl} = 1 \text{ e } \bar{c}_{ij} < 0.$$

3.3 Penalties di Palekar, Karwan e Zionts

Palekar, Karwan e Zionts fanno notare, in un paper del 1990, come, considerando le up penalties, la variabile entrante x_{ij} assuma valore $m_{kl} - \bar{x}_{kl}$ e da un lato, nel caso in cui $m_{ij} > (m_{kl} - \bar{x}_{kl})$, solo una parte del costo fisso venga

pagata, dall'altro, ossia quando $m_{ij} < (m_{kl} - \bar{x}_{kl})$, nessun aggiustamento risulti più possibile in quanto si è già pagato più del necessario (Palekar et al. (1990)).

Essi proposero, dunque, le seguenti up e down penalties, rispettivamente:

$$UP_{kl}^M = \min \left\{ A, \frac{f_{kl}}{m_{kl}} (m_{kl} - x_{kl}) \right\}, \text{ ove}$$

$$A = \min_{x_{ij} \in N} \left\{ \left(c_{ij} + \frac{f_{ij}}{m_{ij}} \right) (m_{kl} - \bar{x}_{kl}) + \max \left\{ 0, (m_{ij} - m_{kl} + \bar{x}_{kl}) \frac{f_{ij}}{m_{ij}} \right\} \right\}, \text{ se } a_{ij}^{kl} = -1;$$

$$DN_{kl}^M = \min_{x_{ij} \in N} \left\{ \left(c_{ij} + \frac{f_{ij}}{m_{ij}} \right) \bar{x}_{kl} + \max \left\{ 0, (m_{ij} - \bar{x}_{kl}) \frac{f_{ij}}{m_{ij}} \right\} \right\}, \text{ se } a_{ij}^{kl} = -1.$$

Essi dimostrarono, inoltre, come tali penalties fossero valide e più forti di quelle precedentemente proposte sia da Driebeek (1966), che da Cabot e Erenguc (1986).

3.4 Penalties di Lamar e Wallace

Nel 1997, Lamar e Wallace fanno notare che le penalties proposte da Palekar et al. (1990), non sono effettivamente valide in quanto possono violare la condizione

$$z'_q + PEN \leq z_q$$

ossia, il valore della soluzione ottima del rilassamento lineare del sottoproblema q con aggiunte le M-penalties diventa maggiore del valore della soluzione ottima del sottoproblema q stesso, quando $\bar{c}_{ij} < 0$ (Lamar e Wallace (1997)).

Lamar e Wallace hanno proposto una versione rivisitata delle M-penalties che soddisfa la suddetta condizione. Le penalties risultanti sono le seguenti:

$$UP_{ijkl} \begin{cases} f_{ij} + \bar{c}_{ij}(m_{kl} - \bar{x}_{kl}) & \text{se } m_{ij} \geq (m_{kl} - \bar{x}_{kl}) \text{ e } \bar{c}_{ij} \geq 0 \\ f_{ij} + \bar{c}_{ij}m_{ij} & \text{se } m_{ij} \geq (m_{kl} - \bar{x}_{kl}) \text{ e } \bar{c}_{ij} < 0 \\ \left(\bar{c}_{ij} + \frac{f_{ij}}{m_{ij}} \right) (m_{kl} - \bar{x}_{kl}) & \text{se } m_{ij} < (m_{kl} - \bar{x}_{kl}) \end{cases}$$

$$DN_{ijkl} \begin{cases} f_{ij} + \bar{c}_{ij}\bar{x}_{kl} & \text{se } m_{ij} \geq \bar{x}_{kl} \text{ e } \bar{c}_{ij} \geq 0 \\ f_{ij} + \bar{c}_{ij}m_{ij} & \text{se } m_{ij} \geq \bar{x}_{kl} \text{ e } \bar{c}_{ij} < 0 \\ \left(\bar{c}_{ij} + \frac{f_{ij}}{m_{ij}} \right) \bar{x}_{kl} & \text{se } m_{ij} < \bar{x}_{kl} \end{cases}$$

3.5 Il metodo di Agarwal e Aneja

Recentemente, Agarwal e Aneja hanno studiato la struttura del poliedro proiettato dalla formulazione classica nello spazio delle variabili y_{ij} (Agarwal e Aneja (2012)). Hanno in seguito sviluppato diverse classi di disequazioni valide e derivato condizioni sotto le quali tali disequazioni sono facet defining. Il loro metodo esatto risolve istanze generate in maniera random sia di FCTP che di PFCTP con un massimo di 15 origini e 15 destinazioni.

3.5.1 La formulazione di Agarwal e Aneja

Sia \mathcal{H} l'insieme di tutte le soluzioni ammissibili (x, y) al problema P e \mathcal{H}_C la convoluzione di \mathcal{H} . Allo stesso modo si definiscano $\mathcal{F} = \{y \in \mathbb{B}^{nm} : (x, y) \in \mathcal{H}\}$ e $\mathcal{F}_C = \text{Conv}(\mathcal{F})$. In altre parole, sia \mathcal{F}_C è la proiezione di \mathcal{H}_C nello spazio delle variabili y_{ij} .

Sebbene lo studio della struttura poliedrica sia chiaramente utile alla risoluzione del PFCTP, in cui si ha a che fare con i soli costi fissi, esso risulta di eguale utilità nella risoluzione del FCTP poiché, nonostante non sia possibile risolverlo direttamente lavorando col poliedro \mathcal{F}_C , in quanto i costi c_{ij} non sono nulli, le facets di \mathcal{F}_C risultano disequazioni forti valide per \mathcal{H}_C .

Una formulazione del FCTP, già proposta in Aneja (1974), basata su un teorema dovuto a Gale (Ford e Fulkerson (1962)) è la seguente.

Sia dato una rete direzionata e capacitata, $G = (V, E)$, i cui nodi sono divisi in destinazioni, nodi intermedi e origini. Interpretando le disponibilità alle origini come richieste negative, il teorema di Gale afferma che il grafo ha un flusso ammissibile se e solo se la richiesta attraverso un qualsiasi sottoinsieme $\bar{X} \subseteq V$ non supera la capacità totale degli archi che entrano in \bar{X} .

Dati $P \subset S$ e $Q \subset T$, siano $A(P) = \sum_{s_i \in P} a_i$ e $B(Q) = \sum_{t_j \in Q} b_j$. Siano poi $\bar{P} = S \setminus P$ e $\bar{Q} = T \setminus Q$. Utilizzando il teorema di Gale, scegliendo $\bar{X} = (\bar{P} \cup Q)$, il problema può essere formulato utilizzando solo le variabili binarie y_{ij}

$$P_G \begin{cases} \text{Min} & \sum_{i=1}^n \sum_{j=1}^m f_{ij} y_{ij} \\ \text{s.t.} & \sum_{s_i \in P} \sum_{t_j \in Q} m_{ij} y_{ij} \geq B(Q) - A(\bar{P}) \quad \forall P \subset S \quad \forall Q \subset T \\ & y_{ij} \in \{0,1\} \quad \forall i = 1, \dots, n \quad \forall j = 1, \dots, m \end{cases}$$

Poiché il teorema di Gale non richiede l'integralità delle variabili y_{ij} , si intuisce chiaramente come il rilassamento lineare di P e P_G coincidano. Chiaramente, \mathcal{F}_C è l'involuppo convesso di \mathcal{F} , l'insieme di tutte le soluzioni ammissibili di P_G .

Definiamo ora $\alpha(P, Q) = B(Q) - A(\bar{P}) = B(Q) + A(P) - D$, ove $D = \sum_{i=1}^n a_i = \sum_{j=1}^m b_j$. Se $\alpha(P, Q) \leq 0$, allora il corrispondente vincolo in P_G risulta ridondante. Dobbiamo considerare, dunque, solo i vincoli per quei sottoinsiemi P e Q tali che $\alpha(P, Q) > 0$. Sia Ω l'insieme di tutti i sottoinsiemi (P, Q) siffatti, detti *cutset* o *insiemi di taglio*, e definiamo

$$E_{PQ} = \{(i, j): s_i \in P, t_j \in Q\},$$

$$\underline{m}(P, Q) = \min\{m_{ij}: (i, j) \in E_{PQ}\},$$

$$\bar{m}(P, Q) = \max\{m_{ij}: (i, j) \in E_{PQ}\}.$$

Sia $(P, Q) \in \Omega$ un *cutset minimo* se $\alpha(P, Q) \leq \underline{m}(P, Q)$ e Ω_1 l'insieme di tutti i cutset minimi (P, Q) così definiti. Si osservi che, se un cutset (P, Q) non è minimo, allora esiste un cutset $(P', Q') \in \Omega_1$, diverso da (P, Q) , tale che $P' \subseteq P$ e $Q' \subseteq Q$. Consideriamo ora un cutset minimo $(P, Q) \in \Omega_1$. Il vincolo corrispondente a tale cutset in P_G può essere riscritto come

$$\sum_{(i,j) \in E_{PQ}} \left(\frac{m_{ij}}{\underline{m}(P, Q)} \right) y_{ij} \geq \frac{\alpha(P, Q)}{\underline{m}(P, Q)} = \varepsilon(P, Q)$$

Notiamo che ogni y_{ij} è binaria, $\frac{m_{ij}}{\underline{m}(P, Q)} \geq 1$ per ogni $(i, j) \in E_{PQ}$ e $0 < \varepsilon(P, Q) \leq$

1. A meno che $\bar{m}(P, Q) = \underline{m}(P, Q) = \alpha(P, Q)$, applicando una procedura di arrotondamento degli interi, otteniamo una disequazione valida più forte:

$$\sum_{(i,j) \in E_{PQ}} y_{ij} \geq 1 \quad (3.2)$$

Di conseguenza, otteniamo una formulazione più forte di P_G :

$$P_A \left\{ \begin{array}{l} z_A = \text{Min} \quad \sum_{i=1}^n \sum_{j=1}^m f_{ij} y_{ij} \quad (3.3) \\ s. t. \quad \sum_{(i,j) \in E_{PQ}} y_{ij} \geq 1 \quad \forall (P, Q) \in \Omega_1 \quad (3.4) \\ \sum_{(i,j) \in E_{PQ}} m_{ij} y_{ij} \geq \alpha(P, Q) \quad \forall (P, Q) \in \Omega \setminus \Omega_1 \quad (3.5) \\ y_{ij} \in \{0,1\} \quad \forall i = 1, \dots, n \quad \forall j = 1, \dots, m \quad (3.6) \end{array} \right.$$

Nonostante P_A sia una formulazione più forte di P_G , Agarwal e Aneja (2012), non propongono di sostituirla alla formulazione originale P , per due ragioni. In primis, tale formulazione implica un numero esponenziale di variabili binarie, il che la rende poco pratica; inoltre, P_G , non facendo uso delle variabili x_{ij} , non può essere utilizzata per risolvere problemi con costi dipendenti dal flusso (i.e. c_{ij}). Il loro approccio, piuttosto, è quello di utilizzare le facets e le disequazioni valide derivate da tale formulazione e aggiungerle al problema originale P al fine di risolvere quest'ultimo.

3.5.2 Facets, euristici e metodo

Nel loro paper del 2012, Agarwal e Aneja propongono tre tipi di disequazioni facet defining valide: facets a copertura d'insieme (SC), facets del secondo ordine e facets del secondo ordine estese.

3.5.2.1 Facets a copertura d'insieme

Agarwal e Aneja (2012), dimostrano come le disequazioni (3.4) della formulazione P_A siano facet defining o, sotto due condizioni blande, possano essere elevate a facet defining.

Il Teorema 2 contenuto nel loro paper asserisce, infatti, che, dato $(P, Q) \in \Omega_1$, supponendo che

- le origini in P siano etichettate s_1, \dots, s_p , con $p = |P|$,

Capitolo 3. Metodi di risoluzione esatti

- le destinazioni in Q siano etichettate t_1, \dots, t_q , con $q = |Q|$,
- $a_1 \geq a_2 \geq \dots \geq a_p$,
- $b_1 \geq b_2 \geq \dots \geq b_q$,
- $a_{p+1} \geq a_{p+2} \geq \dots \geq a_n$,
- $b_{q+1} \geq b_{q+2} \geq \dots \geq b_m$ e
- $a_1 \geq b_1$, scambiando le disponibilità con le richieste se necessario,

$$\sum_{(i,j) \in E_{PQ}} y_{ij} \geq 1 \quad (3.7)$$

sia facet defining se e solo se $a_1 + b_{q+1} - B(\bar{Q}) \leq b_1$, ossia se e solo se la disponibilità dell'origine $s_1 \in P$ (i.e. quella con disponibilità massima) più la richiesta della destinazione $t_{q+1} \in \bar{Q}$ (i.e. quella con richiesta massima) meno il flusso totale richiesto in \bar{Q} è minore o uguale alla richiesta della prima destinazione $t_1 \in Q$.

Nel caso, in cui la condizione del Teorema 2 non sia soddisfatta, definendo

- $\delta = a_1 - (B(\bar{Q}) - b_{q+1})$, la quantità minima che deve essere spedita da s_1 a Q se $y_{1,q+1} = 0$,
- $\zeta = \min\{r: \sum_{\ell=1}^r b_\ell \geq \delta\}$, il numero minimo di cammini aperti necessari a spedire δ unità da s_1 a Q ,

ed elevando la precedente disequazione a

$$(\zeta - 1)y_{1,q+1} + \sum_{(i,j) \in E_{PQ}} y_{ij} \geq \zeta \quad (3.8)$$

se $a_1 + b_{q+2} - B(\bar{Q}) \leq b_1$ e $a_2 + b_{q+1} - B(\bar{Q}) \leq b_1$, essa è facet defining. È triviale dimostrare la validità di tale disequazione poiché, nel caso in cui $y_{1,q+1} = 0$, la (3.8) si riduce alla (3.7). Nel caso in cui, al contrario, $y_{1,q+1} = 1$, la (3.8) risulta valida se e solo se abbiamo almeno ζ cammini aperti da E_{PQ} , il che è ovviamente vero come conseguenza della definizione di ζ .

3.5.2.2 Facets del secondo ordine e facets estese

Agarwal e Aneja (2012), definiscono altri due tipi di disequazioni valide che siano anche facet defining per il problema P , le facets del secondo ordine e le facets del secondo ordine estese.

La dimostrazione del fatto che siano facet defining va ben oltre lo scopo di questa tesi, per cui vengono di seguito riportate le disequazioni come espresse nel paper originale.

Consideriamo la disequazione valida (3.5) della formulazione P_A :

$$\sum_{(i,j) \in E_{PQ}} m_{ij} y_{ij} \geq \alpha(P, Q)$$

Supponiamo $\alpha(P, Q) > \underline{m}(P, Q)$. In questo caso

$$\sum_{(i,j) \in E_{PQ}} y_{ij} \geq 2 \quad (3.9)$$

è una disequazione valida per \mathcal{F}_C .

Allo stesso modo, siano

$$E_1 = \{(i, j) \in E_{PQ} : m_{ij} < \alpha(P, Q)\}, \text{ e}$$

$$E_2 = \{(i, j) \in E_{PQ} : m_{ij} \geq \alpha(P, Q)\}.$$

Allora, banalmente

$$\sum_{(i,j) \in E_1} y_{ij} + 2 \sum_{(i,j) \in E_2} y_{ij} \geq 2 \quad (3.10)$$

è una disequazione valida per \mathcal{F}_C .

3.5.2.3 Separazione delle disequazioni

Per la separazione delle disequazioni valide, Agarwal e Aneja (2012), utilizzano tre diversi euristici: un *Local Search Heuristic* (H1) ed un *Knapsack Heuristic* (H2) per le facets a copertura d'insieme e un *Local Search Heuristic* (H3) per le facets del secondo ordine e le facets del secondo ordine estese. Il motivo per cui vengono

impiegati due diversi euristici per la separazione delle facets a copertura d'insieme, spiegano gli autori, è che, in molti casi, il secondo euristico riesce a separare facets che il primo non separa.

Il metodo esatto può essere riassunto come segue:

- (1) Dato il problema in formulazione P , si risolva il rilassamento lineare standard P' . Sia \bar{y}_{ij} la soluzione ottima.
- (2) Utilizzare il *Local Search Heuristic* (H1), iterandolo R_1 volte per trovare la facet più violata. Se si trova una tale facet, aggiungerla al problema, risolverlo e ripetere questo passo fin quando possibile. Altrimenti, proseguire allo step (3).
- (3) Utilizzare il *Knapsack Heuristic* (H2), iterandolo R_2 volte per trovare la facet più violata. Se si trova una tale facet, aggiungerla al problema, risolverlo e tornare allo step (2). Altrimenti, proseguire allo step (4).
- (4) Utilizzare il *Local Search Heuristic* (H3) per le facet del secondo ordine, iterandolo R_3 volte per trovare la facet più violata. Se si trova una tale facet, aggiungerla al problema, risolverlo e tornare allo step (2). Altrimenti, proseguire per lo step (5).
- (5) Il calcolo del nodo radice è ora completo. Utilizzare Cplex per la risoluzione del problema tramite branch-and-cut. Durante la procedura branch-and-cut, solo l'euristico (H1) è utilizzato per generare disequazioni a copertura d'insieme, con un valore piccolo di R_1 e fino a una profondità limitata.

