UNIVERSITÀ DEGLI STUDI DI MILANO

Facoltà di Scienze Matematiche, Fisiche, Naturali

Dipartimento di Tecnologie dell'Informazione

Corso di Laurea in Informatica

Polo di Crema

ALGORITMI DI OTTIMIZZAZIONE BRANCH&PRICE PER PROBLEMI DI LOCALIZZAZIONE MULTIRISORSE

Relatore: Prof. Giovanni Righini

Correlatore: Dott. Alberto Ceselli

Tesi di Laurea di

Federico Liberatore

Matricola 626150

Anno Accademico 2003/2004

Indice

1. Introduzione 4

2. Modelli 10

2.1 Single Source Capacitated Facility Location Problem 10

2.2 Capacitated Concentrator Location Problem 14

2.3 Capacitated P-Median Problem 15

2.4 Capacitated P-Concentrator Location Problem senza costi di apertura 16

2.5 Single Source Capacitated P-Facility Location Problem 17

2.6 Capacitated P-Concentrator Location Problem 18

3. Richiami teorici sul Column Generation 20

3.1 Introduzione 20

3.2 Riformulazione di Dantzig – Wolfe di un IP 21

3.3 Risolvere il Master Linear Program 22

4. Algoritmo Branch & Price 26

4.1 Bound duale 27

4.2 Column generation e rilassamento Lagrangeano 31

4.4 Euristica primale 34

4.5 Enumerazione implicita 39

4.6 Column Management 40

5. Esperimenti 43

5.1 Strumenti e dati utilizzati 43

5.2 Confronto tra Branch & Price e MIP-Solver 44

5.3 Confronto tra modelli 44

5.4 Conclusioni 46

1. Introduzione

La logistica della distribuzione di punti di servizio su larga scala ha acquistato, nel corso degli anni, interesse crescente. Data la natura combinatoria di molti problemi reali, la loro risoluzione richiede tempi di calcolo proibitivi. Ad esempio, problemi di *network design* particolarmente complessi prevedono diverse fasi di sviluppo: il *partizionamento* dell'insieme degli utenti (ad esempio clients su rete di telecomunicazione) in clusters, la *localizzazione* delle risorse (i servers) all'interno dei clusters così descritti e la progettazione del sistema di interconnessioni tra utenti e risorse.

La formulazione del problema di partizionare un insieme di entità in gruppi *omogenei*, basandosi sulle *differenze* fra le entità associate ad ogni sottoinsieme è stato a lungo studiato; le sue applicazioni coinvolgono, oltre che ai problemi di network design descritti, situazioni eterogenee come la caratterizzazione di aree geografiche e pattern classification.
Il primo modello per problemi di localizzazione è stato proposto da Alfred Weber nel 1909; a lui si deve soprattutto l'introduzione del paradigma di localizzazione basata sulla minimizzazione dei costi di trasporto.
Nei problemi multirisorse, tuttavia, problemi di localizzazione e clustering interagiscono tra loro e devono essere, pertanto, affrontati *contemporaneamente*.

Tra i problemi di localizzazione multirisorse, il problema delle *p*-mediane, oltre ad essere interessante dal punto di vista teorico, trova applicazioni negli scenari di network design appena descritti. Metodi basati su programmazione mista intera hanno dimostrato di essere efficaci in tale ambito; il recente sviluppo tecnologico

dei MIP solvers ha favorito, inoltre, l'applicazione di tecniche di programmazione matematica basate sui rilassamenti lineari, come la *column generation*.

Capacitated Facility Location Problem

Nella famiglia dei problemi di localizzazione multirisorsa hanno grande importanza i problemi di *capacitated facility location* (CFLP), che modellano, ad esempio problemi di distribuzione di servizi di emergenza sul territorio.

Su un grafo avente *N* nodi vengono individuati *M* potenziali siti candidati all'apertura di facilities (i punti di servizio). Ogni nodo del grafo ha una domanda che deve essere soddisfatta ed ogni facility ha una capacità; l'obiettivo è soddisfare la domanda di ogni nodo, aprendo delle facilities in alcuni degli *M* siti candidati ed assegnando i nodi alle varie facilities, senza eccederne la capacità. La domanda di ogni nodo può essere soddisfatta utilizzando più facilities. Si vuole, inoltre, minimizzare il costo dell'operazione, dato dai costi di trasporto (proporzionali alla distanza del nodo dalle facilities a cui è assegnato) e dai costi fissi per l'apertura di ogni facility.

La letteratura inerente i problemi di CFLP è molto ricca; per la soluzione esatta sono stati proposti diversi algoritmi in cui il bound duale è valutato tramite rilassamento lineare [Erle82] e rilassamento Lagrangeano [VRoy86] [Chri83], o anche tramite scomposizione di Benders [Geof74] [VRoy86]. Possono essere risolti all'ottimo, in tal modo, problemi che coinvolgono fino a 50 utenti e 50 siti candidati ad ospitare facilities. Per la risoluzione approssimata si dimostrano molto efficaci algoritmi euristici *greedy* combinati con *scambi miglioranti* [Kueh63] [Lehr66], oppure algoritmi basati su euristiche Lagrangeane [Corn91], che sono in grado di risolvere problemi su grafi di centinaia di nodi.

Per un'analisi completa dei metodi utilizzati per la risoluzione euristica o esatta del problema si rimanda all'articolo di Sridharan [Srid95] ed al testo [FL].

Single Source Capacitated Facility Location Problem

Per l'applicazione a problemi di network design è diffuso l'utilizzo di una variante di CFLP: il problema di *capacitated facility location* con *single source constraints*

(CFLP-SS), ovvero un problema di CFLP in cui ogni utente deve essere servito da una sola facility.

Neebe e Rao [Neeb83] studiano il problema di CFLP-SS. Viene presentato un algoritmo esatto che ricorre ad una riformulazione di CFLP-SS come problema di *set partitioning* con *side constraints* per sfruttare tecniche di column generation, risolvendo problemi su grafi di 35 nodi aventi fino a 25 possibili facilities, in alcuni secondi. Klincewicz e Luss [Klin86] realizzano un algoritmo esatto con rilassamento Lagrangeano dei vincoli di capacità per lo stesso problema, risolvendo in pochi minuti istanze su grafi di 50 nodi.

Pirkul [Pirk87] propone un algoritmo basato su un rilassamento Lagrangeano per la soluzione esatta del problema, risolvendo all'ottimo problemi su grafi di 100 nodi in pochi minuti. Pirkul analizza anche la possibilità di ottenere soluzioni approssimate con un algoritmo di branch & bound troncato.

Rönnqvist e altri [Ronn99] realizzano un'euristica basata sulla risoluzione di una serie di problemi di matching. L'euristica è stata testata su istanze di varie dimensioni (fino a un massimo di 200 nodi) ed è giunta all'ottimo nel 62% dei casi. Cortinhal e Captivo [Cort03] combinano euristiche Lagrangeane con ricerca tabù ottenendo un gap medio del 0.15%. Holmberg e altri [Holm99] presentano un algoritmo esatto basato su euristica Lagrangeana in grado di risolvere all'ottimo problemi su grafi di 200 nodi in pochi minuti.

Capacitated Concentrator Location Problem

Il problema in cui ciascun centro di servizio deve appartenere al cluster che descrive è detto *capacitated concentrator location problem.* L'aggiunta di questo vincolo rende il CCLP più vicino alle situazioni reali rispetto al SS-CFLP.

Aggiungendo un vincolo di cardinalità, ovvero specificando il massimo numero di clusters in cui partizionare l'insieme dei nodi del grafo, si ottiene il *capacitated p-concentrator location problem*.

Bramel e Simchi-Levi [Bram95] presentano un'euristica per problemi di routing ottenuta risolvendo la formulazione del *capacitated vehicle routing problem* (CVRP) come CCLP. Ambrosino e Sciomachen [Ambr03] si ispirano a questa

strategia risolutiva per ottenere un algoritmo di clustering per problemi di progettazione per reti di distribuzione. Labbè e Yaman [Labb03] studiano le proprietà del politopo del CCLP per sviluppare un algoritmo branch & cut per il *Quadratic Capacitated Concentrator Location Problem* (QCL).

P-Median Problem

Il problema delle p-mediane (PMP) consiste nel partizionare un insieme di nodi di un grafo in p clusters disgiunti, minimizzando le *differenze* all'interno di ciascun cluster. Il partizionamento è equivalente alla scelta di p nodi (le *mediane*) da un insieme di candidati, che identifichino i clusters. Le differenze tra i nodi all'interno dei clusters sono calcolate come la somma pesata delle distanze (intese come cammino minimo) tra ogni nodo e la mediana del cluster in cui è inserito.
Questo problema, definito per la prima volta da Hakimi (1963), è stato ampiamente studiato all'interno della famiglia dei problemi localizzazione multirisorsa; l'articolo [Hans97] offre una panoramica sui problemi di clustering, nel volume [DLT] (capitolo 2) vengono analizzati nel dettaglio il problema delle p-mediane classico e le sue generalizzazioni. Algoritmi basati su rilassamento Lagrangeano sono stati proposti in [Naru77], [Corn77], [Chri81], [Beas85]. Sono presentati risultati per la soluzione di problemi aventi grafi fino a 200 nodi, con 5 mediane e 150 nodi ed un numero arbitrario di mediane, in alcuni minuti. Utilizzando macchine per il calcolo vettoriale sono stati risolti problemi aventi fino a 900 nodi e 90 mediane o 600 nodi e 120 mediane in alcune decine di minuti. In [Galv79] e [Hanj85] sono presentati approcci basati sulla formulazione duale del problema che risolvono problemi su grafi di 75 nodi ed un numero arbitrario di mediane, in poche decine di secondi.
Avella, Sassano e Vasilév [Avel03] presentano un algoritmo branch & price & cut per risolvere istanze di p-median molto grandi. La strategia risolutiva presentata prevede *column-and-row generation* per risolvere il rilassamento lineare, e piani di taglio per rafforzare la formulazione. Gli esperimenti sono stati condotti su istanze il cui numero di nodi varia da 600 a 3795: i problemi più piccoli vengono

risolti all'ottimo in pochi minuti, mentre per le istanze da 3795 nodi i tempi sono molto variabili e l'ottimo viene raggiunto mediamente in 27 ore.

I problemi PMP in cui il grafo è un albero possono essere risolti in tempo polinomiale, come proposto da Goldman [Gold71] (per p = 1) o tramite l'algoritmo di Kariv e Hakimi [Kari79].

Capacitated P-Median Problem

PMP può essere generalizzato associando ad ogni mediana una capacità e ad ogni nodo un peso ed imponendo che la somma dei pesi dei nodi in ogni cluster non possa eccedere la capacità della mediana relativa. Questa generalizzazione è conosciuta come problema delle p-mediane con capacità (CPMP).
Il problema delle p-mediane con capacità è conosciuto anche come *sum of stars problem* [Hans97]. Kariv e Hakimi [Kari79] ne hanno dimostrato l'appartenenza alla classe di problemi *NP*-hard. CPMP si differenzia da CFLP per due aspetti: in CFLP può essere attivato un numero qualsiasi di facilities, l'apertura di una nuova facility comporta, come descritto, dei costi fissi; in CPMP il numero di facilities è assegnato (*p*) e l'apertura di una facility in un nodo non comporta penalizzazione con costi fissi. CFLP prevede, inoltre, che un utente possa essere servito da più facilities, CPMP impone che ogni utente si riferisca ad una sola facility (come nel caso di CCLP).
Per CPMP non sono stati presentati molti metodi: algoritmi euristici in
grado di raggiungere soluzioni per problemi con 100 nodi e 20 mediane in poche decine di secondi sono stati presentati da Mulvey e Beck [Mulv84].
Maniezzo, Mingozzi e Baldacci [Mani98] propongono l'utilizzo di tecniche di programmazione evolutiva, Golden e Skiscim [Gold86] ricorrono a Simulated Annealing, Osman e Christofides [Osma94] sviluppano un algoritmo ibrido tra Simulated Annealing e Tabu Search. Questi algoritmi forniscono soluzioni sperimentalmente "vicine" all'ottimo su problemi aventi N = 100 p = 20 in meno di un'ora.

Baldacci, Hadjiconstantinou, Maniezzo e Mingozzi [Bald02] descrivono un algoritmo per la soluzione euristica di CPMP che permette la valutazione della massima distanza dall'ottimalità della soluzione trovata, presentando risultati in cui problemi su grafi aventi 200 nodi con 20 mediane vengono risolti in meno di un'ora. Analizzano anche il caso di problemi con vincoli aggiuntivi di *incompatibilità* tra nodi su problemi dalle dimensioni analoghe, la cui soluzione è raggiunta entro 2 ore.

Lorena e Senne [Lore04] hanno recentemente proposto un algoritmo che combina column generation e rilassamento Lagrangeano surrogato, ottenendo su istanze da 400 nodi un gap medio inferiore a 1% in qualche decina di minuti. Ceselli e Righini [Cese05] presentano un algoritmo Branch & Price per la soluzione esatta di CPMP in grado di risolvere istanze fino a 200 nodi entro 2 ore.

Struttura dell'esposizione

Nei capitoli successivi viene presentato un algoritmo esatto di tipo *Branch & Price* in grado di trattare in maniera uniforme l'intera classe di problemi illustrata.

Nel capitolo 2 vengono mostrate le formulazioni dei problemi affrontati e le riformulazioni disaggregate con Dantzig - Wolfe. Il capitolo 3 offre una panoramica della teoria che sta alla base degli algoritmi di column generation. Nel capitolo 4 viene descritto l'algoritmo implementato e le diverse componenti che lo costituiscono e nel capitolo 5 sono raccolti i risultati degli esperimenti effettuati e delle brevi conclusioni.

2. Modelli

2.1 Single Source Capacitated Facility Location Problem

Formulazione

Dati:

- Sia $N = \{1...N\}$ l'insieme dei nodi di un grafo $G = (N, E)$ e $M = \{1...M\}$ l'insieme dei nodi candidati ad essere scelti come *facilities*.
- Ad ogni nodo $i \in N$ è associato un coefficiente $w_i \in R_+$ che rappresenta la sua domanda.
- Ogni coefficiente $Q_j \in R_+$ rappresenta la capacità del nodo $j \in M$ candidato *facility*.
- Per i nodi candidati ad essere *facility* è definito anche il coefficiente $f_j \in R_+$ che rappresenta il costo di apertura della *facility*.
- Ogni coefficiente $d_{ij} \in R_+ \quad i, j \in N$ esprime la *distanza* tra due nodi i e j del grafo ed è proporzionale alla lunghezza del cammino minimo tra di essi (t_{ij}) ed alla domanda del nodo i: $d_{ij} = t_{ij} w_i$.
- Si assume che il grafo non contenga coefficienti di distanza negativi $d_{ij} \geq 0 \quad \forall i, j \in N$.