3.6 Il metodo esatto di Roberti, Bartolini e Mingozzi (2013)

L'algoritmo che andiamo ora a presentare è stato sviluppato da Roberti, Bartolini e Mingozzi nel 2013 (Roberti et al. (2013)) ed è tuttora in revisione pre-pubblicazione presso *Management Science*. Tale esatto risulta essere, sulla base delle informazioni in nostro possesso, lo stato dell'arte degli esatti per la risoluzione del FCTP, capace di risolvere istanze di FCTP e PFCTP con fino a 70 origini e 70 destinazioni, con prestazioni migliori sia dell'esatto di Agarwal e Aneja (2012), presentato nel paragrafo 3.2 sia dell'ultima versione del solver MILP Cplex.

Questo esatto si basa sulla nuova formulazione matematica, P_N , proposta dagli autori stessi e presentata nel paragrafo 2.5. La necessità di tale nuova formulazione è data da due ragioni:

- il suo rilassamento lineare produce un lower bound migliore di quello ottenibile dal rilassamento lineare della formulazione classica;
- può essere migliorata con qualsiasi disequazione valida applicabile a P ;
- ad essa possono essere aggiunte disequazioni valide che non possono essere aggiunte alla formulazione classica, P .

3.6.1 Disequazioni valide

Il rilassamento lineare P'_N può essere risolto solamente da metodi a generazione di colonne, poiché il numero delle variabili può divenire troppo grande per istanze del FCTP non triviali. Il lower bound z'_N può essere migliorato ulteriormente aggiungendo disequazioni valide, le cui variabili duali possono essere considerate nella risoluzione del problema di pricing senza intaccarne la robustezza.

Nel seguito, descriviamo disequazioni valide, alcune delle quali introdotte per la prima volta da Roberti et al. (2013), che possono essere aggiunte al problema P'_N per migliorare il lower bound z'_N .

3.6.1.1 Disequazioni Set Covering (SC)

Le disequazioni a copertura d'insieme sono state proposte per la prima volta da Göthe-Lundgren e Larsson (1994), ed è stato dimostrato da Agarwal e Aneja (2012), come siano, sotto certe circostanze, facet defining.

Nel corso di questa tesi, sono già state definite nel paragrafo 3.2.2.1 e non verranno pertanto ulteriormente descritte. Si vuole, invece, descrivere come Roberti et al. (2013), le abbiano applicate alla nuova formulazione P'_N .

Siano $K \subseteq S$, $L \subseteq T$, $\mathcal{S} = \{(K, L): K \subseteq S, L \subseteq T \mid \sum_{i \in K} a_i < \sum_{j \in L} b_j\}$. Definiamo per ogni pattern $\ell \in \mathcal{W}_i$, $i \in S \setminus K$, il coefficiente $\tau_\ell(K, L)$ che indica il numero di destinazioni dell'insieme L servite dal pattern ℓ , ossia

$$\tau_\ell(K, L) = |\{j \in L: w_j^\ell > 0\}| \quad \ell \in \mathcal{W}_i \quad i \in S \setminus K \quad (3.11)$$

Poiché una qualsiasi soluzione di P'_N può essere trasformata in una valida soluzione di P' , come mostrato nel paragrafo 2.5, **Proposizione 2.3**, allora le disequazioni set-covering possono essere introdotte nel problema P'_N come

$$(SC) \quad \sum_{i \in S \setminus K} \sum_{\ell \in W_i} \tau_\ell(K, L) \xi_\ell \geq 1 \quad (K, L) \in \mathcal{S} \quad (3.12)$$

Gli autori riportano, come metodo per la separazione di tali disequazioni, la risoluzione di un problema binario, già presentato da Agarwal e Aneja (2012), mediante un qualsiasi solver MILP (i.e. Cplex).

3.6.1.2 Disequazioni Upper Bound Cover estese (EGUBC)

Le disequazioni *Upper Bound Cover* generalizzate (GUBC) sono un tipo di disequazioni ben note in letteratura (Wolsey (1998), Gu et al. (1998), Gu et al. (1999)) per problemi MILP con variabili binarie soggette sia a vincoli *knapsack* sia a vincoli *upper bound* generalizzati. Roberti et al. (2013), presentano una variante delle GUBC le cui variabili duali possono essere gestite quando si esegue il pricing delle colonne per risolvere il rilassamento lineare P'_N .

Si consideri l'insieme \mathcal{P} di tutte le soluzioni ammissibili di un vincolo *knapsack* (equazione) ed un insieme di n vincoli *GUB* non-sovrapposti che coinvolgano k variabili binarie, i.e. $\mathcal{P} = \{\vartheta \in \{0,1\}^k : \sum_{j \in K} \omega_j \vartheta_j = \beta_j; \sum_{j \in K_r} \vartheta_j \leq 1, r = 1, \dots, n; K = \{1, \dots, k\}; K_r \cap K_s = \emptyset, r, s = 1, \dots, n, r \neq s; \bigcup_{r=1}^n K_r = K\}$, ove $\omega_j \in \mathbb{Z}, 0 < \omega_j \leq \beta, j = 1, \dots, k$ e $\beta \in \mathbb{Z}_+$.

Una GUBC per l'insieme \mathcal{P} è un sottoinsieme $C \subseteq K$ di variabili di peso totale maggiore di β (i.e. $\sum_{j \in C} \omega_j > \beta$) e tale che non esista una coppia di variabili di C appartenenti allo stesso insieme K_r . Una *GUBC* è minima se $\sum_{j \in C \setminus \{s\}} \omega_j \leq \beta, \forall s \in C$. Sia \mathcal{C} l'insieme di tutte *GUBC* minime dell'insieme \mathcal{P} , allora si definiscono disequazioni *GUBC* le seguenti

$$(GUBC) \quad \sum_{j \in C} \vartheta_j \leq |C| - 1 \quad C \in \mathcal{C} \quad (3.13)$$

Sia ora $\rho(C) = \max_{j \in C} \{\omega_j\}$ per ogni $C \in \mathcal{C}$ e $\gamma(C) = \{\gamma_1(C), \gamma_2(C), \dots, \gamma_n(C)\}$ il vettore definito come

3.6 Il metodo esatto di Roberti, Bartolini e Mingozzi (2013)

$$\gamma_r(C) = \begin{cases} \omega_j & \text{se } \exists j \in C \cap K_r \\ \rho(C) & \text{altrimenti} \end{cases}$$

Allora, le seguenti disequazioni, dette disequazioni *Upper Bound Cover* generalizzate estese (EGUBC), sono valide per l'insieme \mathcal{P} e sono più stringenti (*lifting*) delle disequazioni GUBC (3.13)

$$\sum_{r=1}^n \sum_{j \in K_r: \omega_j \geq \gamma_r(C)} \vartheta_j \leq |C| - 1 \quad C \in \mathcal{C} \quad (3.14)$$

Definendo

$$\gamma_i(C) = \begin{cases} w_j^\ell & \text{se } \exists \ell \in C \cap \mathcal{W}_i \\ \rho(C) & \text{altrimenti} \end{cases} \quad i \in S$$

le (3.14) possono essere aggiunte alla formulazione P'_N come

$$(EGUBC) \quad \sum_{i \in S} \sum_{\ell \in \mathcal{W}_i: w_j^\ell \geq \gamma_i(C)} \xi_\ell \leq |C| - 1 \quad C \in \mathcal{C}_j \quad j \in T \quad (3.15)$$

Anche questo tipo di disequazioni sono separabili risolvendo un problema binario tramite una ricorsione di programmazione dinamica.

3.6.1.3 Disequazioni di ammissibilità (FSB)

Le disequazioni di ammissibilità (FSB) introdotte da Roberti et al. (2013), come apparirà di seguito sono applicabili non solo al FCTP ma anche ad un qualsiasi problema di programmazione a numeri interi.

Si consideri un'equazione generica nella forma

$$\omega^T \vartheta = \beta \quad (3.16)$$

ove $\vartheta \in \{0,1\}^k$, $\omega \in \mathbb{Z}_+^k$, $\beta \in \mathbb{Z}_+$ e $1 \leq \omega_j \leq \beta, j = 1, \dots, k$. Sia Θ l'insieme di tutte le soluzioni ammissibili dell'equazione (3.16). È facilmente osservabile che, per ogni soluzione $\vartheta \in \Theta$ per la quale esista j^* tale che $1 \leq j^* \leq k$, $\vartheta_{j^*} = 1, \omega_{j^*} > \lfloor \frac{\beta}{2} \rfloor$, allora vale la seguente equazione

$$\sum_{j=1, \dots, k: \omega_j \leq \beta - \omega_{j^*}} \omega_j \vartheta_j = \beta - \omega_{j^*}$$

Dato un intero $q \in \mathbb{Z}_+$, tale che $\frac{\beta}{2} < q < \beta$, qualsiasi soluzione $\vartheta \in \Theta$ può contenere al massimo una variabile $\vartheta_{j^*} = 1$ tale che $\omega_{j^*} \geq q$. In tal caso, ϑ deve contenere almeno una variabile ϑ_j con $\omega_j \leq \beta - \omega_{j^*}$ tale che $\sum_{j=1, \dots, k: \omega_j \leq \beta - \omega_{j^*}} \omega_j \vartheta_j = \beta - \omega_{j^*}$. Questa osservazione porta alla seguente disequazione che è soddisfatta da qualsiasi soluzione $\vartheta \in \Theta$

$$\sum_{j=1, \dots, k: \omega_j \leq \beta - q} \omega_j \vartheta_j \geq \sum_{j=1, \dots, k: q \leq \omega_j < \beta} (\beta - \omega_j) \vartheta_j \quad \frac{\beta}{2} < q < \beta \quad (3.17)$$

Sia \mathcal{F} l'insieme di tutte le coppie (j, q) tali che $j \in T$ e $\frac{\beta}{2} < q < \beta$. Per ogni coppia (j, q) ed ogni pattern $\ell \in W$, si definiscano i seguenti coefficienti

$$\lambda_\ell(j, q) = \begin{cases} \omega_j^\ell & \text{se } \omega_j^\ell \leq b_j - q \\ \omega_j^\ell - b_j & \text{se } q \leq \omega_j^\ell < b_j \\ 0 & \text{altrimenti} \end{cases}$$

Allora, la disequazione (3.17) può essere aggiunta al problema P'_N come

$$(FSB) \quad \sum_{\ell \in W} \lambda_\ell(j, q) \xi_\ell \geq 0 \quad (j, q) \in \mathcal{F} \quad (3.18)$$

Data una soluzione ottima, ξ , del problema P'_N , le disequazioni (3.18) possono essere separate per enumerazione completa.

3.6.1.4 Disequazioni Couple (CPL)

Si consideri l'insieme \mathcal{P} definito nel paragrafo 3.3.1.2. È triviale osservare che qualsiasi soluzione $\vartheta \in \mathcal{P}$ contiene al massimo due variabili ϑ_j e ϑ_h , $1 \leq j, h \leq k$, uguali a 1 tali che $\frac{\beta}{2} < \omega_j < \beta$ e $\frac{\beta - \omega_j}{2} < \omega_h < \beta - \omega_j$. In tal caso, deve esistere almeno una variabile ϑ_s , tale che $0 < \omega_s \leq \beta - \omega_j - \omega_h$, nella soluzione. Dunque, sia r l'indice di una data partizione e siano q_1 e q_2 definiti come $\frac{\beta}{2} < q_1 < \beta$, $\frac{\beta - q_1}{2} < q_2 < \beta - q_1$, allora la disequazione

$$\sum_{j \in K: \omega_j \leq \beta - q_1 - q_2} \vartheta_j - \sum_{j \in K: \omega_j = q_1} \vartheta_j - \sum_{j \in K: \omega_j = q_2} \vartheta_j \geq -1 \quad (3.19)$$

è osservata da qualsiasi soluzione $\vartheta \in \mathcal{P}$.

Sia \mathcal{Q} definito come $\mathcal{Q} = \{(i, j, q_1, q_2): i \in S, j \in T, \frac{b_j}{2} < q_1 < b_j, \frac{b_j - q_1}{2} < q_2 < b_j - q_1\}$. Per ogni $Q = (i, j, q_1, q_2) \in \mathcal{Q}$ ed ogni pattern $\ell \in \mathcal{W}$, si definiscano i seguenti coefficienti

$$\pi_\ell(Q) = \begin{cases} 1 & \text{se } w_j^\ell \leq b_j - q_1 - q_2 \\ -1 & \text{se } w_j^\ell = q_1 \\ -1 & \text{se } w_j^\ell = q_2 \text{ e } \ell \in \mathcal{W}_i \\ 0 & \text{altrimenti} \end{cases}$$

la seguente famiglia di disequazioni, dette *disequazioni Couple* (CPL), può essere aggiunta alla formulazione P'_N

$$(CPL) \quad \sum_{\ell \in \mathcal{W}} \pi_\ell(Q) \geq -1 \quad Q \in \mathcal{Q} \quad (3.20)$$

Data una soluzione ottima, ξ , del problema P'_N , le disequazioni (3.20) possono essere separate per enumerazione completa.

3.6.1.5 Disequazioni di Chvátal-Gomory (CGD-CGU)

Qualsiasi soluzione di P_N deve soddisfare le seguenti, ben note, disequazioni di *Chvátal-Gomory* (Nemhauser e Wolsey (1999)), definite considerando un vincolo (2.30) alla volta e un moltiplicatore intero $q \in \mathbb{Z}$

$$(CGU) \quad \sum_{\ell \in \mathcal{W}} \left\lfloor \frac{w_j^\ell}{q} \right\rfloor \xi_\ell \leq \left\lfloor \frac{b_j}{q} \right\rfloor \quad 2 \leq q < b_j \quad j \in T \quad (3.21)$$

$$(CGD) \quad \sum_{\ell \in \mathcal{W}} \left\lceil \frac{w_j^\ell}{q} \right\rceil \xi_\ell \geq \left\lceil \frac{b_j}{q} \right\rceil \quad 2 \leq q < b_j \quad j \in T \quad (3.22)$$

Data una soluzione ottima, ξ , del problema P'_N , le disequazioni (3.21) e (3.22) possono essere separate tramite enumerazione completa.

3.6.1.6 Lifting delle disequazioni di Chvátal-Gomory (LCGD)

Si consideri la disequazione (3.21) per una data destinazione $j \in T$ e un dato $2 \leq q \leq b_j$. Tale disequazione afferma che al massimo $\lfloor \frac{b_j}{q} \rfloor$ pattern ξ_ℓ che inviino un flusso q alla destinazione j possono appartenere ad una qualsiasi soluzione ammissibile di un FCTP e che, se un pattern ℓ invia un flusso strettamente maggiore di q alla destinazione j , esso può essere considerato con un coefficiente $\lfloor \frac{w_j^\ell}{q} \rfloor$. Si consideri ora un pattern ℓ^* tale che $w_j^{\ell^*} = b_j - \lfloor \frac{b_j}{q} \rfloor q + 1$. È facile notare che la disequazione (3.21) può essere elevata aggiungendo una variabile ξ_{ℓ^*} come

$$\sum_{\ell \in \mathcal{W}: w_j^\ell \geq q} \left\lfloor \frac{w_j^\ell}{q} \right\rfloor \xi_\ell + \xi_{\ell^*} \leq \left\lfloor \frac{b_j}{q} \right\rfloor \quad 2 \leq q < b_j \quad j \in T \quad (3.23)$$

La disequazione (3.23) può ulteriormente essere estesa aggiungendo qualsiasi altra variabile $\xi_\ell, \ell \in \mathcal{W}$, tale che $w_j^\ell \geq w_j^{\ell^*}$ e che abbia una relazione di mutua esclusione con la variabile ξ_{ℓ^*} . Dunque, è ulteriormente possibile rendere più stringente la (3.23) come segue

$$\sum_{i \in S} \sum_{\ell \in \mathcal{W}_i: w_j^\ell \geq q} \left\lfloor \frac{w_j^\ell}{q} \right\rfloor \xi_\ell + \sum_{\ell \in \mathcal{W}_r: \rho_{qj} \leq w_j^\ell < q} \xi_\ell \leq \left\lfloor \frac{b_j}{q} \right\rfloor \quad 2 \leq q \leq b_j \quad r \in S \quad j \in T \quad (3.24)$$

ove $\rho_{qj} = b_j - \lfloor \frac{b_j}{q} \rfloor q + 1$.