Variabili:

- I clusters sono insiemi $S_j \subseteq N$ con il vincolo che $\bigcup_{j \in M} S_j = N, \quad S_{j'} \cap S_{j''} = \varnothing \quad \forall j', j'' \in M$.
- Ogni cluster è individuato tramite il suo nodo *facility* $m_j, \quad j \in M$.
- La formulazione classica del problema contempla l'uso di $N * M$ variabili binarie x_{ij} .
- In particolare $x_{ij} = 1$ se il nodo i viene assegnato alla *facility* j e $x_{ij} = 0$ altrimenti ($x_{ij} = 1$ se e solo se $i \in S_j$).
- M variabili binarie $y_j, \quad j = 1...M$ indicano, per ogni nodo j, se j è selezionato come *facility*.

Vincoli:

- La cardinalità di ciascun cluster non è vincolata, ma la somma delle domande dei suoi nodi non può superare la capacità della *facility*.
- Ogni nodo deve essere servito da una sola *facility*.

Funzione obiettivo:

- L'obiettivo è minimizzare i costi complessivi di apertura delle *facilities* e la somma delle distanze tra ogni nodo e la *facility* del cluster a cui è assegnato, nel rispetto dei vincoli descritti.

Modello:

Ecco la formulazione completa del problema.

SS-CFLP)

$$\min v = \sum_{i \in N} \sum_{j \in M} d_{ij} x_{ij} + \sum_{j \in M} f_j y_j \qquad (2.1)$$

s.t.

$$\sum_{j \in M} x_{ij} = 1 \quad \forall i \in N \tag{2.2}$$

$$\sum_{i \in N} w_i x_{ij} \leq Q_j y_j \quad \forall j \in M \tag{2.3}$$

$$x_{ij} \in \{0,1\} \quad \forall i \in N, \forall j \in M \tag{2.4}$$

$$y_j \in \{0,1\} \quad \forall j \in M \tag{2.5}$$

L'obiettivo (2.1) è minimizzare i costi di apertura delle *facilities* e la somma delle distanze tra ciascun nodo e la *facility* associata.

I vincoli di set partitioning (2.2) impongono che ogni nodo sia assegnato ad una *facility*. I vincoli (2.3) hanno un doppio effetto: imporre che la capacità di ciascuna facility non possa essere inferiore alla somma dei pesi dei nodi ad essa assegnati ed impedire che dei nodi vengano associati a *facilities* non attive (per le quali $y_j = 0$). I vincoli (2.4 e 2.5) impongono l'integralità della soluzione.

Riformulazione secondo Dantzig – Wolfe

La regione di ammissibilità di SS-CFLP è l'intersezione di M insiemi indipendenti Ω^j che soddisfano vincoli di capacità (2.3). Solo i vincoli (2.2) pongono in relazione i diversi insiemi di variabili; i vincoli di questa famiglia vengono detti *di unione*, per la loro proprietà di determinare l'aggregazione degli M sottoproblemi descritti.

Dantzig e Wolfe [Dant60] propongono un metodo generale per riformulare problemi di programmazione (mista) intera in modo da ottenerne la disaggregazione in sottoproblemi distinti. Nel capitolo 3 viene spiegato come viene utilizzata questa riformulazione all'interno di algoritmi branch & price.

La riformulazione di Dantzig-Wolfe per il Single Source-Capacitated Facility Location Problem è la seguente:

IPM SS-CFLP)

$$\min v = \sum_{k=1}^{K} c^k \lambda_k \tag{2.6}$$

s.t.

$$\sum_{k \in Z} x_i^k \lambda_k = 1 \quad \forall i \in N \tag{2.7}$$

$$\sum_{k \in Z} \gamma_j^k \lambda_k \leq 1 \quad \forall j \in M \tag{2.8}$$

$$\lambda_k \in \{0,1\} \quad \forall k \in Z \tag{2.9}$$

Ogni colonna rappresenta un *cluster* ammissibile. Pertanto SS-CFLP (e gli altri problemi di localizzazione multirisorse) viene riformulato come problema di *set partitioning* con dei *side constraints*. Ciascuna colonna viene costruita rispettando i vincoli di capacità (2.3).

Le variabili λ_k sono associate alle colonne mentre i coefficienti γ_j^k indicano se il cluster k è riferito alla facility j. Infine i coefficienti x_i^k esprimono l'appartenenza del nodo i al cluster k.

L'obiettivo (2.6) è minimizzare la somma dei costi delle colonne scelte. Poiché ogni colonna identifica un cluster, il costo c^k è dato dal costo di apertura f_j della *facility* e dalla somma delle distanze della facility dai nodi contenuti nel cluster. I vincoli (2.7) assicurano che ogni nodo sia assegnato ad una facility e i vincoli (2.8) che ciascun cluster contenga un solo distributore di servizio.

2.2 Capacitated Concentrator Location Problem

Come descritto nell'introduzione, indichiamo con CCLP la tipologia di problemi di localizzazione multirisorse in cui i centri di servizio devono appartenere al cluster che descrivono. Inoltre gli insiemi M e N, che contengono rispettivamente i nodi candidati e i nodi del grafo, coincidono. Di seguito vengono esaminate nel dettaglio le differenze tra questi due problemi.

Formulazione

CCLP)

$$\min v = \sum_{i \in N} \sum_{j \in M} d_{ij} x_{ij} + \sum_{j \in M} f_j x_{jj} \qquad (2.10)$$

s.t.

(2.2) , (2.4)

$$\sum_{i \in N} w_i x_{ij} \leq Q_j x_{jj} \quad \forall j \in M \qquad (2.11)$$

Come si può vedere dal modello, nella funzione obiettivo (2.10) e nel vincolo di capacità (2.11) in questo problema le M variabili binarie y_j, $j = 1...M$ sono sostituite dalle variabili x_{jj}: dato che ciascun concentratore viene assegnato a se stesso, $x_{jj} = 1$ se e solo se j è concentratore.

Riformulazione secondo Dantzig – Wolfe

IPM CCLP)

$$\min v = \sum_{k=1}^{K} c^k \lambda_k$$

s.t.

(2.7), (2.9)

I vincoli di *set partitioning* (2.7) impongono che ciascun nodo compaia all'interno di un solo cluster scelto. In questo caso, i vincoli (2.8) sono ridondanti, e possono essere rimossi dal problema.

Come per il SS-CFLP, i costi c^k delle colonne tengono conto dei costi di apertura del concentratore associato e della distanza di quest'ultimo dai nodi.

2.3 Capacitated P-Median Problem

Come visto nell'introduzione, il PMP consiste nel partizionare un grafo in un numero definito di cluster disgiunti in maniera tale da minimizzare le differenze all'interno dei cluster. Il CPMP nasce come generalizzazione del PMP: a ogni nodo è assegnato un peso e alle mediane è associata una capacità che non può essere ecceduta. Le mediane si distinguono dalle *facilities* in quanto non vi sono costi di apertura.

Formulazione

CPMP)

$$\min v = \sum_{i \in N} \sum_{j \in M} d_{ij} x_{ij} \qquad (2.12)$$

s.t.

(2.2), (2.3), (2.4), (2.5)

$$\sum_{j \in M} y_j \leq p \qquad (2.13)$$

L'obiettivo (2.12) è minimizzare la somma delle distanze tra ciascun nodo e la mediana associata. Il vincolo (2.13) impone che le mediane attive siano al più p; il vincolo (2.13) può essere posto in forma di $\leq$ in quanto in caso di disuguaglianza stretta, è possibile completare la soluzione con dei clusters di costo nullo.

Riformulazione secondo Dantzig – Wolfe

IPM CPMP)

$$\min v = \sum_{k=1}^{K} c^k \lambda_k$$

s.t.

(2.7), (2.8), (2.9)

$$\sum_{k \in Z} \lambda_k \leq p \qquad (2.14)$$

In questa riformulazione appare il vincolo di cardinalità (2.14) che specifica il numero di mediane attive. Poiché le mediane non prevedono costi di apertura, il costo di ciascuna colonna è dato esclusivamente dalla somma delle distanze della mediana dai nodi contenuti nel cluster.

2.4 Capacitated P-Concentrator Location Problem senza costi di apertura

Questo problema è assimilabile a un caso particolare di CPMP in cui ogni nodo mediana deve essere inserito nel cluster da esso descritto (come per il CCLP).

Formulazione

CPCLP0)

$$\min v = \sum_{i \in N} \sum_{j \in M} d_{ij} x_{ij}$$

s.t.

(2.2), (2.3), (2.4)

$$\sum_{j \in M} x_{jj} \leq p \qquad (2.15)$$

Esattamente come per il CCLP, in questa formulazione sono state sostituite le variabili $y_j \quad j = 1...M$ con le variabili x_{jj}.

Riformulazione secondo Dantzig – Wolfe

IPM CPCLP0)

$$\min v = \sum_{k=1}^{K} c^k \lambda_k$$

s.t.

(2.7), (2.14), (2.9)

2.5 Single Source Capacitated P-Facility Location Problem

Questo problema si distingue dal SS-CFLP in quanto è definito il massimo numero di *facility* da aprire (p).

Formulazione

La formulazione del problema è la seguente.

SS-CPFLP)

$$\min v = \sum_{i \in N} \sum_{j \in M} d_{ij} x_{ij} + \sum_{j \in M} f_j y_j$$

s.t.

(2.2), (2.3), (2.13), (2.4), (2.5)

Riformulazione secondo Dantzig – Wolfe

IPM SS-CPFLP)

(2.6)

s.t.

(2.7), (2.14), (2.8),(2.9)

2.6 Capacitated P-Concentrator Location Problem

La formulazione di questo problema è identica a quella del CCLP, con l'aggiunta del vincolo di cardinalità che specifica il massimo numero di *concentratori* utilizzabili (p).

Formulazione

CPCLP)

$$\min v = \sum_{i \in N} \sum_{j \in M} d_{ij} x_{ij} + \sum_{j \in M} f_j x_{jj}$$

s.t.

(2.2) , (2.11), (2.15), (2.4)

Il vincolo di cardinalità (2.15) è posto in forma di $\leq$ in quanto rappresenta un limite al massimo numero di concentratori apribili: infatti in presenza di costi fissi potrebbe essere vantaggioso utilizzarne un numero inferiore.

Riformulazione secondo Dantzig – Wolfe

IPM CPCLP)

$$\min v = \sum_{k=1}^{K} c^k \lambda_k$$

s.t.

(2.7), (2.14), (2.9)

3. Richiami teorici sul Column Generation

3.1 Introduzione

La decomposizione è una delle tecniche più sfruttate nell'ottimizzazione. Consideriamo ora il problema di programmazione intera (IP) $\max\{cx : x \in X\}$; tale problema ha una regione di ammissibilità X che può essere scritta come l'intersezione di due o più insiemi avente la struttura $X = \bigcap_{k=0}^{K} X^k, K \geq 1$. Consideriamo il caso particolare in cui i vincoli assumono la forma:

$$\begin{array}{llllll} A^1x^1 & + A^2x^2 & + \ldots & + A^Kx^K & = b \\ D^1x^1 & & & & \leq d_1 \\ & \ldots & & & \leq . \\ & & \ldots & & \leq . \\ & & & D^Kx^K & \leq d_K \\ x^1 \in Z_+^{n_1}, & \ldots & \ldots & , x^K \in Z_+^{n_K} & \end{array}$$

tale che gli insiemi $X^k = \left\{x^k \in Z_+^{n_k} : D^k x^k \leq d_k\right\}$ sono indipendenti per $k = 1,\ldots,K$, e solo i vincoli di *unione* $\sum_{k=1}^{K} A^k x^k = b$ raggruppano i diversi insiemi di variabili.

Data la funzione obiettivo $\max \sum_{k=1}^{K} c^k x^k$, due approcci che consentono di beneficiare di questa struttura sono il *cut generation,* in cui si cerca di generare disuguaglianze valide per ogni sottoinsieme $X^k, k = 1,\ldots,K$, e il rilassamento Lagrangeano, in cui si rendono duali i vincoli di unione per ottenere un problema duale:

$$\min_u \quad L(u),$$

dove

$$L(u) = \max\left\{\sum_{k=1}^{K}(c^k - uA^k)x^k + ub : x^k \in X^k \quad k = 1,...,K\right\},$$

e il calcolo di $L(u)$ porta ad avere K sottoproblemi distinti:

$$L(u) = \sum_{k=1}^{K}\max\left\{(c^k - uA^K)x^k : x^k \in X^k\right\} + ub.$$

Esiste, tuttavia, un terzo modo per disaggregare la struttura di problemi interi della forma descritta. Si assume che ogni insieme X^k è limitato per $k = 1,...,K$. L'approccio essenzialmente implica la risoluzione di un problema equivalente nella forma:

$$\max\left\{\sum_{k=1}^{K}\gamma^k\lambda^k : \sum_{k=1}^{K}B^k\lambda^k = \beta, \lambda^k \geq 0, \lambda^k \in \mathrm{Z}, \forall k = 1,...,K\right\}$$

dove ogni matrice B^k ha un numero alto di colonne, una per ognuno dei punti ammissibili in X^k, e ogni vettore λ^k contiene le variabili corrispondenti.

3.2 Riformulazione di Dantzig – Wolfe di un IP

Consideriamo ora il problema nella forma:

IP)

$$z = \max\left\{\sum_{k=1}^{K}c^k x^k : \sum_{k=1}^{K}A^k x^k = b, x^k \in X^k \quad \forall k = 1,...,K\right\}$$

dove $X^k = \left\{x^k \in Z_+^{n_k} : D^k x^k \leq d_k, \forall k = 1,...,K\right\}$. Assumendo che ogni insieme X^k contenga un insieme di punti grande ma finito $\left\{x^{k,t}\right\}_{t=1}^{T_k}$, abbiamo che

$$X^k = \left\{x^k \in R^{n_k} : x^k = \sum_{t=1}^{T_k}\lambda_{k,t}x^{k,t}, \sum_{t=1}^{T_k}\lambda_{k,t} = 1, \lambda_{k,t} \in \{0,1\}, \forall t = 1,...,T_k\right\}.$$

Sostituendo x^k si ottiene il *Master Problem* di un problema intero equivalente:

IPM)

$$\begin{aligned} z &= \max \textstyle\sum_{k=1}^{K}\sum_{t=1}^{T_k}\left(c^k x^{k,y}\right)\lambda_{k,t} \\ &\textstyle\sum_{k=1}^{K}\sum_{t=1}^{T_k}\left(A^k x^{k,t}\right)\lambda_{k,t} = b \\ &\textstyle\sum_{t=1}^{T_k}\lambda_{k,t} = 1, \forall k = 1,...,K \\ \lambda_{k,t} &\in \{0,1\}, \forall t = 1,...,T_k, \forall k = 1,...,K. \end{aligned}$$

3.3 Risolvere il Master Linear Program

Attraverso l'uso di tecniche di column generation è possibile risolvere il rilassamento lineare dell'IPM, chiamato *Linear Master Problem*:

LPM)

$$
\begin{aligned}
z^{LPM} = \max & \sum_{k=1}^{K} \sum_{t=1}^{T_k} \left(c^k x^{k,t}\right) \lambda_{k,t} \\
& \sum_{k=1}^{K} \sum_{t=1}^{T_k} \left(A^k x^{k,t}\right) \lambda_{k,t} = b \\
& \sum_{t=1}^{T_k} \lambda_{k,t}, \forall k = 1,...,K \\
& \lambda_{k,t} \geq 0, \forall t = 1,...,T^k, \forall k = 1,...,K
\end{aligned}
$$

dove si ha una colonna $\begin{pmatrix} c^k x \\ A^k x \\ e_k \end{pmatrix}$ per ogni $x \in X^k$. Si usa $\{\pi_i\}_{i=1}^{m}$ per rappresentare le variabili duali associate ai vincoli di unione, e $\{\mu_k\}_{k=1}^{K}$ per le variabili duali del secondo insieme di vincoli, i vincoli di *convessità*.