Definendo

$$v_\ell(q, i, j) = \begin{cases} 1 & \text{se } \ell \in \mathcal{W}_i \text{ e } \rho_{qj} \leq w_j^\ell < q \\ \left\lfloor \frac{w_j^\ell}{q} \right\rfloor & \text{se } w_j^\ell \geq q \\ 0 & \text{altrimenti} \end{cases}$$

Possiamo riscrivere i vincoli (3.24) come

$$(LCGD) \quad \sum_{\ell \in \mathcal{W}} v_\ell(q, i, j) \xi_\ell \leq \left\lfloor \frac{b_j}{q} \right\rfloor \quad 2 \leq q < b_j \quad r \in S \quad j \in T \quad (3.25)$$

Data una soluzione ottima, ξ , del problema P'_N , le disequazioni (3.25) possono essere separate tramite enumerazione completa.

3.6.2 Risoluzione del problema di pricing

Sia \overline{P}'_N il problema ottenuto aggiungendo a P'_N le disequazioni (3.12), (3.15), (3.18), (3.20) – (3.22) e (3.25).

Siano $v_j \in \mathbb{R}$, $u_i \in \mathbb{R}$, $g_{KL} \in \mathbb{R}_+$, $h_{C_j} \in \mathbb{R}_-$, $\alpha_{jq} \in \mathbb{R}_+$, $\eta_Q \in \mathbb{R}_+$, $\mu_{qj} \in \mathbb{R}_-$, $\chi_{qj} \in \mathbb{R}_+$ e $\psi_{qij} \in \mathbb{R}_-$ le variabili duali associate con i vincoli (2.30), (2.31), (3.12), (3.15), (3.18), (3.20) – (3.22) e (3.25), rispettivamente.

Gli autori dimostrano come, data una qualsiasi soluzione duale di \overline{P}'_N , $(v, u, g, h, \alpha, \eta, \mu, \chi, \psi)$, il problema del pricing delle colonne consiste nel trovare il pattern dell'insieme \mathcal{W} con costo ridotto negativo. Ciò viene eseguito tramite la risoluzione di un *Multiple Choice Knapsack Problem* con una ricorsione di programmazione dinamica (cfr. Roberti et al. (2013)).

3.6.3 Un esatto per la risoluzione del FCTP

Di seguito vengono riassunte la procedura di bounding e quella di branching sviluppate da Roberti et al. (2013). Come esplicitato dagli autori stessi nel loro paper, si ricorda che l'algoritmo è stato sviluppato in linguaggio C utilizzando come IDE e compilatore Visual Studio 2010 64-bit e come solver di LP e IP Cplex 12.5.

3.6.3.1 Procedura di bounding

In quanto segue, denotiamo con \overline{z}'_N^* il lower bound finale e con $\overline{z}'_{N\Delta}$ il lower bound ottenuto aggiungendo soltanto i primi Δ tipi di disequazioni al problema \overline{P}'_N .

- (1) **Inizializzazione di \overline{P}'_N .** Inizializzare \overline{P}'_N con l'insieme dei pattern corrispondenti alla soluzione base ammissibile calcolata con un algoritmo *greedy*. Impostare $\Delta = 0$.
- (2) **Risoluzione di \overline{P}'_N .** Risolvere \overline{P}'_N con un generico solver di LP. Sia $(u, v, g, h, \alpha, \eta, \mu, \chi, \psi)$ la soluzione duale ottima di \overline{P}'_N e sia \overline{z}'_N il suo costo.

(3) Generazione delle colonne. Calcolare, per ogni origine $i \in S$, il pater di costo ridotto minimo rispetto a $(\mathbf{u}, \mathbf{v}, \mathbf{g}, \mathbf{h}, \boldsymbol{\alpha}, \boldsymbol{\eta}, \boldsymbol{\mu}, \boldsymbol{\chi}, \boldsymbol{\psi})$, tramite la procedura di pricing delle colonne.

- Se esistono pattern di costo ridotto negativo, aggiungerli a \overline{P}'_N . Se $\Delta = 0$, tornare allo step 2.
- Altrimenti, se $\Delta = 0$, impostare $z'_N = \overline{z}'_N$ e $\Delta = \Delta + 1$.

(4) Risoluzione di \overline{P}'_N . Risolvere \overline{P}'_N con un solver di LP generico. Sia ξ la soluzione primale ottima di \overline{P}'_N .

(5) Separazione dei tagli.

- Se $\Delta \geq 1$, aggiungere, per ogni destinazione $j \in T$, le disequazioni CGD (3.21) e CGU (3.22) più violate da ξ .
- Se $\Delta \geq 2$, aggiungere, per ogni destinazione $j \in T$, la disequazione LCGD (3.25) più violata da ξ .
- Se $\Delta \geq 3$, aggiungere, per ogni destinazione $j \in T$, la disequazione FSB (3.18) più violata da ξ .
- Se $\Delta \geq 4$, aggiungere, per ogni destinazione $j \in T$, la disequazione EGUBC (3.15) più violata da ξ .
- Se $\Delta \geq 5$, aggiungere, per ogni destinazione $j \in T$, la disequazione CPL (3.20) più violata da ξ .
- Se $\Delta \geq 6$, aggiungere la disequazione SC (3.12) più violata da ξ .

(6) Criteri di arresto

- Se, all'iterazione corrente, nessuna colonna è stata aggiunta allo step 3 e nessun taglio è stato aggiunto allo step 5, impostare $\overline{z}'_{N_\Delta} = \overline{z}'_N$ e $\Delta = \Delta + 1$.
- Se $\Delta \leq 6$, tornare allo step 2. Altrimenti, impostare $\overline{z}'_{N^*} = \overline{z}'_N$ e terminare.

3.6.3.2 Strategia di branching

L'algoritmo *branch-and-price* (B&P) viene eseguito dopo la procedura di bounding, descritta nel paragrafo precedente, nel caso in cui il lower bound \overline{z}'_{N^*} non corrisponda ad una soluzione intera ammissibile per il FCTP. B&P risolve il

3.6 Il metodo esatto di Roberti, Bartolini e Mingozzi (2013)

problema intero ottenuto aggiungendo, a P'_N tutte le disequazioni valide separate dalla procedura di bounding.

Il branching avviene sulle variabili aggregate $y_{ij} = \sum_{\ell \in \mathcal{W}_i: w_j^\ell > 0} \xi^\ell$. Data una soluzione frazionaria ξ , l'algoritmo sceglie l'arco $(i, j) \in E$ con associata la variabile y_{ij} con il valore più prossimo a 0,6 e impone la disgiunzione $y_{ij} = 0 \vee y_{ij} = 1$. I pareggi sono rotti scegliendo l'arco con il costo fisso f_{ij} maggiore.

Il lower bound è calcolato ad ogni nodo dell'albero di branching utilizzando la generazione di colonne senza aggiungere alcun'altra disequazione valida. I nodi sono esplorati con strategia *best-bound-first*.

4. Metodi di risoluzione euristici

In letteratura sono stati presentati vari algoritmi euristici per il FCTP. Metodi basati sul rilassamento lineare P' sono stati proposti da Wright e Haehling von Lanzenuer (1989) e Wright e Haehling von Lanzenuer (1991). La maggior parte dei metodi euristici, sviluppati sia per il FCTP che per il FCP, ricercano soluzioni duali buone muovendosi da una base ammissibile (BFS) all'altra (Balinsky (1961), Kuhn e Baumol (1961), Dwyer (1966), Steinberg (1970), Walker (1976), Shetty (1990), Sun e McKeown (1993)). Tra questi, Cooper (1975), presenta un euristico che fa uso di tre fasi di inserimento in base di uno, due e tre archi contemporaneamente, rispettivamente.

Un approccio *tabu search*, basato su memorie *recency-based* e *frequency-based* (ossia memorie che considerano il tempo di permanenza in base o fuori dalla base e la frequenza di ingresso o uscita dalla base, rispettivamente), insieme con due strategie di memoria a medio e lungo termine, è stato proposto da Sun et al. (1998). Adlakha e Kowalsky (2003), hanno proposto un semplice euristico per risolvere piccole istanze di FCTP. Glover et al. (2005), hanno sviluppato un processo di tipo *parametric ghost image* per risolvere istanze di FCP e hanno testato il loro euristico su istanze di FCTP. Aguado (2009), ha presentato un approccio euristico basato su tecniche di rilassamento Lagrangiano, metodi di decomposizione e algoritmi *branch-and-cut* per risolvere una sequenza di problemi fondamentali. Le performance dell'algoritmo di Aguado sono state messe a confronto con quelle di Sun et al. (1998), e Glover et al. (2005), su alcune istanze ove il rapporto tra costi fissi e variabili è al massimo del 25%; risultati interessanti sono stati raggiunti su un insieme ristretto di istanze, nelle quali tale rapporto era approssimativamente del 5-10%.

Recentemente, El-Sherbiny e Alhamali (2013), hanno illustrato un algoritmo *particle swarm* (a sciame di particelle), testato su due istanze di dimensioni ridotte – la prima con 4 origini e 5 destinazioni e la seconda con 5 origini e 10 destinazioni. Lotfi e Tavakkoli-Moghaddam (2013), hanno presentato un algoritmo genetico, testato su istanze con fino a 30 origini e 50 destinazioni; non riportato alcun confronto con euristici in letteratura.

Buson et al. (2013), descrivono un euristico basato su una *local search* guidata dai costi ridotti derivati dalla nuova formulazione a tre indici proposta dagli stessi autori e già presentata nel paragrafo 2.4. I risultati di calcolo riportati da Buson et al. (2013) indicano che questo euristico domina tutti i metodi euristici fin ora presentati in letteratura.

4.1 Il metodo di Cooper

Nel 1975, Cooper ha presentato un algoritmo euristico per il FCP che, mediante scambi doppi e tripli di archi (i.e. portando in base 2-3 archi in una sola volta), consente di muoversi da una soluzione base ammissibile ad un'altra non adiacente a quella di provenienza (i.e. come dimostrato da Hirsch e Dantzig (1968), ciò corrisponde a spostarsi, nello spazio delle soluzioni, da un punto estremo ad un altro non adiacente).

Sia data la formulazione del FCP

$$P_{FCP} \begin{cases} z = \text{Min} & c'x + d'y & (4.1) \\ \text{s. t.} & Ax = b & (4.2) \\ & x \geq 0 & (4.3) \\ & y_j = \begin{cases} 0, & x_j = 0 \\ 1, & x_j > 0 \end{cases} & (4.4) \end{cases}$$

dove c , x , d e y sono m -vettori, b è un n -vettore e A è una matrice ($n \times m$).

Poiché è noto che la soluzione al problema si trovi in uno dei punti estremi dell'insieme $S = \{x | Ax = b, x \geq 0\}$, è normale pensare di utilizzare una metodologia *adjacent extreme point* (i.e. spostandosi tra punti estremi adiacenti). Allo stesso modo, è noto come sia possibile formulare un problema tale per cui ogni punto estremo sia un minimo locale, tramite una metodologia di programmazione lineare con algoritmo del simpleso standard. Ciò avviene cercando un nuovo valore della funzione obiettivo $\hat{z} < z$ facendo entrare, ad ogni iterazione, un vettore in base, tale da minimizzare

$$\frac{x_{Br}}{y_{rk}}(c_k - z_k) + (d_k - d_{Br}) \quad (4.5)$$

dove x_{Br} è la r -esima variabile in base, y_{rk} è l'elemento pivot nel tableau del semplice nel quale b_r verrà rimpiazzato nella base a_k . Si noti come a_k viene scelto tale che

$$\hat{z} = z + \frac{x_{Br}}{y_{rk}}(c_k - z_k) + (d_k - d_{Br}) < z \quad (4.6)$$

Il problema principale con tale metodologia è che raggiunge velocemente un minimo locale, il quale potrebbe essere molto distante dal minimo globale.

Consideriamo, ora, la possibilità di alternative o aggiunte alla nozione base dello spostarsi da un punto estremo di S ad un punto estremo adiacente. Se continuiamo a muoverci da un qualche punto estremo ad un altro, scambiando un vettore alla volta tramite l'operazione di pivoting del semplice, dobbiamo adottare criteri diversi da quello dato per minimizzare l'equazione (4.5). Assumendo che abbia ancora senso considerare come insieme potenziale delle soluzioni i punti estremi di S , necessitiamo di una metodologia e di un razionale per muoverci verso punti estremi non adiacenti. Relativamente all'operazione di pivoting del semplice, il seguente teorema ci fornisce alcune linee guida.

Teorema 4.1: Se $b_i, i = 1, 2, \dots, n$ formano un insieme di vettori base per E^n e $a_j = \sum_{i=1}^n \gamma_{ij} b_i, j = 1, 2, \dots, p \leq n$, allora l'insieme

$$Q = \{a_j, j = 1, 2, \dots, p \wedge b_i, i = p + 1, p + 2, \dots, n\}$$

fornisce una base per E^n se il determinante della matrice $\Gamma = \|\gamma_{ij}\|$ non è zero, i.e. $\det\|\gamma_{ij}\| \neq 0, i = 1, 2, \dots, p, j = 1, 2, \dots, p$.

Dimostrazione. Poiché l'insieme Q contiene n vettori, dobbiamo solo dimostrare che tali vettori sono linearmente indipendenti per stabilire che l'insieme Q è una base. Per far ciò, consideriamo l'espressione

$$\sum_{j=1}^p \alpha_j a_j + \sum_{i=p+1}^n \alpha_i b_i = 0 \quad (4.7)$$

Sappiamo che, dal momento che $b_i, i = 1, \dots, n$ costituisce una base, possiamo rappresentare qualsiasi vettore a_j in termini della base. In particolare

$$a_j = \sum_{i=1}^n \gamma_{ij} b_i \quad j = 1, 2, \dots, p \quad (4.8)$$

Sostituendo (4.8) in (4.7) otteniamo

$$\sum_{j=1}^p \sum_{i=1}^n \alpha_j \gamma_{ij} b_i + \sum_{i=p+1}^n \alpha_i b_i = 0 \quad (4.9)$$

Possiamo, dunque, riscrivere l'equazione (4.9) come

$$\sum_{j=1}^p \sum_{i=1}^p \alpha_j \gamma_{ij} b_i + \sum_{j=1}^p \sum_{i=p+1}^n \alpha_j \gamma_{ij} b_i + \sum_{i=p+1}^n \alpha_i b_i = 0 \quad (4.10)$$

Risistemando l'equazione (4.10) otteniamo

$$\sum_{i=1}^p \left[\sum_{j=1}^p \alpha_j \gamma_{ij} \right] b_i + \sum_{i=p+1}^n \left[\alpha_i + \sum_{j=1}^p \alpha_j \gamma_{ij} \right] b_i = 0 \quad (4.11)$$

Per provare l'indipendenza lineare, ciascuno dei coefficienti dei b_i nell'equazione (4.11) deve annullarsi, i.e.

$$\sum_{j=1}^p \alpha_j \gamma_{ij} = 0 \quad i = 1, 2, \dots, p \quad (4.12)$$

$$\alpha_i + \sum_{j=1}^p \alpha_j \gamma_{ij} = 0 \quad i = p + 1, p + 2, \dots, n \quad (4.13)$$

Il teorema dimostrato sopra fornisce una condizione sufficiente per rimpiazzare più vettori in base alla volta in un algoritmo *simplex-like*. Tuttavia, i dettagli di come ciò possa esser fatto sono ben lontani dall'essere semplici. Precedenti studi in merito comprendono quelli eseguiti da Bittner e Juttler (1962), Paranjape (1965), e Blocher (1966).

4.1.1 L'algoritmo

Il metodo di Cooper considera un rimpiazzo a p variabili, dove $p \geq 2$, in modo tale da evitare la situazione di stallo in un minimo locale.

Esso consiste nei seguenti passi:

- (1) Si ottenga una soluzione base ammissibile al problema $Ax = b, x \geq 0$.
- (2) Utilizzare uno scambio a un vettore per raggiungere un minimo locale.
- (3) Utilizzare uno scambio a due vettori. Se nessuna coppia di vettori migliora il valore di z , procedere allo step 4. Altrimenti, tornare allo step 2.
- (4) Utilizzare uno scambio a tre vettori. Se nessuna tripla di vettori migliora il valore di z , procedere allo step 5. Altrimenti, tornare allo step 2.
- (5) Inserire in base il primo vettore non base, senza considerare l'apporto alla funzione obiettivo, ed etichettarlo. Ritornare allo step 3. Il processo termina quando tutti i vettori non base sono stati etichettati.

4.2 Il metodo di Adlakha e Kowalski

Nel 2003, Adlakha e Kowalski proposero un semplice euristico per la risoluzione di FCTP di piccole dimensioni. Tale algoritmo, a differenza dei metodi di Cooper e Drebes (1967), Drenzler (1969), Steinberg (1970), Cooper (1975), Walker (1976) e Sun et al. (1998), tenta di ottenere la migliore soluzione nella prima iterazione e utilizza iterazioni *simplex-like* per verificare le condizioni di minimo o, nel caso, operare la convergenza finale.