Il problema lineare viene risolto attraverso l'algoritmo del simplesso primale. Comunque, il passo di pricing per la scelta della colonna da far entrare in base deve essere modificato a causa dell'enorme numero di colonne. Non si effettua il price di ogni colonna, ma si risolvono K problemi di ottimizzazione per trovare le colonne con il più alto costo ridotto.

Inizializzazione

Si suppone che un sottoinsieme di colonne (almeno una per ogni k) sia disponibile, in maniera tale da fornire un *Restricted Linear Master Problem* ammissibile

RLPM)

$$
\begin{aligned}
\tilde{z}^{LPM} = \max & \tilde{c}\tilde{\lambda} \\
& \tilde{A}\tilde{\lambda} = \tilde{b} \\
& \tilde{\lambda} \geq 0
\end{aligned}
$$

dove $\widetilde{b} = \begin{pmatrix} b \\ 1 \end{pmatrix}$, $\widetilde{A}$ è generato dall'insieme di colonne disponibili ed è una sottomatrice di

$$\begin{pmatrix} A^1x^{1,1} & \dots & A^1x^{1,T_1} & A^2x^{2,1} & \dots & A^2x^{2,T_2} & \dots & A^Kx^{K,1} & \dots & A^Kx^{K,T_K} \\ 1 & \dots & 1 & & & & & & & \\ & & & 1 & \dots & 1 & & & & \\ & & & & & & \dots & & & \\ & & & & & & & 1 & \dots & 1 \end{pmatrix},$$

e $\widetilde{c}, \widetilde{\lambda}$ sono i costi e le variabili corrispondenti. Risolvendo RLPM si ottiene una soluzione ottima primale $\widetilde{\lambda}^*$ e una soluzione ottima duale $(\pi, \mu) \in R^m \times R^K$.

Ammissibilità primale

Ogni soluzione ammissibile per RLPM è ammissibile per LPM. In particolare, $\widetilde{\lambda}^*$ è una soluzione ammissibile per LPM, e così $\widetilde{z}^{LPM} = \widetilde{c}\widetilde{\lambda}^* = \sum_{i=1}^{m} \pi_i b_i + \sum_{k=1}^{K} \mu_k \le z^{LPM}$.

Verifica del raggiungimento dell'ottimo per LPM

Bisogna verificare che (π, μ) sia un duale ammissibile per LPM. Ciò implicherebbe verificare per ogni colonna, ovvero per ogni k e ogni $x \in X^k$, che sia $c^k x - \pi A^k x - \mu_k \le 0$. Invece di esaminare ogni punto separatamente, è possibile considerare tutti i punti in X^k implicitamente risolvendo un sottoproblema di ottimizzazione:

$$\zeta_k = \max\left\{\left(c^k - \pi A^k\right)x - \mu_k : x \in X^k\right\}.$$

Criterio di termine

Se $\zeta_k = 0, \forall k = 1,...,K$, la soluzione (π, λ) è un duale ammissibile per LPM, e quindi $z^{LPM} \leq \sum_{i=1}^{m} \pi_i b_i + \sum_{k=1}^{K} \mu_k$. Quando il valore della soluzione primale ammissibile $\tilde{\lambda}$ eguaglia quello di questo upper bound, $\tilde{\lambda}$ è ottimo per LPM.

Generare nuove colonne

Se $\zeta_k > 0$ per qualche k, la colonna che corrisponde alla soluzione ottima ($\tilde{x}^k$) del sottoproblema ha costo ridotto positivo. Introducendo la colonna $\begin{pmatrix} c^k \tilde{x}^k \\ A^k \tilde{x}^k \\ e_k \end{pmatrix}$ si ottiene un nuovo RLPM che può essere riottimizzato facilmente (attraverso l'algoritmo del simplesso primale).

Bound Duale

Dal sottoproblema, si ha che $\zeta_k \geq (c^k - \pi A^k)x - \mu_k, \forall x \in X^k$. Ne consegue che $(c^k - \pi A^k)x - \mu_k - \zeta_k \leq 0, \forall x \in X^k$. Di conseguenza ponendo $\zeta = (\zeta_1,...,\zeta_K)$, abbiamo che $(\pi, \mu + \zeta)$ è un duale ammissibile per LPM. Quindi si ha che

$$z^{LPM} \leq \pi b + \sum_{k=1}^{K} \mu_k + \sum_{k=1}^{K} \zeta_k .$$

Questa osservazione conduce a un algoritmo per LPM che termina quando $\zeta_k = 0, \forall k = 1,...,K$. Comunque, come per il rilassamento Lagrangeano, è possibile terminare prima del verificarsi di questa condizione.

Criterio di termine alternativo

Se la soluzione $(\tilde{x}^1....,\tilde{x}^K)$ del sottoproblema soddisfa i vincoli di unione $\sum_{k=1}^{K} A^k x^k = b$, allora la soluzione trovata è ottima.

Questo avviene in quanto $\zeta_k = \left(c^k - \pi A^k\right)\tilde{x}^k - \mu_k$ implica che $\sum_k c^k \tilde{x}^k = \sum_k \pi A^k \tilde{x}^k + \sum_k \mu_k + \sum_k \zeta_k = \pi b + \sum_k \mu_k + \sum_k \zeta_k$. Di conseguenza la soluzione ammissibile primale ha lo stesso valore dell'upper bound su z^{LPM}.

4. Algoritmo Branch & Price

In questa tesi e, in particolare, in questo capitolo viene presentato un algoritmo esatto di tipo *Branch & Price* in grado di trattare in maniera uniforme l'intera classe di problemi illustrata. Il progetto è stato sviluppato a partire da un algoritmo esistente per CPMP. Per trattare tutte le classi di problema descritte, è stato necessario generalizzare diverse componenti. Per i dettagli implementativi dell'algoritmo originario si rimanda a [Cese05].

Branch & Bound

I problemi di localizzazione multirisorse studiati in questo lavoro sono *NP*-Hard per cui, a meno che $NP = P$, per ottenere la soluzione ottima è necessario ricorrere all'enumerazione di tutte le possibili soluzioni. Questa enumerazione viene effettuata in maniera implicita ricorrendo ad algoritmi di *branching*, ovvero esplorando un albero in cui ogni nodo corrisponde ad un sottoproblema. La radice è il problema iniziale, in cui tutte le variabili sono libere. L'insieme di soluzioni corrispondenti (S_0) può essere partizionato in F sottoinsiemi S_k, che rappresentano le soluzioni di altrettanti *sottoproblemi* ($S_1 ... S_F$, $\bigcup_{k=1...F} S_k = S_0$), fissando opportunamente il valore di una o più variabili libere. Ogni sottoproblema è radice di un nuovo albero; ai suoi figli corrispondono F sottoinsiemi dell'insieme di soluzioni S_k. Le foglie dell'albero di branching rappresentano tutte le possibili soluzioni del problema.

Ricorrendo ad algoritmi euristici è possibile sperare di ottenere soluzioni ammissibili partendo dalle soluzioni parziali definite in ciascun sottoproblema. Il valore delle soluzioni ammissibili eventualmente trovate è un *bound primale*.