4.2.1 Fase 1: soluzione ammissibile di partenza

La ricerca della soluzione ammissibile di partenza si basa sul rilassamento lineare P' , già proposta nel paragrafo 2.3.1, su di un algoritmo di *pre-screening* e sul *Metodo di Approssimazione di Vogel (VAM)*. Inoltre, la matrice *pre-VAM* viene *Ungarizzata* (i.e. processo di riduzione dei costi per righe e colonne).

L'algoritmo di *pre-screening* mira ad identificare tutti i costi fissi f_{st}^* che appariranno in ogni possibile distribuzione (x_{ij}, y_{ij}) per un dato FCTP.

$$Z = f_{st}^* + \min \sum \sum \{c_{ij}x_{ij} + (f_{ij}y_{ij} | i \neq s \wedge j \neq t)\}$$

ove $f_{ij} = f_{st}^*$ se $\sum_{i \neq s} a_i < b_t$ o $\sum_{j \neq t} b_j < a_s$.

Dopo aver identificato un costo fisso f_{st}^* , il valore di f_{st} è impostato a 0 per risolvere il FCTP e tale costo fisso è aggiunto al costo della soluzione finale.

Di seguito sono riportati i passi principali della procedura per la ricerca della soluzione ammissibile di partenza.

- (1) Utilizzare l'algoritmo di pre-screening per identificare e separare ogni costo fisso f_{st}^* .
- (2) Generare la formulazione P' .
- (3) Ungarizzare la matrice relativa a P' .
- (4) Identificare le location critiche utilizzando VAM.
- (5) Caricare le location critiche trovate con il massimo quantitativo ammissibile.
- (6) Ridurre/assestare la matrice relativa a P' eliminando i vincoli soddisfatti.
- (7) Se l'assegnamento non è completo, tornare allo step 1.
- (8) Stop. Salvare la soluzione $x^{(1)}$ come soluzione ammissibile di partenza.

4.2.2 Fase 2: miglioramento e verifica dell'ottimo

La seconda parte del metodo, basata sulla concavità della funzione obiettivo, investiga complessivamente tutti i picchi adiacenti alla miglior soluzione trovata. La ricerca dei picchi viene eseguita perturbando la distribuzione di carico ottenuta nella fase 1 tramite elementi della *one step pivot operation* di Murty (1968). Nello specifico, i passi della fase 2 sono i seguenti:

- (1) Impostare $i = 1$.
- (2) Determinare tutti i picchi adiacenti a $x^{(i)}$. Sia x^* il picco con il più basso valore di $Z(x)$.
- (3) Se $Z(x^{(i)}) < Z(x^{(*)})$, procedere allo step 5.
- (4) Impostare $i = i + 1$. Sia $x^{(i)} = x^*$. Ritornare allo step 2.
- (5) Stop. Salvare $x^{(i)}$ come soluzione ottima del FCTP.

Gli autori, Adlakha e Kowalski, riportano due esempi illustrativi nel loro paper.

4.3 Il metodo di Aguado

Nel 2009, Jesús Sáez Aguado ha presentato un algoritmo *branch-and-cut* basato su tre fasi distinte:

- (1) rilassamento o decomposizione Lagrangiana del problema iniziale;

- (2) **(Core 2):** definizione di sotto problemi con un numero minore di variabili rispetto al problema originale tramite rilassamento e decomposizione Lagrangiana, basata su due fattori, K e F , indicanti rispettivamente il minimo numero di indici (i, j) voluti nel sotto problema e un fattore atto a compensare i costi fissi e variabili.
- (3) **(Core 3):** enumerazione tramite l'applicazione di un algoritmo *branch-and-cut*, con tempo limitato, al sotto problema che ha prodotto la soluzione migliore nella fase 2.

4.3.1 Rilassamento e Decomposizione Lagrangiana

Sia $\mathcal{F} = \{(x, y) \in R^{2nm}: (2.6) - (2.10)\}$ e sia $Y_p = \text{proj}(\mathcal{F})$ la proiezione di \mathcal{F} nello spazio delle variabili y , ossia $Y_p = \{y \in \{0,1\}^{nm}: \exists x \in R^{nm} | (x, y) \in \mathcal{F}\}$.

Poiché l'insieme Y_p è definito solo implicitamente, la risoluzione di un problema di ottimizzazione lineare su di esso richiederebbe la risoluzione di una serie di problemi *set covering*, il che renderebbe l'ottimizzazione piuttosto inefficiente. Aguado, considera, invece, una serie di rilassamenti dell'insieme Y_p , così che la risoluzione dei sotto problemi corrispondenti risulti più facile.

Siano

$Y_b = \{0,1\}^{nm}$, l'insieme binario di dimensione nm .

$Y_r = \{y \in \{0,1\}^{nm}: \sum_{j=1}^m u_{ij}y_{ij} \geq a_i, i = 1, \dots, n\}$ l'insieme *knapsack* delle righe. Esso è separabile, per ogni $i = 1, \dots, n$, ottenendo l'insieme *knapsack* per la riga i

$$Y_i = \left\{ (y_{ij}) \in \{0,1\}^m: \sum_{j=1}^m u_{ij}y_{ij} \geq a_i \right\} \quad (4.14)$$

$Y_c = \{y \in \{0,1\}^{nm}: \sum_{i=1}^n u_{ij}y_{ij} \geq b_j, j = 1, \dots, m\}$ l'insieme *knapsack* delle colonne. Anch'esso, come il precedente, è separabile, per ogni $j = 1, \dots, m$, ottenendo l'insieme *knapsack* per la colonna j

$$Y^j = \left\{ (y_{ij}) \in \{0,1\}^n: \sum_{i=1}^n u_{ij}y_{ij} \geq b_j \right\} \quad (4.15)$$

Date tali definizioni, la relazioni $Y_p \subseteq Y_r \subseteq Y_b$ e $Y_p \subseteq Y_c \subseteq Y_b$ sono verificate. Poiché sappiamo che più l'insieme su cui si esegue l'ottimizzazione è piccolo, più il corrispondente sarà forte e, d'altro canto, la risoluzione del sotto problema corrispondente sarà difficile, gli insiemi Y_r e Y_c forniscono una buona via di mezzo.

Una volta che un simile insieme Y è stato scelto, risulta possibile aggiungere al problema i vincoli ridondanti $y \in Y$, ottenendo

$$z = \text{Min} \quad \sum_{i=1}^n \sum_{j=1}^m c_{ij}x_{ij} + \sum_{i=1}^n \sum_{j=1}^m f_{ij}y_{ij}$$

$$\text{s. t.} \quad x \in X \quad (4.16)$$

$$y \in Y \quad (4.17)$$

$$x_{ij} \leq u_{ij}y_{ij} \quad \forall i, j \quad (4.18)$$

Dualizzando i vincoli (4.18) con moltiplicatori $\lambda_{ij} \geq 0$, il sotto problema Lagrangiano ha forma

$$(LR(\lambda)) \quad z = \text{Min} \quad \sum_{i=1}^n \sum_{j=1}^m \{c_{ij}x_{ij} + f_{ij}y_{ij} + \lambda_{ij}(x_{ij} - u_{ij}y_{ij})\} \quad (4.19)$$

$$\text{s. t.} \quad x \in X$$

$$y \in Y$$

e può essere ulteriormente decomposto in due sotto problemi

$$(LR_1(\lambda)) \quad z = \text{Min} \quad \sum_{i=1}^n \sum_{j=1}^m \{(c_{ij} + \lambda_{ij})x_{ij}\} \quad (4.20)$$

$$\text{s. t.} \quad x \in X$$

$$(LR_2(\lambda)) \quad z = \text{Min} \quad \sum_{i=1}^n \sum_{j=1}^m \{(f_{ij} - \lambda_{ij}u_{ij})y_{ij}\} \quad (4.21)$$

$$\text{s. t.} \quad y \in Y$$

Il sottoproblema $(LR_1(\lambda))$ è il ben noto problema dei trasporti lineare. La forma del sottoproblema $(LR_2(\lambda))$ dipende, invece, dall'insieme Y scelto. Per esempio, Wright e Haehling von Lanzenuer (1989), Wright e Haehling von Lanzenuer (1991), e Cruz et al. (1998), hanno usato il rilassamento Lagrangiano con $Y = Y_b$. Tuttavia, la risoluzione del relativo problema duale Lagrangiano non migliora il bound ottenuto con il rilassamento lineare. Aguado (2009), invece, utilizza

sottoinsieme strettamente contenuti in Y_b (i.e. Y_r e Y_b), ottenendo, ad esempio per $Y = Y_r$, il sotto problema

$$(LR_2(\lambda)) \quad z = \text{Min} \quad \sum_{i=1}^n \sum_{j=1}^m \{(f_{ij} - \lambda_{ij} u_{ij}) y_{ij}\}$$

$$\text{s. t.} \quad \sum_{j=1}^m u_{ij} y_{ij} \geq a_i \quad i = 1, \dots, n$$

$$y_{ij} \in \{0,1\} \quad i = 1, \dots, n \quad j = 1, \dots, m$$

il quale può essere decomposto in n sotto problemi *knapsack* binari, ciascuno con m variabili, per i quali esistono algoritmi di risoluzione specifici (Martello e Toth (1990)).

Dato un qualsiasi insieme Y come sopra, denotiamo con $LD(Y)$ il duale Lagrangiano risultante, ossia

$$\max_{\lambda \geq 0} \{v(LR(\lambda))\}$$

ove $v(LR(\lambda))$ indica il valore della soluzione ottima del problema $LR(\lambda)$. Come conseguenza delle relazioni di inclusione sopra descritte, le seguenti disequazioni appaiono chiare

$$v(P') = v(LD(Y_b)) \leq v(LD(Y_r)) \leq z^*$$

Per la risoluzione del problema duale, Aguado propone l'utilizzo di un algoritmo *subgradient type*.

Aggiungendo i vincoli ridondanti $y \in Y_r$ e $y \in Y_c$ alla formulazione del FCTP, e eseguendo uno split di variabili, si ottiene la formulazione

$$z = \text{Min} \quad cx + f y_1$$

$$\text{s. t.} \quad x \in X$$

$$y_1 \in Y_r$$

$$y_2 \in Y_c$$

$$x \leq u y_1 \quad (4.22)$$

$$y_1 = y_2 \quad (4.23)$$

Dualizzando i vincoli (4.22) con moltiplicatori non negativi $\lambda_{ij} \geq 0$ ed i vincoli (4.23) con moltiplicatori liberi $\mu_{ij}, i = 1, \dots, n, j = 1, \dots, m$, si ottiene il sottoproblema Lagrangiano $(LD(\lambda, \mu))$, separabile in tre sottoproblemi

$$(LR_1(\lambda)) \quad z = \text{Min} \quad (c + \lambda)x \quad (4.24)$$

$$s. t. \quad x \in X$$

$$(LR_3(\lambda, \mu)) \quad z = \text{Min} \quad (f + \mu - \lambda u)y_1 \quad (4.25)$$

$$s. t. \quad y_1 \in Y_r$$

$$(LR_4(\lambda, \mu)) \quad z = \text{Min} \quad (-\mu)y_2 \quad (4.26)$$

$$s. t. \quad y_2 \in Y_c$$

Come visto in precedenza i sotto problemi $LR_3(\lambda, \mu)$ e $LR_4(\lambda, \mu)$ si decompongono in n e m sotto problemi *knapsack*, rispettivamente. Indicando con $LD(Y_r, Y_c)$ il duale Lagrangiano corrispondente, è noto come il valore della sua soluzione ottima sia sempre migliore o uguale a quello corrispondente ai rilassamenti semplici Y_r o Y_c , ossia

$$\max \left(v(LD(Y_r)), v(LD(Y_c)) \right) \leq v(LD(Y_r, Y_c))$$

Come nel caso del rilassamento Lagrangiano il problema duale

$$\max_{\lambda \geq 0, \mu} \{v(LD(\lambda, \mu))\}$$

va risolto con un algoritmo *subgradient type*.

4.3.2 Generazione dei sotto problemi

A partire dalla soluzione del problema duale, ottenuta nella fase 1, avendo a disposizione i moltiplicatori ottimi $\bar{\lambda}_{ij}$ del rilassamento Lagrangiano e $(\bar{\lambda}_{ij}, \bar{\mu}_{ij})$ della decomposizione Lagrangiana, vengono definiti dei sotto problemi contenenti le variabili che ci si aspetta di trovare nella soluzione ottima. Quando una variabile x_{ij} fa parte di un sotto problema, anche la corrispondente y_{ij} sarà presente. Pertanto, considerando le coppie (i, j) , il sotto problema sarà definito da un insieme di coppie

$$S \subset \{1, \dots, n\} \times \{1, \dots, m\}$$

Se nella fase 1 è stato applicato il rilassamento Lagrangiano, i costi ridotti per le variabili x_{ij} e y_{ij} sono, rispettivamente, $c_{ij} + \bar{\lambda}_{ij}$ e $f_{ij} - u_{ij}\bar{\lambda}_{ij}$. Nel caso della decomposizione Lagrangiana, invece, il costo ridotto della variabile y_{ij} è uguale a $f_{ij} - u_{ij}\bar{\lambda}_{ij} + \bar{\mu}_{ij}$ nel primo sotto problema e $-\bar{\mu}_{ij}$ nel secondo. Definiamo un unico costo ridotto per la variabile y_{ij} aggiungendo entrambi i costi così da ottenere $f_{ij} - u_{ij}\bar{\lambda}_{ij}$.

Pertanto i costi Lagrangiani ridotti per le coppie (i, j) sono definiti come

$$\pi_{ij} = f_{ij} - u_{ij}\bar{\lambda}_{ij} + F(c_{ij} + \bar{\lambda}_{ij})$$

ove F è un fattore di compensazione per costi fissi e variabili. Per definire l'insieme degli indici S presenti nel sotto problema, si imposta una soglia π e si scelgono tutte le coppie il cui costo ridotto non eccede π , ossia

$$S = \{(i, j) : \pi_{ij} < \pi\}$$

Il sotto problema ha la stessa struttura dell'originale ma in formato sparso.

$$z = \text{Min} \sum_{(i,j) \in S} (c_{ij}x_{ij} + f_{ij}y_{ij})$$

$$s. t. \sum_{j \in S_i} x_{ij} = a_i \quad i = 1, \dots, n \quad (4.27)$$

$$\sum_{i \in S^j} x_{ij} = b_j \quad j = 1, \dots, m \quad (4.28)$$

$$x_{ij} \geq 0 \quad (i, j) \in S \quad (4.29)$$

$$x_{ij} \leq u_{ij}y_{ij} \quad (i, j) \in S \quad (4.30)$$

$$y_{ij} \in \{0,1\} \quad (i, j) \in S \quad (4.31)$$

Il rilassamento e la decomposizione Lagrangiani vengono applicati ulteriormente ai sotto problemi per ottenere la miglior soluzione euristica possibile. Grazie alla sparsità degli insiemi *knapsack* ottenuti, la risoluzione del rilassamento Lagrangiano o della decomposizione Lagrangiana risulta molto più veloce che per il problema completo.

4.3.3 Euristico per la risoluzione dei sotto problemi

Per la risoluzione dei sotto problemi, Aguado propone un *euristico migliorato*, prendendo spunto dalla procedura dinamica di *slope scaling* e dai *parametric ghost image processes*. Sia $x \in X(S)$ la soluzione alla corrente iterazione, \bar{x}_{ij} il massimo valore raggiunto dalla variabile x_{ij} fino a questa iterazione e siano i coefficienti della funzione obiettivo per la coppia $(i, j) \in S$ dati da

$$\lambda_{ij} = \begin{cases} c_{ij} + \frac{f_{ij}}{x_{ij}} & \text{se } x_{ij} > 0 \\ c_{ij} + \frac{f_{ij}}{\bar{x}_{ij}} & \text{se } x_{ij} = 0, \bar{x}_{ij} > 0 \\ \infty & \text{se } x_{ij} = 0, \bar{x}_{ij} = 0 \end{cases}$$

4.4 Il metodo di Glover, Amini e Kochenberger

Nel 2005, Glover, Amini e Kochenberger hanno presentato un nuovo algoritmo basato sull'utilizzo di una *parametric ghost image* (GIP, immagine fantasma parametrica), ossia un vettore parametrico non negativo $v = (v_j: j \in N)$ ed il vettore parametrizzato dei costi associato $c(v) = (c_j + \frac{F_j}{v_j}: j \in N)$, per fornire il rilassamento lineare parametrico del problema dei costi fissi

$$P'_v = \begin{cases} z'_v = \text{Min} & x \cdot c(v) \\ \text{s. t.} & Ax = b \\ & 0 \leq x \leq U \end{cases}$$

Una descrizione sommaria del metodo GIP per il problema dei costi fissi è la seguente:

- (0) Risolvere P'_v . Sia x^* la soluzione ottima. Impostare $v \leftarrow U$.
- (1) Risolvere P'_v e sia x' la soluzione.
- (2) A partire da x' , utilizzare delle regole per ottenere una soluzione raffinata e applicare un Metodo di Miglioramento per ottenere una soluzione ulteriormente raffinata x'' .
- (3) Aggiornare v come funzione del suo valore corrente e x'' . Se è stato raggiunto il numero massimo di iterazioni, terminare l'algoritmo con la miglior soluzione x^* . Altrimenti, tornare allo step 1.