Per ogni sottoproblema viene valutato il valore di una soluzione ammissibile duale, sicuramente non peggiore di qualsiasi sua soluzione ammissibile. Questo valore viene detto *bound duale*.
Se in un sottoproblema il bound duale non risulta migliore del bound primale, la valutazione del nodo dell'albero di branching corrispondente e di tutti i suoi nodi-figlio può essere tralasciata.

Branch & Price

Nelle sezioni seguenti viene proposto un algoritmo per la soluzione esatta di problemi multirisorse: il bound duale, per ogni sottoproblema, viene valutato risolvendo il rilassamento lineare della riformulazione come *set partitioning problem* con *side constraints*. Questa nuova formulazione contempla un numero di variabili esponenziale nella dimensione del problema affrontato. E' possibile considerare solo un sottoinsieme delle variabili del problema riformulato, e le relative colonne nella matrice dei vincoli, ottenendo un problema *ridotto*: partendo da una soluzione di base ammissibile si possono inserire nel problema ridotto solo le colonne corrispondenti a variabili candidate ad entrare in base. Il problema di determinare quali siano le colonne aventi costo ridotto – *price* – opportuno (negativo, per problemi di minimizzazione) è detto *pricing problem*. L'individuazione ed inserimento iterativo di opportune colonne nel problema ridotto è chiamato *column generation*. L'applicazione di column generation a problemi di ottimizzazione (mista) intera, per i quali è richiesta l'enumerazione implicita delle soluzioni, conduce ad algoritmi di *branch & price*.

4.1 Bound duale

In questo capitolo è descritto il metodo utilizzato per risolvere il LPM. Per rendere le formulazioni modulari rispetto al problema, i termini opzionali sono stati posti tra parentesi quadre.

Rilassamento di IPM

I vincoli di *set partitioning* (2.7) possono essere posti in forma di $\geq$, poiché nella soluzione ottima nessun nodo è assegnato a due clusters: esisterebbe in tal caso una soluzione ammissibile migliore, ottenibile rimuovendo il nodo dal cluster più svantaggioso. Per mantenere la notazione uniforme, tutte le disuguaglianze in forma di $\leq$ sono portate in forma di $\geq$.

LPM)

$$\min \omega_{CG} = \sum_{k=1}^{K} c_k \lambda_k \tag{4.1}$$

s.t.

$$\sum_{k \in Z} x_i^k \lambda_k \geq 1 \quad \forall i \in N \tag{4.2}$$

$$\left[-\sum_{k \in Z} \lambda_k \geq -p \right] \tag{4.3}$$

$$\left[-\sum_{k \in Z} \gamma_j^k \lambda_k \geq -1 \quad \forall j \in M \right] \tag{4.4}$$

$$0 \leq \lambda_k \left[\leq 1 \right] \quad \forall k \in Z \tag{4.5}$$

I vincoli (4.4), quando presenti, implicano i vincoli $\lambda_k \leq 1$, che pertanto possono essere rimossi.

E' possibile dimostrare [Wols98] che il rilassamento continuo del IPM fornisce risultati equivalenti a quelli ottenibili risolvendo all'ottimo il *problema duale lagrangiano* relativo al rilassamento dei vincoli di *partitioning*.

Problema di Pricing

La cardinalità di Z (il numero di variabili in IPM) cresce esponenzialmente nel numero di nodi del grafo. All'aumentare delle dimensione dell'istanza, risolvere il rilassamento lineare di IPM (LPM) con metodi diretti diventa improponibile.

Infatti utilizzare il metodo del simplesso per risolvere LPM renderebbe necessario valutare il costo ridotto di $|Z|$ colonne ad ogni iterazione. E' possibile, tuttavia, risolvere il rilassamento lineare con le tecniche di *column generation* precedentemente descritte: rimuovendo dall'insieme Z un opportuno insieme di colonne si ottiene un *Master Problem Ristretto* (RLPM). RLPM deve contenere almeno una soluzione di base ammissibile. Risolto il rilassamento lineare di RLPM, l'informazione derivante dal valore delle variabili duali può essere sfruttata per determinare quali colonne di Z, non inserite in RLPM, avendo costo ridotto negativo sono candidate ad entrare in base.

Siano π, μ i vettori rispettivamente delle N variabili duali associate ai vincoli (4.2) e delle M variabili duali associate ai vincoli (4.4); sia υ la variabile duale associata al vincolo (4.3). Il costo ridotto di una colonna k che rappresenti un assegnamento per la facility j è

$$\zeta_k = \sum_{i \in N} (d_{i,j} - \pi_i) x_i^k \quad \left[+ \mu_j \gamma_j^k\right] \quad [+\upsilon] \quad \left[+ f_j \gamma_j^k\right]$$

dove $\pi_i \geq 0 \quad \forall_i \in N$, $\mu_j \geq 0 \quad \forall_j \in M$, $\upsilon \geq 0$, essendo associati a vincoli di $\geq$.

Invece che calcolare il costo ridotto di ogni colonna viene risolto il *pricing problem*, che consiste in M problemi di ottimizzazione volti ad individuare potenziali nuove colonne vantaggiose:

$(\forall j \in M)$

PP)

$$\min \zeta_j = \sum_{i \in N} (d_{i,j} - \pi_i) x_i^k \quad \left[+ \mu_j \gamma_j^k\right] \quad [+\upsilon] \quad \left[+ f_j \gamma_j^k\right]$$

s.t.

$$x^j \in \Omega^j$$

Se la facility non viene aperta, ovvero $\gamma_j^k = 0$ ($x_j^k = 0$ nel caso concentratori) e il cluster è descritto da un vettore $(0,...,0)^T$, si ha una colonna svantaggiosa che non viene inserita in RLPM. Quando $\lambda_j^k = 1$ viene calcolato il costo fisso e il *pricing problem* è un problema di *knapsack* $(\forall j \in M)$ [MT89].

KP^j)

$$\min \sum_{i \in N} (d_{i,j} - \pi_i) x_i^k$$

s.t.

$$\sum_{i \in N} w_i x_i^j \leq Q_j$$

$$x_i^j \in \{0,1\} \quad \forall i \in N$$

Risolto il pricing problem, può essere inserito in RLPM l'intero insieme delle colonne individuate aventi costo ridotto negativo, o parte di esso, per valutare nuovamente il valore ottimo del rilassamento lineare e ottenere un nuovo vettore di valori delle variabili duali.

Si procede iterativamente ricercando, come descritto, nuove colonne candidate ad entrare in base. E' importante notare come la successione dei valori del rilassamento lineare LPM ad ogni iterazione t di column generation ($\{\omega_{CG}\}_t$) sia monotona decrescente e converga al valore del rilassamento lineare di IPM. Se la soluzione esatta dei problemi di knapsack non permette di individuare nessuna nuova colonna potenzialmente vantaggiosa significa che la soluzione di base corrente è ottima. In tal caso il valore ottenuto è un bound duale valido per IPM.

Per la soluzione esatta dei problemi di knapsack si è scelto di utilizzare una versione adattata dell'algoritmo *MINKNAP* di Pisinger [Pisi95], che combina programmazione dinamica a tecniche di bounding e riduzione.

Come descritto, nel caso dei *concentratori* si ha che il distributore di servizio appartiene al cluster che descrive. L'algoritmo è stato quindi ulteriormente modificato per essere utilizzato col CCLP e le sue varianti studiate (CPCLP e CPCLP0), forzando l'inserimento del concentratore all'interno dell'insieme dei nodi serviti. Fissando $x_j^j = 1$ la funzione obiettivo viene incrementata di $(d_{j,j} - \pi_j)$ e la capacità del centro di servizio si riduce di w_j, ma la struttura del problema di pricing non cambia.