Verranno ora definiti ulteriormente i vari passi dell'algoritmo, ciascuno con una breve descrizione a parole dello stesso.

In quanto segue indicheremo con

- U_o e U_j gli upper bound stimati per x_j che verranno introdotti al posto del bound originale U_j in alcuni calcoli dell'algoritmo;
- Max_{iter} il numero massimo di iterazioni dell'algoritmo;
- $Iter$ il numero dell'iterazione corrente;
- λ , λ_{max} e λ_{min} rispettivamente uno scalare ed i suoi valori massimo e minimo;
- λ_{iter} e λ_{litter} , rispettivamente, il numero di iterazioni dopo cui incrementare λ e il numero dell'ultima iterazione a cui λ è stato incrementato;
- λ_{Δ} l'incremento da applicare a λ ogni λ_{iter} iterazioni;
- z_{iter} il numero di iterazioni senza miglioramenti;
- τ il numero di iterazioni senza miglioramenti dopo cui il processo termina;
- B e NB gli insiemi delle variabili base e non base, rispettivamente;
- $N = B \cup NB$;
- ω un moltiplicatore positivo utilizzato per aggiornare il vettore parametrico v' .

4.4.1 Step 0

0.1 $Iter = Iter + 1$

0.2 Risolvere P' , ottenendo una soluzione ottima x'

0.3 $x^* = x'$

0.4 Impostare $U_o = \max_{j \in N} \{x'_j\}$ e $U_j = x'_j$

In questo step di inizializzazione, eseguito solo *una tantum* nel corso dell'intero algoritmo, viene risolto il problema P' , ottenendo una soluzione x' , la quale viene utilizzata per inizializzare la miglior soluzione x^* . Al fine di aggiornare, in seguito, il vettore parametrico v , U_o viene inizializzato al massimo valore x_j ottenuto nella

risoluzione di P' . Inoltre, U_j viene scelto come il massimo valore nella soluzione per ogni variabile x_j .

4.4.2 Step 1

1.1 $Iter = Iter + 1$

1.2 Se $Iter > Max_{iter}$, terminare il processo e restituire la miglior soluzione x^* .

1.3 Risolvere P'_v tramite post-ottimizzazione lineare, ottenendo una soluzione ottima x' .

1.4 Se $z' < z^*$, impostare $x^* = x'$.

1.5 Se $Iter \leq v_{iter}$, allora $U_j = \max_{j \in N} \{U_j, x'_j\}$.

In questo primo passo dell'algoritmo, nel caso si sia raggiunto il limite massimo di iterazioni, si restituisce la miglior soluzione trovata e si termina il processo. Altrimenti, a partire dall'istanza più recentemente risolta, si costruisce e risolve una nuova istanza di P'_v , ottenendo una soluzione ottima x' che, nel caso sia migliore di x^* , sostituisce quest'ultima. Viene inoltre aggiornato, nelle prime v_{iter} iterazioni, il vettore U_j con i valori massimi ottenuti da ogni x_j .

4.4.3 Step 2: Fase I

2.1.1 Facendo riferimento alla soluzione x' dello step 1, impostare, per ogni $j \in N$, $v_j = 0$ se $x'_j > 0$, $v_j = M$ altrimenti.

2.1.2 Partendo dalla base di P' che ha prodotto x' , identificare e risolvere il problema P'_v , ottenendo una soluzione ottima x'' .

2.1.3 Se $z'' < z^*$, allora $x^* = x''$.

2.1.4 Se $Iter \leq v_{iter}$, allora aggiornare $U_j = \max_{j \in N} \{U_j, x''_j\}$.

Per investigare il potenziale per ulteriori miglioramenti alla soluzione corrente, in questa prima fase i coefficienti delle variabili con valori non nulli e (virtualmente) nulli vengono settati a 0 e M , rispettivamente. Ciò che si ottiene è un nuovo P'_v che viene in seguito risolto tramite post-ottimizzazione. La nuova soluzione ottenuta, nel caso sia migliore di quella corrente, va a sostituire quest'ultima. Inoltre, durante le prime v_{iter} iterazioni, i valori U_j vengono aggiornati.

4.4.4 Step 2: Fase II

- 2.2.1 Considerare la base di FC corrente.
- 2.2.2 Impostare $v = 0$. Dunque, la funzione obiettivo di P'_v diviene $z'_v = cx''$.
- 2.2.3 Impostare $j^* = 0$ e $z_{j^*} = \infty$, inizializzando così l'indice di una "miglior" variabile non base x_{j^*} e il suo apporto alla funzione obiettivo z_{j^*} causato da un pivot che porti x_{j^*} in base. Impostare inoltre $k^* = 0$, indice della variabile base x_{k^*} che lascerà la base quando x_{j^*} entrerà.
- 2.2.4 Risolvere P'_v e considerare ogni variabile $x_j, j \in NB$ come candidata potenziale all'ingresso in base.
- 2.2.4.1 Condurre un *ratio test* per determinare la variabile uscente x_k e determinare il relativo apporto alla funzione obiettivo r_j . Nel caso in cui x_j , la variabile entrante, rimanga non basica spostandosi da un suo bound all'altro, allora nessuna variabile x_k è scelta, impostando $k = 0$.
- 2.2.4.2 Determinare il *net impact* del potenziale pivot sulla funzione obiettivo z'' , calcolato come

$$z_j = r_j + \sum (G_h : h \in P)$$

ove P è l'insieme degli archi sul cammino aumentante, determinato e comprendente l'arco per la variabile entrante x_j , e G_h è definito relativamente al costo fisso F_h dalle seguenti transizioni:

$$G_h = \begin{cases} F_h & x_h = 0 \Rightarrow x_h > 0 \\ -F_h & x_h > 0 \Rightarrow x_h = 0 \\ 0 & x_h \text{ invariato} \end{cases}$$

- 2.2.4.3 Se $z_j < z_{j^*}$, allora $j^* = j$ e $z_{j^*} = z_j$.
- 2.2.5 Se $z_{j^*} < 0$, allora effettuare il pivoting con la coppia di variabili (j^*, k^*) . Identificare la soluzione base corrispondente e salvarla come x'' . Tornare allo step 2.2.3 per tentare ulteriormente di migliorare la soluzione.

2.2.6 Altrimenti, se $z_j^* \geq 0$,

2.2.6.1 se $z'' < z^*$, allora $z^* = z''$ e $z_{iter} = 0$;

2.2.6.2 altrimenti, se $z'' \geq z^*$, allora $z_{iter} = z_{iter} + 1$.

2.2.7 Se $z_{iter} = \tau$, terminare il processo restituendo la miglior soluzione x^* .

2.2.8 Se $Iter \leq v_{iter}$, allora aggiornare $U_j = \max_{j \in N} \{U_j, x_j''\}$.

Per migliorare ulteriormente la soluzione corrente, la fase II si concentra sulla base di P' che ha generato x'' . In primis, il vettore dei coefficienti della funzione obiettivo viene rimpiazzato dal vettore originale dei costi variabili, c . In seguito, tramite un processo di pivoting per tentativi, ogni variabile x_j non base è considerata per l'ingresso in base, si esegue un *ratio test* per determinare la variabile uscente associata, x_k , seguito dal calcolo del relativo apporto alla funzione obiettivo, z_{jk} . La coppia associata con l'apporto più negativo è scelta per l'operazione di pivoting e il processo è reiterato fino a quando nessuna coppia (j, k) risulti migliorare la base. Ancora una volta, si calcola il costo della nuova soluzione, la quale, nel caso sia migliore di quella corrente, va a sostituire quest'ultima. Altrimenti, se il processo è giunto alla fase II per τ iterazioni senza ottenere miglioramenti si ferma; se, invece, $Iter < \tau$, i valori U_j vengono aggiornati.

4.4.5 Step 3

3.1 Se $Iter - \lambda_{liter} = \lambda_{iter}$ allora $\lambda = \lambda + \lambda_{\Delta}$ e $\lambda_{liter} = Iter$.

3.2 Se $\lambda > \lambda_{max}$, terminare il processo restituendo la miglior soluzione x^* .

3.3 Se $Iter \leq v_{iter}$, aggiornare $v_j' = \lambda \omega U_o + (1 - \lambda)x_j''$ per ogni $j \in N$.

3.4 Altrimenti, aggiornare $v_j' = \lambda \omega U_j + (1 - \lambda)x_j''$ per ogni $j \in N$.

3.5 Se $v = v'$, diversificare $v' = \lambda(U_j - x_j'') + (1 - \lambda)v_j$.

3.6 Impostare $v = v'$ e tornare allo step 1.

Notare che U_j presente nello step 3.5 rappresenta il bound originale, diverso dal bound *delegato* presente nello step 3.4.

In questo terzo step, la variabile λ viene incrementata di λ_{Δ} ogni λ_{iter} iterazioni. Se supera il valore massimo accettabile λ_{max} , il processo termina restituendo la miglior soluzione x^* . Altrimenti vengono applicate due diverse formule per

aggiornare il vettore parametrico v , la prima, basata sul concetto che un unico valore U_o fornisca un bound valido per ogni variabile, viene applicata durante le prime v_{iter} iterazioni, la seconda, in cui per ogni variabile si utilizza un bound U_j più sensibile alle differenze tra le singole variabili, nelle iterazioni successive. Inoltre, per differenziare il nuovo vettore parametrico, nel caso risulti invariato, una terza formula viene applicata. Infine, il processo torna allo step 1 per cercare ulteriori miglioramenti tramite il nuovo vettore parametrico ottenuto.

4.5 Il metodo euristico di Buson, Roberti e Toth (2013)

Come visto nel paragrafo 2.4, Buson et al. (2013), hanno presentato una nuova formulazione matematica a tre indici sulla base della quale hanno sviluppato un nuovo algoritmo euristico a ricerca locale con costi ridotti. Tale algoritmo si basa su quattro concetti, oltre a tale formulazione: delle nuove disequazioni valide, una procedura di *lower bounding*, una *local search* e una *restart phase*. Esso rappresenta lo stato dell'arte degli euristici per la risoluzione del FCTP e, pertanto, verrà descritto di seguito, riportando, inoltre, i risultati computazionali che lo definiscono tale.

4.5.1 Disequazioni valide per il problema

Al fine di rinforzare il lower bound z'_3 , Buson, Roberti e Toth, ritengono conveniente, in termini computazionali, l'aggiunta di un numero pseudo polinomiale di *tagli* di Chvátal-Gomory con un singolo moltiplicatore maggiore di 0.

Si consideri il vincolo (2.16) per una data origine $i \in S$. Dato $\sigma \in \mathbb{N}, 1 < \sigma < a_i$, un taglio di Chvátal-Gomory valido per il problema P'_3 è

$$\sum_{j \in T} \sum_{q=1}^{m_{ij}} \left\lfloor \frac{q}{\sigma} \right\rfloor \xi_{ij}^q \leq \left\lfloor \frac{a_i}{\sigma} \right\rfloor$$

Sia \mathcal{S} l'insieme di tutte le coppie (i, σ) , con $i \in S$ e $1 < \sigma < a_i$ (i.e. $\mathcal{S} = \{(i, \sigma): i \in S, \sigma \in \mathbb{N}, 1 < \sigma < a_i\}$). I tagli CG seguenti sono validi per P'_3

$$\sum_{j \in T} \sum_{q=1}^{m_{ij}} \left\lfloor \frac{q}{\sigma} \right\rfloor \xi_{ij}^q \leq \left\lfloor \frac{a_i}{\sigma} \right\rfloor \quad (i, \sigma) \in \mathcal{S} \quad (4.32)$$

Tagli simili possono essere derivati dal vincolo (2.17). Considerando $\mathcal{T} = \{(j, \sigma) : j \in T, \sigma \in \mathbb{N}, 1 < \sigma < b_j\}$, i tagli CG seguenti sono validi per P'_3

$$\sum_{i \in S} \sum_{q=1}^{m_{ij}} \left\lfloor \frac{q}{\sigma} \right\rfloor \xi_{ij}^q \leq \left\lfloor \frac{b_j}{\sigma} \right\rfloor \quad (j, \sigma) \in \mathcal{T} \quad (4.33)$$

Due ulteriori insiemi di disequazioni valide possono essere derivate da (4.32) e (4.33), semplicemente sostituendo l'operazione di arrotondamento per difetto con un arrotondamento per eccesso e \leq con \geq . Pertanto, le seguenti disequazioni sono valide per P'_3

$$\sum_{j \in T} \sum_{q=1}^{m_{ij}} \left\lceil \frac{q}{\sigma} \right\rceil \xi_{ij}^q \geq \left\lceil \frac{a_i}{\sigma} \right\rceil \quad (i, \sigma) \in \mathcal{S} \quad (4.34)$$

e

$$\sum_{i \in S} \sum_{q=1}^{m_{ij}} \left\lceil \frac{q}{\sigma} \right\rceil \xi_{ij}^q \geq \left\lceil \frac{b_j}{\sigma} \right\rceil \quad (j, \sigma) \in \mathcal{T} \quad (4.35)$$

Tutti e quattro gli insiemi di tagli CG possono essere separati tramite enumerazione completa.

Da qui in avanti indichiamo con $\overline{P'_3}$ il problema P'_3 con aggiunti i tagli di cui sopra e con $\overline{z'_3}$ il costo della sua soluzione ottima.

Siano $\mathbf{u} \in \mathbb{R}^n$, $\mathbf{v} \in \mathbb{R}^m$, $\boldsymbol{\alpha} \in \mathbb{R}^{|\mathcal{S}|}$, $\boldsymbol{\beta} \in \mathbb{R}^{|\mathcal{T}|}$, $\boldsymbol{\gamma} \in \mathbb{R}_+^{|\mathcal{S}|}$ e $\boldsymbol{\delta} \in \mathbb{R}_+^{|\mathcal{T}|}$ le variabili duali associate ai vincoli (2.16), (2.17), (4.32) – (4.35), rispettivamente. Il duale del problema $\overline{P'_3}$, detto $\overline{D'_3}$, è il seguente

$$\begin{aligned}
 \overline{w}_3' = \text{Max} \quad & \sum_{i \in S} a_i u_i + \sum_{j \in T} b_j v_j \\
 & + \sum_{(i, \sigma) \in \mathcal{S}} \left\lfloor \frac{a_i}{\sigma} \right\rfloor \alpha_{i\sigma} + \sum_{(j, \sigma) \in \mathcal{T}} \left\lfloor \frac{b_j}{\sigma} \right\rfloor \beta_{j\sigma} \\
 & + \sum_{(i, \sigma) \in \mathcal{S}} \left\lfloor \frac{a_i}{\sigma} \right\rfloor \gamma_{i\sigma} + \sum_{(j, \sigma) \in \mathcal{T}} \left\lfloor \frac{b_j}{\sigma} \right\rfloor \delta_{j\sigma} \\
 \text{s. t.} \quad & qu_i + qv_j \\
 & + \sum_{(k, \sigma) \in \mathcal{S}: k=i} \left\lfloor \frac{q}{\sigma} \right\rfloor \alpha_{k\sigma} + \sum_{(k, \sigma) \in \mathcal{T}: k=j} \left\lfloor \frac{q}{\sigma} \right\rfloor \beta_{k\sigma} \\
 & + \sum_{(k, \sigma) \in \mathcal{S}: k=i} \left\lfloor \frac{q}{\sigma} \right\rfloor \gamma_{k\sigma} + \sum_{(k, \sigma) \in \mathcal{T}: k=j} \left\lfloor \frac{q}{\sigma} \right\rfloor \delta_{k\sigma} \leq d_{ij}^q \quad \{i, j\} \in E \quad q = 1, \dots, m_{ij} \\
 & u_i \in \mathbb{R} \quad i \in S \\
 & v_j \in \mathbb{R} \quad j \in T \\
 & \alpha_{i\sigma} \in \mathbb{R}_- \quad (i, \sigma) \in \mathcal{S} \\
 & \beta_{j\sigma} \in \mathbb{R}_- \quad (j, \sigma) \in \mathcal{T} \\
 & \gamma_{i\sigma} \in \mathbb{R}_+ \quad (i, \sigma) \in \mathcal{S} \\
 & \delta_{j\sigma} \in \mathbb{R}_+ \quad (j, \sigma) \in \mathcal{T}
 \end{aligned}$$

4.5.2 Procedura di Lower Bounding

La procedura di lower bounding si basa su una riformulazione del problema, da qui in poi chiamata *MP* (*master problem*), costituito da un insieme limitato di variabili ξ_{ij}^q e un sottoinsieme di tagli (4.32) – (4.35).

In primis, la procedura risolve il problema P' e utilizza la soluzione ottima per determinare l'insieme di variabili presenti in *MP*, definito dalle variabili $\xi_{ij}^q, i \in S, j \in T$, tali che $x_{ij} > 0$ più tutte le variabili $\xi_{ij}^{x_{ij}}, i \in S, j \in T$, tali che $0 < x_{ij} < m_{ij}$.

In seguito, si esegue un certo numero di iterazioni di *pricing* delle colonne e separazione dei tagli violati. Ad ogni iterazione, i seguenti passi vengono eseguiti:

- *MP* viene risolto con il semplice duale;
- la soluzione duale ottima di *MP* viene utilizzata per generare, per ogni origine $i \in S$, la variabile ξ_{ij}^q con il costo ridotto più negativo e, se una simile colonna esiste, aggiungerla in base;
- *MP* viene risolto con il semplice primale;

- la soluzione primale ottima di MP viene utilizzata per separare i disequazioni valide e, per ognuno dei tagli (4.32) – (4.35), la disequazione più violata è aggiunta a MP .

Ogni \mathcal{K}_C tagli aggiunti al problema, il processo smette di separare i primi e aggiunge iterativamente variabili di costo ridotto negativo fino a raggiungere un lower bound valido. Pertanto, la procedura calcola una sequenza di soluzioni di \overline{D}_3' corrispondente a lower bound non decrescenti per il FCTP.

4.5.3 Local Search

L'algoritmo in questione genera una sequenza di BFS per il FCTP, ciascuna consistente di $n + m - 1$ archi dell'insieme E e rappresentante uno *spanning tree* del grafo G . I tentativi di miglioramento della base vengono effettuati tramite ricerca di una BFS adiacente migliore, equivalente ad una operazione di pivoting dell'algoritmo del semplice.

Sia \mathbf{x}_B una BFS corrispondente alla base B e sia $E_B \subset E$ il corrispondente insieme di archi. Consideriamo un arco $\{r, s\} \in E \setminus E_B$ come candidato all'ingresso in base; esso chiuderebbe il cammino da r a s , rendendolo un ciclo; dunque, per mantenere l'ammissibilità della base, uno degli archi dell'insieme E_B appartenente al cammino da r a s deve lasciare la base.

Sia $P_{rs}^B \subset E_B$ il cammino dall'origine r alla destinazione s nella BFS \mathbf{x}_B e siano $O_{rs}^B \subseteq P_{rs}^B$ e $E_{rs}^B \subseteq P_{rs}^B$ i sottoinsiemi degli archi appartenenti al cammino in posizioni dispari e pari, rispettivamente. Al fine di mantenere l'ammissibilità primale, l'arco $\{r, s\}$ deve entrare in base con un valore λ_{rs}^B definito come

$$\lambda_{rs}^B = \min_{\{i,j\} \in O_{rs}^B} \{x_{ij}\}$$

e l'arco dell'insieme O_{rs}^B che determina tale valore deve lasciare la base. L'apporto alla funzione obiettivo se tale scambio venisse eseguito è calcolato come

$$\Delta_{rs}^B = f_{rs} + \lambda_{rs}^B \left(c_{rs} - \sum_{\{i,j\} \in O_{rs}^B} c_{ij} + \sum_{\{i,j\} \in E_{rs}^B} c_{ij} \right) - \sum_{\{i,j\} \in O_{rs}^B: x_{ij} = \lambda_{rs}^B} f_{ij} + \sum_{\{i,j\} \in E_{rs}^B: x_{ij} = 0} f_{ij} \quad (4.36)$$

L'arco $\{r, s\} \in E \setminus E_B$ che permette il Δ_{rs}^B più negativo è scelto per l'ingresso in base.

4.5.4 Restart Phase

Una volta raggiunto un minimo locale, la seguente procedura viene applicata al fine di raggiungere una nuova BFS per riprendere la ricerca di un upper bound.

Per ogni arco $\{i, j\} \in E$ e ogni possibile flusso q , $1 \leq q \leq m_{ij}$, si definisca η_{ij}^q come il costo ridotto medio della variabile ξ_{ij}^q rispetto alle h^* soluzioni \overline{w}_3' , ottenute durante la procedura di lower bounding, calcolato come segue

$$\eta_{ij}^q = \frac{1}{h^*} \sum_{h=1}^{h^*} \left(d_{ij}^q - qu_i^h - qv_j^h - \sum_{(k,\sigma) \in \mathcal{S}: k=i} \left(\left\lfloor \frac{q}{\sigma} \right\rfloor \alpha_{k\sigma}^h + \left\lceil \frac{q}{\sigma} \right\rceil \gamma_{k\sigma}^h \right) - \sum_{(k,\sigma) \in \mathcal{T}: k=j} \left(\left\lfloor \frac{q}{\sigma} \right\rfloor \beta_{k\sigma}^h + \left\lceil \frac{q}{\sigma} \right\rceil \delta_{k\sigma}^h \right) \right) \quad (4.37)$$

Definiamo, inoltre, per ogni arco $\{i, j\} \in E$, un valore η_{ij}^0 calcolato come

$$\eta_{ij}^0 = \frac{2 \sum_{q=1}^{m_{ij}} q \eta_{ij}^q}{m_{ij}(m_{ij} + 1)} \quad (4.38)$$

La *restart phase* parte da una BFS $x_B = [x_{i_1 j_1}, \dots, x_{i_\rho j_\rho}]$, con $\rho = n + m - 1$, corrispondente ad un minimo locale e genera una nuova BFS $x_{B'}$ attraverso i passi seguenti.

Sia E_B l'insieme degli archi corrispondenti alla BFS x_B . In primis, si definisca un insieme $E^{in} \subset E_B$ di archi eleggibili ad entrare in base.

Poi, la *restart phase* esegue un determinato numero di iterazioni, a ciascuna delle quali due TP vengono risolti, il primo atto a selezionare un altro sottoinsieme di archi eleggibili ad entrare in base, insieme con l'insieme E^{in} , il secondo a ricercare una BFS che utilizzi solo gli archi precedentemente scelti come eleggibili.

Se la BFS generata non corrisponde ad un minimo locale, la procedura termina e l'euristico la ricerca dalla BFS corrente. Altrimenti, si esegue una nuova iterazione modificando la funzione obiettivo del primo problema, impedendo ad alcune variabili di entrare a far parte della base.

Se tutte le BFS ottenute con l'insieme iniziale E^{in} sono minimi locali, un nuovo insieme E^{in} viene generato fino a quando una BFS non corrispondente ad un

minimo viene trovata o tutti i possibili insiemi E^{in} sono stati esplorati. In quest'ultimo caso, viene impiegato un algoritmo greedy per trovare una BFS.

4.5.5 L'euristico

L'intera procedura euristica può essere riassunta tramite i seguenti passi. Denotiamo con t il contatore delle iterazioni, con $x_B^{(t)}$ la BFS corrente e con UB il miglior upper bound raggiunto corrispondente alla soluzione x_B^* del FCTP.

- (1) Eseguire la procedura di Lower Bounding.
- (2) Calcolare il vettore η tramite le espressioni (4.37) e (4.38).
- (3) Risolvere il problema P' . Sia x_B la BFS corrispondente alla soluzione ottima trovata. Impostare $t = 0$, $UB = \infty$ e $x_B^{(t)} = x_B$.
- (4) Applicare iterativamente la procedura di Local Search per migliorare la BFS $x_B^{(t)}$ fino a raggiungere un minimo locale.
- (5) Sia $c(x_B^{(t)})$ il costo della BFS $x_B^{(t)}$. Se $c(x_B^{(t)}) < UB$, allora $UB = c(x_B^{(t)})$ e $x_B^* = x_B^{(t)}$.
- (6) Se si è raggiunto il limite di tempo, terminare l'algoritmo.
- (7) Applicare la *restart procedure* per trovare una nuova BFS $x_B^{(t+1)}$ partendo da $x_B^{(t)}$.
- (8) $t = t + 1$. Tornare allo step 4.

4.5.6 Risultati computazionali

In questa sezione, riportiamo alcuni dei risultati ottenuti tramite l'algoritmo RCILS di Buson et al. (2013). La scelta di presentare tali risultati per questo algoritmo e non per tutti gli altri deriva dal fatto che esso rappresenta lo stato dell'arte degli euristici per il FCTP, i quali sono l'oggetto principale della tesi.

Le istanze delle quali riportiamo menzione sono 12 piccole e (relativamente) semplici istanze di FCTP, generate da Sun et al. (1998), sulle quali sono stati testati sia il presente algoritmo, sia il metodo di Glover et al. (2005), sia il *Tabu Search* di Sun et al. (1998), il quale verrà affrontato nel prossimo capitolo.

Sei dimensioni, 10×10 , 15×15 , 10×20 , 10×30 , 50×50 e 30×100 , vengono prese in considerazione, due istanze per ciascuna dimensione; la soluzione

4.5 Il metodo euristico di Buson, Roberti e Toth (2013)

ottima delle prime 8 istanze (con $n = 10$ o 15), è nota. I risultati (RCILS) sono stati confrontati con quelli ottenuti da Sun et al. (1998), (SAMD), Glover et al. (2005), (GAK), e con il solver MILP Cplex 12.5 (CPX) applicato alla formulazione P con tempo limite 300 secondi. La tabella, inoltre, riporta il lower bound raggiunto dall'esatto di Roberti et al. (2013), (RBM), presentato nel paragrafo 3.3, al nodo radice dell'albero di *branch-and-price* con un massimo di 1000 tagli.

| Inst | Size | RCILS | | | | SAMD | | | GAK | | | CPX | | | RBM | | | |
|-------|--------|--------|--------|----------|---------------|----------|--------------|----------|-----------|--------------|----------|-----------|--------|----------|--------------|----------|-----------|--------|
| | | z(LF1) | lb | t_{lb} | ub | t_{ub} | ub | t_{ub} | t_{tot} | ub | t_{ub} | t_{tot} | lb | t_{lb} | ub | t_{ub} | t_{tot} | lb |
| n104 | 10×10 | 39633 | 40001 | 1 | 40258 | 1 | 40258 | 0 | 0 | 40258 | 0 | 0 | 40231 | 1 | 40258 | 1 | 1 | 40255 |
| n107 | 10×10 | 41440 | 41721 | 1 | 42029 | 1 | 42029 | 0 | 0 | 42029 | 0 | 0 | 41945 | 1 | 42029 | 1 | 1 | 42010 |
| n204 | 15×15 | 53596 | 54082 | 2 | 54502 | 2 | 54502 | 0 | 0 | 54502 | 0 | 1 | 54370 | 1 | 54502 | 1 | 1 | 54453 |
| n207 | 15×15 | 52649 | 53165 | 3 | 53596 | 3 | 53601 | 0 | 0 | 53601 | 0 | 1 | 53390 | 1 | 53596 | 2 | 2 | 53470 |
| n304 | 10×20 | 55633 | 55929 | 1 | 56366 | 1 | 56391 | 0 | 0 | 56366 | 0 | 0 | 56324 | 1 | 56366 | 1 | 1 | 56365 |
| n307 | 10×20 | 49002 | 49312 | 1 | 49742 | 1 | 49742 | 0 | 0 | 49742 | 0 | 0 | 49548 | 1 | 49742 | 1 | 1 | 49712 |
| n504 | 10×30 | 56383 | 56602 | 1 | 57130 | 1 | 57130 | 0 | 0 | 57130 | 1 | 2 | 56877 | 1 | 57130 | 7 | 30 | 57092 |
| n507 | 10×30 | 52165 | 52383 | 1 | 52903 | 1 | 52977 | 0 | 0 | 52903 | 1 | 1 | 52677 | 1 | 52903 | 4 | 7 | 52866 |
| n1004 | 50×50 | 161240 | 162364 | 35 | 163558 | 48 | 163793 | 4 | 5 | 163585 | 4 | 22 | 162578 | 5 | 163817 | 119 | 300 | 162774 |
| n1007 | 50×50 | 159659 | 160849 | 69 | 162108 | 261 | 162313 | 1 | 6 | 162237 | 26 | 35 | 160917 | 5 | 162352 | 225 | 300 | 161310 |
| n2004 | 30×100 | 102249 | 102717 | 3 | 103892 | 16 | 104193 | 0 | 15 | 104001 | 36 | 39 | 102994 | 4 | 104140 | 226 | 300 | 103437 |
| n2007 | 30×100 | 102466 | 102926 | 3 | 104003 | 110 | 104341 | 0 | 13 | 104256 | 5 | 27 | 103223 | 4 | 104257 | 176 | 300 | 103617 |
| Avg | | 98.51 | 99.15 | 10 | 100.00 | 37 | 100.09 | 0 | 3 | 100.04 | 6 | 11 | 99.57 | 2 | 100.07 | 64 | 104 | 99.80 |
| Best | | | | | 12 | | 5 | | | 7 | | | | | 8 | | | |

Tabella 4.1: Istanze di Sun et al., 1998: risultati dettagliati sulle 12 istanze di piccole dimensioni.

Come si vede dalla tabella soprastante, reperita da Buson et al. (2013), l'euristico *RCILS* (*Reduced Cost Iterated Local Search*) fornisce la soluzione ottima al problema nelle prime 8 istanze (i.e. quelle la cui soluzione ottima è nota) e un upper bound migliore degli altri euristici per le restanti. Al lettore viene consigliata, in caso sia interessato, la lettura del paper di Buson, Roberti e Toth, per informazioni dettagliate circa le restanti istanze testate.

Ciò che appare dai risultati presenti nella tabella di cui sopra, nonché nelle altre presentate dagli autori, è la prova dell'efficienza computazionale dell'idea chiave di guidare la fase di perturbazione/restart con le soluzioni duali corrispondenti ai lower bound raggiunti e con i costi ridotti variabili. Tale euristico, infatti, è stato in grado di raggiungere la soluzione ottima nelle 8 istanze per le quali essa è nota e ottenere un miglior upper bound su tutte, tranne una, le restanti 124 istanze di test, in un tempo di computazione nell'ordine di pochi minuti. Il confronto con Glover et al.

Capitolo 4. Metodi di risoluzione euristici

(2005), e Sun et al. (1998), mostra che gli upper bound forniti dall'algoritmo sono, in media, migliori dello 0,74% e 1,44%.

5. Tabu Search

Particolare attenzione, durante lo svolgimento di questa tesi, è stata posta sull'euristico *Tabu Search* sviluppato da Sun et al. (1998). Tale algoritmo è stato di fatto studiato, implementato e testato su istanze di FCTP con fino a 30 origini e 100 destinazioni, al fine di valutare possibili aggiunte future atte a migliorarne le prestazioni. In seguito vengono descritti i concetti fondamentali utili al fine della comprensione dell'algoritmo, viene fornita una visione d'insieme dell'intero processo e vengono discussi dettagli implementativi e risultati computazionali.

Questo euristico si basa su una ricerca locale con criteri di esclusione delle mosse (*tabu conditions*), mirati a evitare il ripetersi eccessivo di scambi delle stesse coppie di archi, articolata su tre processi *memory based* (basati su memorie rispettivamente a breve, medio e lungo termine). In particolare, le memorie presenti sono di due tipi: *frequency based*, ossia basate sulla frequenza di ingresso / uscita degli archi dalla base, e *recency based*, ossia basate sul tempo di permanenza dentro / fuori la base). Inoltre, viene adottato un criterio di ignoramento delle *tabu conditions* fondato sul concetto di livello di aspirazione, proposto da Lewin et al. (1944), il quale indica il grado o la qualità della performance che una persona desidera o crede di poter raggiungere durante lo svolgimento di un compito (si è messo in evidenza che il livello di aspirazione aumenta dopo ripetuti successi nella stessa prestazione e diminuisce dopo ripetuti insuccessi).

5.1 Concetti fondamentali

Nei paragrafi che seguono vengono introdotti e illustrati i concetti fondamentali presenti nell'euristico in questione. Senza perderci in digressioni ripetitive, comunichiamo al lettore, e pertanto assumiamo, da qui in poi, che tale nozione sia acquisita, che il presente algoritmo si basa sulla formulazione matematica classica P del problema e sul relativo rilassamento lineare P' .

5.1.1 Mossa

Con il termine *mossa* si intende il passaggio da una soluzione base ammissibile ad un'altra soluzione base ammissibile adiacente. Dunque, una mossa equivale ad una iterazione o una operazione di pivoting del metodo del simpleso applicato ad un grafo.

Al fine dell'implementazione dei processi basati sulle memorie, vengono utilizzati tre array, di lunghezza $N = n \times m$, z , t e h , indicanti, rispettivamente, il valore della funzione obiettivo subito prima che un dato arco entri od esca dalla base.

5.1.2 Soluzioni adiacenti e pricing

Data una soluzione base ammissibile, un qualsiasi arco $(i, j) \notin \mathcal{T}$, ove \mathcal{T} rappresenta l'insieme degli archi in base, può entrare in base per passare ad una soluzione base ammissibile adiacente. Pertanto, con *neighborhood* (i.e. vicinato) di una soluzione base ammissibile si intendono tutte le soluzioni base ammissibili raggiungibili tramite l'ingresso in base di un singolo arco $(i, j) \notin \mathcal{T}$.