4.2 Column generation e rilassamento Lagrangeano

Come esposto nella sezione 4.1 e nel capitolo 3, è un bound duale valido solo il valore del rilassamento di RLPM al termine del processo di column generation (quando non è possibile individuare nessuna nuova colonna di costo ridotto negativo). Per alcuni problemi il numero di passi necessari potrebbe essere molto elevato; inoltre i valori di ω_{CG}^t decrescono rapidamente durante le prime iterazioni per poi variare più lentamente.

I valori delle variabili duali (π, μ, υ) all'ottimo corrispondono ai valori ottimi dei moltiplicatori del rilassamento Lagrangeano nella formulazione equivalente. E' possibile sfruttare la proprietà di equivalenza per i moltiplicatori per ottenere bounds duali validi durante il processo di column generation: ad ogni iterazione t è un bound duale valido

$$\omega_{LD}^t = \max\left\{\sum_{j\in M}\min\{\zeta_j,0\} + \sum_{i\in N}\pi_i \quad \left[-\sum_{j\in M}\mu_j\right] \quad [-p\upsilon] \quad \left[-\sum_{j\in M} f_j\right], \quad \omega_{LD}^{t-1}\right\}$$
$$= \max\{\omega_{LR}^1,\ldots,\omega_{LR}^t\}$$

dove $\zeta_k = \sum_{i\in N}(d_{i,j} - \pi_i)x_i^k \quad [+\mu_j] \quad [+\upsilon] \quad [+f_j]$ è il costo ridotto della colonna generata risolvendo all'ottimo KP^j. Infatti i valori di ζ_j si ottengono in modo del tutto analogo ai costi ridotti calcolati tramite rilassamento Lagrangeano.

Disponendo del valore Z^* di una soluzione ammissibile, il processo di column generation può essere terminato quando $Z^* \le \lceil \omega_{LD}^t \rceil$.

Sia ω_{CG}^t il valore di RLPM all'iterazione t-esima. Essendo disposti ad accettare bounds duali meno stretti, per la monotonia di $\{\omega_{CG}^t\}$, la condizione di terminazione potrebbe essere posta come $\omega_{CG}^t - \omega_{LD}^t < \Delta$ ($\Delta \ge 0$).

Analisi di sensitività e preprocessing

Ottenuto un bound duale ω_{LR} ed una soluzione ammissibile Z^*, è possibile effettuare un'analisi di sensitività sulla soluzione parziale corrispondente ad ogni sottoproblema.

M_{IN} rappresenta l'insieme dei distributori di servizio scelti e M_{OUT} l'insieme dei centri di servizio non selezionati nel bound duale. j_{BO} e j_{WI} sono rispettivamente il migliore tra i distributori di servizio non selezionati (ovvero quello con costo ridotto più negativo) e il peggiore tra i centri di servizio scelti:

$$j_{BO} = \arg\min_{j \in M_{OUT}} \{\zeta_j\}$$

$$j_{WI} = \arg\max_{j \in M_{IN}} \{\zeta_j\}.$$

Nell'algoritmo originario per CPMP l'analisi di sensitività viene condotta nel seguente modo:

- se $\exists j \left| \zeta_j > 0, \quad \omega_{LR} + \zeta_j - \zeta_{j_{WI}} \geq Z^* \right.$, imponendo $y_j = 1$ si otterrebbe un bound duale superiore al bound primale. Quindi, per poter ottenere soluzioni migliori del bound primale, deve necessariamente essere $y_j = 0$ (e quindi $x_{ij} = 0 \quad \forall i \in N$).
- se $\exists j \left| \zeta_j < 0, \quad \omega_{LR} - \zeta_j + \zeta_{j_{WI}} \geq Z^* \right.$, imponendo $y_j = 0$ si otterrebbe, come prima, un bound duale superiore al bound primale. Quindi necessariamente $y_j = 1$.

Essendo presente il vincolo di cardinalità, ogniqualvolta si valuta la rimozione o l'inserimento di una mediana è necessario considerare il migliore candidato ad entrare (j_{BO}) o uscire (j_{WI}) dall'insieme delle mediane scelte.

Nell'algoritmo presentato in questo lavoro, l'analisi di sensitività viene condotta valutando lo stato del vincolo di cardinalità e il costo ridotto delle mediane non selezionate.

Per valutare il contributo dell'inserimento di una facility è necessario sottrarre il costo ridotto della peggiore tra le mediane scelte (j_{WI}) solo nel caso in cui il vincolo di cardinalità sia attivo, ovvero sono stati scelti esattamente p clusters.
Quando si valuta la rimozione di una facility, bisogna sottrarre il costo ridotto della migliore tra le mediane non selezionate (j_{BO}), solamente se ha costo ridotto negativo ($\zeta_{j_{BO}} < 0$).

Sia S l'insieme dei centri di servizio fissati scelti dall'analisi, e F l'insieme contenente i centri di servizio candidati proibiti attraverso l'analisi. Si definisca C, variabile binaria che vale "vero" se è presente il vincolo di cardinalità e tale vincolo è stringente. L'analisi viene condotta come mostrato in Figura A.

Pseudo-codice analisi di sensitività

Input: C, ω_{LR}, Z^*, $\zeta_j \quad \forall j \in M$, M_{IN}, M_{OUT}, F, S.

Output: F, S.

```
BEGIN
∀j ∈ M_OUT
    if(C̄)
        if(ω_LR + ζ_j ≥ Z*)
            F = F ∪ {j}
        (endif)
    else
        if(ω_LR + ζ_j − ζ_jWI ≥ Z*)
            F = F ∪ {j}
        (endif)
    (endif)
(done)

∀j ∈ M_IN
    if(ζ_jBO ≥ 0)
```

```
        if(ω_LR − ζ_j ≥ Z*)
            S = S ∪ {j}
        (endif)
    else
        if(ω_LR − ζ_j + ζ_jW1 ≥ Z*)
            S = S ∪ {j}
        (endif)
    (endif)
(done)
END
```

Figura A – Pseudo-codice analisi di sensitività

L'analisi richiede tempo $O(M)$, dato che i valori ζ_j sono già stati determinati durante il processo di column generation.

4.4 Euristica primale

Individuare soluzioni buone è importante in quanto tali soluzioni possono essere utilizzate come bounds primali da algoritmi di enumerazione implicita per raggiungere più rapidamente l'ottimo del problema.

Nel nostro algoritmo abbiamo utilizzato un'euristica a due fasi combinando la procedura HEUR1 sviluppata da Pirkul per il CCLP [Pirk87], ad una versione modificata dell'euristica MTH di Martello e Toth (1982) [MT89] per il problema dell'assegnamento generalizzato già utilizzata da diversi autori, come ad esempio Savelsbergh [Save97].

Fase 1: Scelta delle facilities

La scelta del numero di *facilities* da utilizzare e della loro localizzazione dipende fortemente dal rapporto tra costi di apertura (f_j) e costi di accesso (d_{ij}). Se i costi di apertura dominano totalmente i costi di accesso, verranno utilizzate solo un

minimo numero di *facilities*. Al contrario se i costi di apertura sono molto vicini a zero, il problema diventa di assegnare ogni nodo alla *facility* più desiderabile.

Viene definito *fattore di carico* (r) il rapporto tra la somma delle domande dei nodi assegnati a una facility e la sua capacità. (r). Se il fattore di carico è vicino a 1, allora la procedura sceglie solo un numero minimo di *facility* cercando di utilizzare il massimo della loro capacità. Viceversa se il fattore di carico è inferiore, allora il numero di *facility* scelte dalla procedura aumenta. Variando il fattore di carico si possono generare soluzioni con un diverso rapporto tra costi di apertura e costi di accesso.