Supponendo che $(i, j) \notin \mathcal{T}$ sia l'arco candidato ad entrare in base, definiamo con $P \subset \mathcal{T}$ l'insieme ordinato degli archi che costituiscono un cammino da i a j . Così come nell'algoritmo del simpleso applicato al flusso in una rete, l'arco uscente è uno di quegli archi $(r, s) \in P$ tale per cui

$$\delta = \min\{x_{pq} \mid (q, p) \in P\} \quad (5.1)$$

Tale nozione era già stata presentata, ed è pertanto equivalente, nel paragrafo 4.3.3, a proposito della ricerca di una BFS adiacente nell'euristico *RCILS*, sviluppato da Buson et al. (2013). Nel caso in cui $\delta = 0$ (i.e. degenerazione), si sceglie $(r, s) \in P$ come l'arco uscente con il massimo costo fisso tra tutti gli archi eleggibili ad uscire dalla base.

La procedura di pricing adottata calcola, data la coppia di archi $\{(i, j), (r, s)\}$ così scelti, l'apporto dell'eventuale esecuzione della mossa alla funzione obiettivo come

$$\Delta_{ij} = \begin{cases} 0 & \text{se } \delta = 0 \\ \delta(c_{ij} - \pi_i + \pi_j) + f_{ij} - \sum_{(p,q) \in U_1} f_{pq} + \sum_{(p',q') \in U_2} f_{p',q'} & \text{se } \delta > 0 \end{cases} \quad (5.2)$$

ove U_1 sono tutti gli archi il cui flusso decremterà da positivo a 0, U_2 tutti quegli archi il cui flusso incremterà da 0 a positivo, π_i e π_j le variabili duali, definite esclusivamente tramite i costi variabili, corrispondenti ai nodi i e j , rispettivamente. Tra tutti gli archi candidati ad entrare in base, viene scelto il nodo (i, j) per cui

$$\Delta_{ij} = \min\{\Delta_{pq} \mid (p, q) \notin \mathcal{T}\}$$

Nel caso in cui $\Delta_{ij} \geq 0$ per ogni $(i, j) \notin \mathcal{T}$, la soluzione base ammissibile corrente è un minimo locale. Si noti che non è necessario che $\Delta_{ij} < 0$ affinché l'arco (i, j) sia scelto per entrare in base. In certi casi, infatti, è necessario effettuare una mossa che non migliori la base al fine di uscire da una situazione di stallo in corrispondenza di un minimo locale.

5.1.3 Condizioni tabu e livello di aspirazione

Due sono le *tabu conditions* adottate nel corso dell'algoritmo. Esse si basano sul criterio di permanenza dentro o fuori la base della coppia di archi valutata per l'aggiornamento della soluzione base ammissibile.

Più in particolare, definiamo due variabili, $7 \leq \xi_{in} \leq 10$ e $2 \leq \xi_{out} \leq 4$, le quali, come vedremo in seguito nel paragrafo 5.2, vengono, ad ogni iterazione completa, aggiornate scegliendone il valore, nell'intervallo relativo, in maniera casuale. Dunque, riguardo alla mossa per cui $(i, j) \notin \mathcal{T}$ rimpiazzerebbe $(r, s) \in \mathcal{T}$, queste condizioni sono definite tramite l'array t come

$$k - t_{ij} \leq \xi_{in} \vee k - t_{rs} \leq \xi_{out} \quad (5.3)$$

ove k indica il numero dell'iterazione corrente. Se tale condizione risulta soddisfatta, la mossa è considerata *tabu* e non viene pertanto effettuata.

Le condizioni (5.3) considerano tabu una mossa, in breve, nel caso in cui l'arco (i, j) sia uscito dalla base troppo di recente o l'arco (r, s) sia entrato in base troppo di recente.

Il *livello di aspirazione*, invece, è un criterio utilizzato per decidere se ignorare o meno le *tabu conditions*, nel caso una soluzione migliore venga trovata. Come prima, considerando la mossa per cui $(i, j) \notin \mathcal{T}$ rimpiazzerebbe $(r, s) \in \mathcal{T}$, il livello di aspirazione viene implementato tramite l'array z come

$$z + \Delta_{ij} < \min\{z_{ij}, z_{rs}\} \quad (5.4)$$

Nel caso in cui tale condizione risulti vera, le *tabu conditions* vengono ignorate e la mossa eseguita.

5.1.4 Intensificazione, diversificazione e criterio di arresto

Per implementare il processo di memoria a medio termine (*intermediate memory process*), vengono adottate due strategie diverse.

Nella prima strategia, il processo di memoria a medio termine è invocato periodicamente modificando la misura dell'attrattività di una mossa da Δ_{ij} a

$$\Delta'_{ij} = \Delta_{ij} - \alpha \frac{h_{ij}}{k} \quad (5.5)$$

ove $\alpha > 0$ è una costante atta ad assicurare che $\alpha \frac{h_{ij}}{k}$ sia significativa, ma non dominante in Δ'_{ij} . Nell'implementazione dell'algoritmo, il valore di α è calcolato alla prima soluzione ottima locale come

$$\alpha = \frac{1}{N - n - m} \sum_{(i,j) \notin \mathcal{T}} \Delta_{ij} \quad (5.6)$$

Con Δ'_{ij} come misura dell'attrattività di una mossa, risultano più propense ad essere eseguite quelle mosse che portano in base archi $(i,j) \notin \mathcal{T}$ che più frequentemente entrano in base.

Durante la seconda fase del processo di memoria a medio termine, invece, si permette alla ricerca di "scivolare" verso un minimo locale.

Il processo di memoria a lungo termine, invece, è atto a portare la ricerca in una nuova regione non ancora esplorata, così da diversificare la ricerca. Per far ciò vengono applicate, come per il processo a medio termine, due diverse strategie.

Nella prima, la misura dell'attrattività di una mossa viene modificata da Δ_{ij} a

$$\Delta''_{ij} = \Delta_{ij} - \alpha \left(1 - \frac{h_{ij}}{k}\right) \quad (5.7)$$

così da far apparire più attraenti quelle mosse che portano in base archi $(i,j) \notin \mathcal{T}$ che raramente entrano in base. Ogni volta che il processo viene eseguito, vengono effettuate n mosse, una per ciascuna origine.

Nella seconda strategia, invece, per ogni origine viene introdotto in base l'arco che per più tempo ne è rimasto fuori, ossia quell'arco $(i, j) \notin \mathcal{T}$ per cui

$$t_{ij} = \min\{t_{iq} \mid (i, q) \notin \mathcal{T}, 1 \leq q \leq m\} \quad (5.8)$$

Il criterio di arresto utilizzato si basa su tre coefficienti i cui valori sono determinati a priori: L , β e γ . Sia $\beta > 0$ un coefficiente tale per cui il processo a medio termine è invocato se non è stata trovata alcuna soluzione per βN mosse. Sia $\gamma > 0$ un coefficiente tale per cui la prima strategia del processo a medio termine è iterata γN volte ogni volta. I due processi a lungo termine vengono invocati alternatamente dopo la seconda strategia a medio termine. Contiamo il processo a lungo termine come eseguito una volta quando entrambe le strategie sono state applicate. Quando il processo a lungo termine viene invocato L volte, la ricerca si interrompe.

Nell'implementazione, i valori utilizzati per questi tre coefficienti sono $L = 5$, $\beta = 0,5$ e $\gamma = 0,5$.

5.1.5 Soluzione base ammissibile di partenza

La soluzione base ammissibile di partenza è ottenuta tramite la risoluzione lineare del rilassamento P' proposto da Balinsky (1961). Una volta ottenuta la soluzione x' , il problema originale è inizializzato ponendo $x_{ij} = x'_{ij}$ e $y_{ij} = 1$ se $x_{ij} > 0$ (0 altrimenti).

A partire da questa soluzione base ammissibile di partenza, Δ_{ij} è calcolato per ogni $(i, j) \notin \mathcal{T}$ e (i, j) viene portato in base se $\Delta_{ij} < 0$. Quando $\Delta_{ij} \geq 0$ per ogni $(i, j) \notin \mathcal{T}$, la prima *soluzione ottima locale* è stata raggiunta.

5.2 L'algoritmo

5.2.1 Inizializzazione

- (0) Ricercare una *soluzione ottima locale* con valore della funzione obiettivo z e calcolare α . Impostare il contatore delle iterazioni $k = 0$; impostare $l = 0$, $l_1 = 0$. Impostare $h_{ij} = 0$, $z_{ij} = z$ e $t_{ij} = -\infty$ per ogni i e j . Sia $i = 0$.

5.2.2 Processo di memoria a breve termine

- (1) $i = i + 1$. Se $i > n$, impostare $i = 1$. Calcolare $\Delta_{iq}, \forall q = 1, \dots, m \wedge (i, q) \notin \mathcal{T}$.
- (2) Scegliere (i, j) come arco entrante tale che $\Delta_{ij} = \min\{\Delta_{iq} | (i, q) \notin \mathcal{T} \wedge 1 \leq q \leq m\}$, con il corrispondente arco uscente (r, s) . Se $\Delta_{ij} = \infty$ ritornare allo step 1.
- (3) Se $k - t_{ij} \leq \xi_{in}$ o $k - t_{rs} \leq \xi_{out}$ andare allo step 4. Altrimenti andare allo step 5.
- (4) Se $z + \Delta_{ij} > z_{ij}$ o $z + \Delta_{ij} > z_{rs}$ impostare $\Delta_{ij} = \infty$ e tornare allo step 2. Altrimenti andare allo step 5.
- (5) Aggiornare lo *spanning tree*. Impostare $h_{rs} = h_{rs} + (k - t_{rs})$, $t_{ij} = t_{rs} = k$, $z_{ij} = z_{rs} = z$. Impostare $k = k + 1$ e $z = z + \Delta_{ij}$.
- (6) Se $z < z_{min}$, impostare $z_{min} = z$ e $k_{min} = k$. Se $z < z_{best}$, impostare $z_{best} = z$ e salvare la soluzione corrente come \mathbf{x}_0 .
- (7) Se $k - k_{min} < \beta N$, tornare allo step 1. Altrimenti, se $\beta N \leq k - k_{min} < (\beta + \gamma)N$, andare allo step 8. Altrimenti, andare allo step 12.

5.2.3 Processo di memoria a medio termine 1

- (8) $i = i + 1$. Se $i > n$, impostare $i = 1$. Calcolare $\Delta'_{iq}, \forall q = 1, \dots, m \wedge (i, q) \notin \mathcal{T}$.
- (9) Scegliere (i, j) come arco entrante tale che $\Delta'_{ij} = \min\{\Delta'_{iq} | (i, q) \notin \mathcal{T} \wedge 1 \leq q \leq m\}$, con il corrispondente arco uscente (r, s) . Se $\Delta'_{ij} = \infty$ ritornare allo step 8.
- (10) Se $k - t_{ij} \leq \xi_{in}$ o $k - t_{rs} \leq \xi_{out}$ andare allo step 11. Altrimenti ritornare allo step 5.
- (11) Se $z + \Delta_{ij} > z_{ij}$ o $z + \Delta_{ij} > z_{rs}$ impostare $\Delta'_{ij} = \infty$ e tornare allo step 9. Altrimenti ritornare allo step 5.

5.2.4 Processo di memoria a medio termine 2

- (12) Impostare $(i_0, j_0) = (i, m)$.
- (13) $i = i + 1$. Se $i > n$, impostare $i = 1$.
- (14) Per $j = 1, \dots, m$, ripetere i passi a-d:
 - a. Calcolare Δ_{ij} .

- b. Se $\Delta_{ij} \geq 0$ e $(i, j) = (i_0, j_0)$, andare allo step 16.
 - c. Se $(i, j) \notin \mathcal{T}$ e $\Delta_{ij} < 0$, impostare $(i_0, j_0) = (i, j)$ e far entrare (i, j) nello *spanning tree* con un corrispondente arco uscente (r, s) .
 - d. Aggiornare lo *spanning tree*. Impostare $h_{rs} = h_{rs} + (k - t_{rs})$, $t_{ij} = t_{rs} = k$, $z_{ij} = z_{rs} = z$. Impostare $k = k + 1$ e $z = z + \Delta_{ij}$.
- (15) Tornare allo step 13.
- (16) Se $z < z_{best}$, impostare $z_{best} = z$ e salvare la soluzione corrente come \mathbf{x}_0 . Se $l \geq L$, Terminare l'algoritmo. Altrimenti, impostare $i_0 = i$. Se $l_1 = 0$, impostare $l_1 = 1$ e andare allo step 17. Altrimenti, impostare $l_1 = 0$ e andare allo step 23.

5.2.5 Processo di memoria a lungo termine 1

- (17) $i = i + 1$. Se $i > n$, impostare $i = 1$. Calcolare $\Delta''_{iq}, \forall q = 1, \dots, m \wedge (i, q) \notin \mathcal{T}$.
- (18) Scegliere (i, j) come arco entrante tale che $\Delta''_{ij} = \min\{\Delta''_{iq} | (i, q) \notin \mathcal{T} \wedge 1 \leq q \leq m\}$, con il corrispondente arco uscente (r, s) . Se $\Delta''_{ij} = \infty$, tornare allo step 17.
- (19) Se $k - t_{ij} \leq \xi_{in}$ o $k - t_{rs} \leq \xi_{out}$ andare allo step 20. Altrimenti andare allo step 21.
- (20) Se $z + \Delta_{ij} > z_{ij}$ o $z + \Delta_{ij} > z_{rs}$ impostare $\Delta''_{ij} = \infty$ e tornare allo step 18. Altrimenti andare allo step 21.
- (21) Aggiornare lo *spanning tree*. Impostare $h_{rs} = h_{rs} + (k - t_{rs})$, $t_{ij} = t_{rs} = k$, $z_{ij} = z_{rs} = z$. Impostare $k = k + 1$ e $z = z + \Delta_{ij}$.
- (22) Se $i = i_0$, impostare $z_{min} = z$, $k_{min} = k$, resettare i valori di ξ_{in} e ξ_{out} e tornare allo step 1. Altrimenti, tornare allo step 17.

5.2.6 Processo di memoria a lungo termine 2

- (23) $i = i + 1$. Se $i > n$, impostare $i = 1$. Scegliere (i, j) come arco entrante, tale che $t_{ij} = \min\{t_{iq} | (i, q) \notin \mathcal{T} \wedge 1 \leq q \leq m\}$ con il corrispondente arco uscente (r, s) e calcolare Δ_{ij} .
- (24) Aggiornare lo *spanning tree*. Impostare $h_{rs} = h_{rs} + (k - t_{rs})$, $t_{ij} = t_{rs} = k$, $z_{ij} = z_{rs} = z$. Impostare $k = k + 1$ e $z = z + \Delta_{ij}$.

- (25) Se $i = i_0$, impostare $l = l + 1$, $z_{min} = z$, $k_{min} = k$, resettare i valori di ξ_{in} e ξ_{out} e tornare allo step 1. Altrimenti tornare allo step 23.

5.3 Implementazione e risultati computazionali

Oggetto primario della presente tesi è stato lo studio, e successivamente l'implementazione, dell'algoritmo appena descritto al fine di considerare eventuali miglioramenti dello stesso.

Una modifica, risultata valida nelle prestazioni conseguite, è stato l'utilizzo per il calcolo dell'apporto alla funzione obiettivo della formula (4.36), anziché della formula (5.2). Tale scelta ha portato all'ottenimento di un upper bound iniziale migliore rispetto a quello conseguito nella versione originale.

Inoltre, è stato identificato un errore, corretto nella versione qui presentata, nel passo 14 del secondo processo di memoria a lungo termine. In tale passo, per ogni origine $i \in S$, viene portato in base ogni arco (i, j) per cui $\Delta_{ij} < 0$. Il ciclo termina quando, definito (i_0, j_0) come l'ultimo arco portato in base, tale arco viene raggiunto nuovamente dal processo. In questo caso, si prosegue verso i processi di memoria a lungo termine.

Nel metodo originale, però, si controlla che (i, j) non sia in base prima di eseguire il controllo $(i, j) = (i_0, j_0)$, il quale, dunque, non potrebbe mai essere soddisfatto. Supponiamo, infatti, di portare in base (i_0, j_0) : se controlliamo tutti i restanti archi non base del problema senza portarne nessuno in base, una volta raggiunto nuovamente (i_0, j_0) , questo risulterà, trivialmente, ancora in base. Pertanto, secondo la sequenza del metodo originale, il ciclo non terminerebbe, ma si protrarrebbe (virtualmente) all'infinito.

Nella tabella 5.1 sono presentati i risultati dei test effettuati sulle 12 istanze originali (cfr. paragrafo 4.3.6). Nella colonna (TSH) sono presenti gli upper bound raggiunti da tale implementazione, messi a confronto con gli upper bound raggiunti dagli euristici di Buson et al. (2013), nella colonna (RCILS), Glover et al. (2005), nella colonna (GAK), dal metodo originale di Sun et al. (1998), nella colonna (SAMD) e con il solver MILP Cplex 12.5 applicato alla formulazione standard P con un tempo limite di 300 secondi, nella colonna (CPX). Per fornire un ulteriore confronto, sono stati aggiunti, nella colonna (RBM), i lower bound raggiunti

5.3 Implementazione e risultati computazionali

dall'esatto di Roberti et al. (2013) al nodo radice dell'albero di *branch-and-price* con un massimo di 1000 tagli.

Nella fase di test non si è prestata particolare attenzione ai tempi computazionali, quanto piuttosto ai risultati raggiunti. Per ogni metodo vengono presentati, nella riga *Media*, il gap percentuale medio rispetto al miglior upper bound raggiunto e il tempo medio di risoluzione. Nella riga *Best*, invece, è indicato il numero di istanze per le quali, ogni metodo, ha raggiunto il miglior upper bound. L'algoritmo è stato sviluppato in C utilizzando Visual Studio 2010 64-bit.

| Inst. | Size | TSH | | RCILS | | SAMD | | GAK | | CPX | | RBM |
|--------------|--------|--------------|------|---------------|------|--------------|------|--------------|------|--------------|------|--------|
| | | ub | time | ub | time | ub | time | ub | time | ub | time | ub |
| n104 | 10x10 | 40258 | 0 | 40258 | 1 | 40258 | 0 | 40258 | 0 | 40258 | 1 | 40255 |
| n107 | 10x10 | 42029 | 1 | 42029 | 1 | 42029 | 0 | 42029 | 0 | 42029 | 1 | 42010 |
| n204 | 15x15 | 54502 | 1 | 54502 | 2 | 54502 | 0 | 54502 | 1 | 54502 | 1 | 54453 |
| n207 | 15x15 | 53596 | 1 | 53596 | 3 | 53601 | 0 | 53601 | 1 | 53596 | 2 | 53470 |
| n304 | 10x20 | 56366 | 1 | 56366 | 1 | 56391 | 0 | 56366 | 0 | 56366 | 1 | 56365 |
| n307 | 10x20 | 49742 | 1 | 49742 | 1 | 49742 | 0 | 49742 | 0 | 49742 | 1 | 49712 |
| n504 | 10x30 | 57130 | 3 | 57130 | 1 | 57130 | 0 | 57130 | 2 | 57130 | 30 | 57092 |
| n507 | 10x30 | 52903 | 3 | 52903 | 1 | 52977 | 0 | 52903 | 1 | 52903 | 7 | 52866 |
| n1004 | 50x50 | 163719 | 305 | 163558 | 48 | 163793 | 5 | 163585 | 22 | 163817 | 300 | 162774 |
| n1007 | 50x50 | 162274 | 298 | 162108 | 261 | 162313 | 6 | 162237 | 35 | 162352 | 300 | 161310 |
| n2004 | 30x100 | 104047 | 758 | 103892 | 16 | 104193 | 15 | 104001 | 39 | 104140 | 300 | 103437 |
| n2007 | 30x100 | 104246 | 803 | 104003 | 110 | 104341 | 13 | 104256 | 27 | 104257 | 300 | 103617 |
| Media | | 0,05% | 181 | 0,00% | 37 | 0,09% | 3 | 0,04% | 11 | 0,07% | 104 | |
| Best | | 8 | | 12 | | 5 | | 7 | | 8 | | |

Tabella 5.1: risultati dei test sulle istanze di Sun et al., 1998.

Come è possibile osservare dai risultati ottenuti, il *Tabu Search* implementato raggiunge la soluzione ottima nei primi 8 problemi allo stesso modo dell euristico RCILS e di Cplex 12.5. Nei restanti 4 problemi, invece, TSH non raggiunge upper bound buoni come quelli di RCILS. Tuttavia, ottiene risultati migliori dell'originale SAMD, CPX e, in due istanze soltanto, anche di GAK.

6. Conclusioni

In questa tesi è stato studiato un metodo metaeuristico Tabu Search per la risoluzione del *Problema dei Trasporti con Costi Fissi*, noto come FCTP. Questo problema è *NP-Hard* e ha molteplici applicazioni pratiche nel settore della supply chain.

Nella tesi sono state presentate e discusse le principali formulazioni matematiche proposte in letteratura e i metodi esatti ed euristici di soluzione.

È stato descritto nel dettaglio il metaeuristico *Tabu Search* di Sun et al. (1998) e le modifiche proposte per migliorare le prestazioni.

È stato realizzato un codice prototipale per valutare le prestazioni del nuovo euristico su un set di problemi test.

I risultati computazionali ottenuti mostrano che le modifiche apportate migliorano le prestazioni dell'algoritmo originale e che i risultati conseguiti sono competitivi con il miglior euristico fin ora proposto per il FCTP.

Un limite del metodo proposto è l'elevato tempo di calcolo che può essere drasticamente ridotto reingegnerizzando il codice stesso ma anche migliorando le strategie di ricerca locale, ad esempio, estendendo la ricerca a soluzioni base non adiacenti a quella corrente al fine di evitare i minimi locali che, troppo spesso, l'attuale strategia raggiunge.

Bibliografia

Adlakha, V. e Kowalsky, K., *A simple heuristic for solving small fixed-charge transportation problems*, 2003. Omega, **31**(3) 205-211.

Agarwal, Y. e Aneja, Y., *Fixed-charge transportation problem: Facets of the projection polyhedron*, 2012. Operations Research, **60**(3) 638-654.

Agarwal, Y., *K-partition-based facets of the network design problem*, 2006. Networks, **47**(3) 123-139.

Aguado, J. S., *Fixed charge transportation problems: a new heuristic approach base on lagrangean relaxation and the solving of core problems*, 2009. Annuals of Operations Research, **172**(1) 45-69.

Aneja, Y., *On a class of set covering problems*, 1974. Ph.D. thesis, Johns Hopkins University, Baltimore.

Balas, E. e Padberg, M. W., *Set Partitioning: A Survey*, 1976. SIAM Review, **18** 716-760.

Balinsky, M. L., *Fixed cost transportation problem*, 1961. Naval Research Logistics Quarterly, **8** 41-54.

Barr, R. S., Glover, F., Klingman, D., *A new optimization method for large scale fixed charge transportation problems*, 1980. Operations Research, **29**(3) 448-463.

Bittner, L. e Juttler, H., *A variant of the simplex method with simultaneous interchange of several basis variables per iteration step*, 1962. Unternehmensforschung, **6**(3) 137-145.

Blocher, R. H., *The Revised Duplex Algorithm for Linear Programming*, 1966. M. S. Thesis, Dept. of Applied Mathematics and Computer Science, Washington University.

Buson, E., Roberti, R., Toth, P., *A Reduced-Cost Iterated Local Search Heuristic for the Fixed Charge Transportation Problem*, 2013. Technical Reports, **OR-13-10**.

Bibliografia

Cabot, A. V. e Erenguc, S. S., *Some branch and bound procedures for fixed cost transportation problems*, 1986. Naval Research Logistic Quarterly, **31**(1) 145-154.

Cooper, L. e Drebes, C., *An approximate algorithm method for the fixed charge problem.*, 1967. Naval Research Logistics Quarterly, **14** 101-113.

Cooper, L., *The fixed charge problem - I: A new heuristic method*, 1975. Computers and Mathematics with Applications, **1**(1) 89-96.

Cornuejols, G., Fisher, L., Nemhauser, G. L., *Location of Bank Accounts to Optimize Float: An Analytic Study of Exact and Approximate Algorithms*, 1977. Management Science, **23** 789-810.

Cruz, F. R. B., MacGregor Smith, J., Mateus, G. R., *Solving to optimality the uncapacitated fixed-charge network flow problem.*, 1998. Computers and Operations Research, **25**(1) 67-81.

Drenzler, D. R., *An approximate method for the fixed charge problem*, 1969. Naval Research Logistics Quarterly, **16** 411-416.

Driebeek, N. J., *An Algorithm for the Solution of Mixed Integer Programming Problems*, 1966. Management Science, **12** 576-587.

Dutton, R., Hinman, G., Millham, C. B., *The optimal location of nuclear power facilities in the pacific northwest*, 1974. Operations Research, **22**(3) 478-487.

Dwyer, P. S., *Use of completely reduced matrices in solving transportation problems with fixed charges*, 1966. Naval Research Logistics Quarterly, **13**(3) 289-313.

El-Sherbiny, M. M. e Alhamali, R. M., *A hybrid particle swarm algorithm with artificial immune learning for solving the fixed charge transportation problem*, 2013. Computers and Industrial Engineering, **64**(2) 610-620.

Ford, L. R. e Fulkerson, D. R., *Flows in networks*, Princeton University Press, Princeton, NJ 1962.

Geoffrion, A. M. e Marsten, R. E., *Integer Programming Algorithms: A Framework and State-of-the-Art Survey*. In *Perspectives on Optimization: A Collection of Expository Articles*, Addison-Wesley, Reading

Glover, F., Amini, M., Kochenberger, G., *Parametric ghost image processes for fixed-charge problems: A study of transportation networks*, 2005. *Journal of Heuristics*, **11**(4) 307-336.

Göthe-Lundgren, M. e Larsson, T., *A set covering reformulation of the pure fixed charge transportation problem*, 1994. *Discrete Applied Mathematics*, **48**(3) 245-259.

Gray, P., *Exact solution of the fixed-charge transportation problem*, 1971. *Operations Research*, **19**(6) 1529-1538.

Gu, Z., Nemhauser, G. L., Savelsberg, M. W. P., *Lifted Cover Inequalities for 0-1 Integer Programs: Complexity*, 1999. *INFORMS Journal on Computing*, **11**(1) 117-123.

Gu, Z., Nemhauser, G. L., Savelsbergh, M. W. P., *Lifted Cover Inequalities for 0-1 Integer Programs: Computation*, 1998. *INFORMS Journal on Computing*, **10**(4) 427-437.

Guisewite, G. M. e Pardalos, P. M., *Minimum concave-cost network flow problems: Applications, complexity, and algorithms*, 1990. *Operations Research*, **25** 75-100.

Hirsch, W. M. e Dantzig, G. B., *Notes on linear programming: Part XIX, the fixed charge problem.*, 1954. *Rand Research*, **Memorandum No. 1383**.

Hirsch, W. M. e Dantzig, G. B., *The fixed charge problem*, 1968. *Naval Research Logistics Quarterly*, **15** 413-424.

Hultberg, T. H. e Cardoso, D. M., *The teacher assignment problem: A special case of the fixed charge transportation problem*, 1997. *European Journal of Operational Research*, **101**(3) 463-473.

Bibliografia

Jarvis, J. J., Unger, V. E., Rardin, R. L., Moore, R. W., *Optimal design of regional wastewater systems: A fixed charge network flow model*, 1978. Operations Research, **4**(4) 538-550.

Kennington, J. e Unger, E., *A new branch-and-bound algorithm for the fixed-charge transportation problem*, 1976. Management Science, **22**(10) 1116-1126.

Kuhn, H. W. e Baumol, W. J., *An approximate algorithm for the fixed-charge transportation problem*, 1961. Naval Research Logistics Quarterly, **9**(1) 1-16.

Lamar, B. W. e Wallace, C. A., *Revised-modified penalties for fixed-charge transportation problems*, 1997. Management Science, **43**(10) 1431-1436.

Lewin, K., Dembo, T., Festinger, L., Sears, P. S., Hunt, J. M., *Level of aspiration*. In *Personality and the behavior disorders*, Ronald Press, Oxford, England

Lotfi, M. M. e Tavakkoli-Moghaddam, R., *A genetic algorithm using priority-based encoding with new operators for fixed charge transportation problems*, 2013. Applied Soft Computing, **13**(5) 2711-2726.

Luna, H. P. L., Ziviani, N., Cabral, R. M. B., *The Telephonic Switching Center Network Problem: Formalization and Computational Experience*, 1987. Discrete and Applied Mathematics, **18** 199-210.

Martello, S. e Toth, P., *Knapsack problems*, Wiley, New York 1990.

Murty, K. G., *Solving the Fixed Charge Problem by Ranking the Extreme Points*, 1968. Operations Research, **16**(2) 268-279.

Nemhauser, G. L. e Wolsey, L. A., *Integer and Combinatorial Optimization*, Wiley-Interscience, New York, NY, USA 1999.

Nozick, L. e Turnquist, M., *Integrating Inventory Impacts into a Fixed Charge Model for Locating Distribution Centers*, 1998. Transportation Research, **31**(3) 173-186.

Nozick, L. e Turnquist, M., *Two-Echelon Inventory Allocation and Distribution Center Location Analysis*, 1998. Proceedings of Tristan III.

Ortega, F. e Wolsey, L. A., *A branch-and-cut algorithm for the single commodity, uncapacitated, fixed-charge network flow problem*, 2003. Management Science, **41**(3) 143-158.

Palekar, U. S., Karwan, M. H., Zionts, S., *A branch and bound method for the fixed charge transportation problem*, 1990. Management Science, **36**(9) 1092-1105.

Paranjape, J. R., *The simplex method: Two basic variables replacement*, 1965. Management Science, **12**(1) 137-141.

Rardin, R. L. e Wolsey, L. A., *Valid inequalities and projecting the multicommodity extended formulation for uncapacitated fixed charge network flow problems*, 1993. European Journal of Operational Research, **71**(1) 95-109.

Roberti, R., Bartolini, E., Mingozzi, A., *The Fixed Charge Transportation Problem: An Exact Algorithm Based on a New Integer Programming Formulation*, 2013. Technical Reports, **OR-13-8**.

Rothfarb, B., Frank, H., Rosembaun, D. M., Steiglitz, K., *Optimal Design of Offshore Natural-Gas Pipeline Systems*, 1970. Operation Research, **18** 992-1020.

Shetty, U. S., *A relaxation decomposition algorithm for the fixed charge network problem*, 1990. Naval Research Logistics Quarterly, **37**(2) 327-340.

Steinberg, D. I., *The fixed charge problem*, 1970. Naval Research Logistics Quarterly, **17**(2) 217-236.

Stroup, J. W., *Allocation of launch vehicles to space missions: a fixed-cost transportation problem*, 1966. Operations Research, **15**(1) 1157-1163.

Sun, M. e McKeown, P. G., *Tabu search applied to the general fixed charge problem*, 1993. Annuals of Operations Research, **41**(14) 405-420.

Sun, M., Aronson, J. E., McKeown, P. G., Drinka, D., *A tabu search heuristic procedure for the fixed charge transportation problem*, 1998. European Journal of Operational Research, **106**(2-3) 441-456.

Tomlin, J. A., *An Improved Branch and Bound Method for Integer Programming*, 1971. Operations Research, **19** 1070-1075.

Bibliografia

Walker, W. E., *A heuristic adjacent extreme point algorithm for the fixed charge problem*, 1976. *Management Science*, **22**(5) 587-596.

Wolsey, L. A., *Integer Programming*, Wiley-Interscience, New York, NY, USA 1998.

Wright, D. D. e Haehling von Lanzenuer, C., *Solving the fixed charge problem with lagrangian relaxation and cost allocation heuristics.*, 1989. *European Journal of Operational Research*, **42**(3) 305-312.

Wright, D. D. e Haehling von Lanzenuer, C., *COAL: A new heuristic approach for solving the fixed charge problem - computational results*, 1991. *European Journal of Operational Research*, **52**(2) 235-246.

Ringraziamenti

Al “quasi” Dott. Andrea Battaglia 6, per l’aiuto, la compagnia, le ore passate insieme a studiare nonché la passione trasmessami riguardo questa materia.

A Donatella, per avermi sopportato, specialmente in questi ultimi giorni di tensione, senza allontanarmi nonostante il mio insopportabile carattere e avermi sopportato sempre e comunque in ogni mia idea, anche la più stupida.

A Ippo, Chiappo, Ciccio, Chirico e Celestini (in ordine di apparizione, ma non di importanza), per aver apprezzato (parecchio) la mia cucina, per aver creduto nel lato estroverso più nascosto di me, per essere stati presenti nei momenti di bisogno e per avermi lasciato solo nei momenti giusti.

A Ippo, ancora una volta: lui sa il perché.

A mio padre e mia madre, Daniele e Maria Rosa, per essersi presi cura di me negli ultimi, quasi, 25 anni, senza mai chiedere nulla in cambio, per aver creduto in me, a volte, e avermi fatto notare il contrario nelle restanti e per essermi vicini sempre e incondizionatamente.

Infine, a mio zio Didi e a mia nonna Giuliana, per aver sempre creduto in me, per essersi sempre mostrati sinceri, per essermi stati vicini in ogni singolo momento della mia vita e per essermi ancora vicini nonostante tutto, per avermi spronato quando ne avevo bisogno, per avermi cresciuto e per mancarmi ancora tanto: per essere stati ed essere ancora le mie colonne portanti.