Poiché il fattore di carico di una soluzione non è noto a priori, si generano un certo numero di soluzioni assumendo un diverso livello di rapporto e si sceglie la migliore.

Nella fase di scelta delle facilities dell'euristica proposta, ogni Y^t rappresenta l'insieme dei siti in cui è stata aperta una facility, quando il fattore di carico è $t/10$, N_w rappresenta l'insieme dei nodi non ancora assegnati e M_w è l'insieme dei distributori che non sono stati scelti. Lo pseudo-codice relativo alla prima fase dell'euristica è riportato in Figura B.

La procedura *assign* (Figura C) si occupa di assegnare i nodi al sito candidato, in accordo con il fattore di carico r. $N^{j'}$ è l'insieme dei nodi assegnabili e $Q_{j'}^R$ esprime il carico temporaneo del centro di servizio j'.

Pseudo-codice della scelta delle facilities

Input: N, M, $\psi_j \quad \forall j \in M$.

Output: $Y^t \quad \forall t = 1,...,10$.

BEGIN

$r = 1/10$

$t = 1$

while $t \leq 10$ do

$Y^t = \varnothing$

$N_W = N$

$M_W = M$

while $N_W \neq \varnothing$ do

$j' = \arg\max_{j \in M_w} \{\psi_j\}$

$M_W = M_W \setminus \{j'\}$

assign(j')

$Y^t = Y^t \cup \{j'\}$

(done)

$t = t + 1$

$r = t/10$

(done)

END

Figura B – Pseudo-codice della scelta delle facilities

Pseudo-codice per funzione di assegnamento nodi

Input: N_W, $w_i \quad \forall i \in N$, $Q_j \quad \forall j \in M$, r, $d_{ij} \quad \forall i \in N \quad \forall j \in M$.

Output: N_W.

assign(j'):

BEGIN

$N^{j'} = N_W$

$Q_{j'}^R = 0$

while $N^{j'} \neq \varnothing$ AND $Q_j^R + \min_{i \in N_W} \{w_i\} / Q_{j'} \leq r$ do

$i' = \arg\min_{i \in N^{j'}} \{d_{ij'}\}$

if $Q_{j'}^R + w_{i'} / Q_{j'} \leq r$

$N_W = N_W \setminus \{i'\}$

$Q_{j'}^R = Q_{j'}^R + w_{i'}$

(endif)

$N^{j'} = N^{j'} \setminus \{i'\}$

(done)
END

Figura C – Pseudo-codice per funzione di assegnamento nodi

La fase di selezione delle mediane ha complessità $O(NM \log N)$.

Nel caso di problemi su concentratori, la fase di inizializzazione di *assign*

$N^{j'} = N_W$

$Q_{j'}^R = 0$

va sostituita con:

$N_W = N_W \setminus \{j'\}$

$N^{j'} = N_W$

$Q_{j'}^R = w_{j'}$

Criteri di ordinamento delle facilities

Per poter ottenere buone soluzioni è importante individuare dei criteri di valutazione delle *facilities* che consentano di ordinare al meglio i siti candidati.

Indice delle caratteristiche

All'inizio del processo di ottimizzazione non è disponibile alcuna informazione sui siti candidati ad eccezione dei dati del problema stesso. Un sito molto capace è preferibile; al contrario, siti con costi di apertura molto alti o molto distanti dai nodi, probabilmente non saranno contenuti in una soluzione buona in quanto ne peggiorerebbero di molto il valore.

Normalizzando il costo di apertura, la somma delle distanze dai nodi e la capacità di un sito, e combinando questi tre valori, è possibile ottenere un buon indice di valutazione iniziale delle *facilities*.

$$\psi_j = \frac{Q_j}{\max_j (Q_j)} - \frac{\sum_i d_{ij}}{\max_j (\sum_i d_{ij})} - \frac{f_j}{\max_j (f_j)} \quad \forall j \in M$$

Questo criterio di valutazione, poiché viene calcolato esclusivamente sulla base dei dati del problema, viene utilizzato per inizializzare il bound primale.

Valore frazionario di selezione

Un secondo criterio di ordinamento sfrutta l'informazione generata dall'RLMP, in quanto valuta il valore frazionario di selezione di ciascun sito. Tale indice viene calcolato a partire dai coefficienti l_{ij} che esprimono che frazione della domanda l nodo i è soddisfatta dalla facility j in una soluzione del RLMP. In particolare:

$$l_{ij} = \sum_{k \in Z^j} \gamma_j^k x_i^k \lambda_k \quad \forall i \in N \quad \forall j \in M$$

$$\psi_j = \sum_{i \in N} l_{ij} \quad \forall j \in M$$

Il coefficiente ψ_j così costruito consente di produrre una soluzione intera simile alla soluzione ottima dell'RLMP.

Fase 2: Allocazione

Scelte le mediane, il problema si riduce ad un assegnamento generalizzato: è possibile costruire una soluzione come nell'euristica MTH, proposta da Martello e Toth. Per alcune istanze è difficile raggiungere soluzioni ammissibili in questo modo.

Se la soluzione ottenuta è inammissibile, l'euristica di Martello Toth si arresta. Nella versione modificata si è scelto di ricorrere a *1-scambi*. Un simile algoritmo di ricerca locale non fornisce alcuna garanzia: si compiono un numero finito di iterazioni, nella speranza di ottenere una soluzione ammissibile.

Se la soluzione prodotta ai passi precedenti è ammissibile, si possono effettuare, come nell'euristica MTH, scambi miglioranti.

Per una spiegazione più dettagliata dell'euristica MTH modificata, si rimanda a [Cese05]

4.5 Enumerazione implicita

In questa sezione sono presentate la strategia di ricerca nell'albero e la regola di *branching* adottate in questo lavoro.

Strategia di ricerca Best First

Ricorrendo ad una politica di esplorazione *best first* dell'albero di branching si dà precedenza alla valutazione di sottoproblemi dal bound duale più promettente. Tramite l'euristica primale è possibile ottenere soluzioni ammissibili buone partendo dalle soluzioni parziali descritte dai nodi più promettenti, e quindi *probabilmente* vicine all'ottimo. Adottando questa strategia è possibile rimuovere da RMP clusters aventi costo ridotto superiore alla differenza tra bound primale e bound duale, poiché sicuramente svantaggiosi.
Lo svantaggio di questa strategia di ricerca è che non è possibile risolvere il rilassamento lineare di un sottoproblema partendo dalla base del padre senza memorizzare esplicitamente le colonne in base all'ottimo. Inoltre, in generale, alla valutazione di un sottoproblema segue quella di un sottoproblema in cui sono state fissate variabili differenti (e quindi con struttura differente).

Branching di tipo "forbid"

La scelta della regola di branching in un algoritmo ad enumerazione implicita è molto importante in quanto l'aggiunta di vincoli può cambiare la struttura del problema di pricing.

Sia, per ogni nodo i, nel problema avente insieme di soluzioni S ed insieme di candidati mediane M^S

$$H_i = \left\{ j \in M^S \middle| x_{ij} \neq 0 \right\} = \left\{ h_i^1 \ldots h_i^{F_i} \right\}$$

l'insieme degli indici delle mediane per le quali non è stato impedito l'assegnamento. In generale, inizialmente $H_i = M \quad \forall i \in N$, successivamente, ponendo condizioni del tipo $x_{ij'} = 0$ è necessario rimuovere j' da H_i.

Come proposto da Savelsbergh in [Save97] è possibile ottenere uno schema di branching binario

1. selezionando un nodo i^b per cui $H_{i^b} \neq \varnothing$
2. selezionando un intero $j^* \simeq \frac{\left|H_{i^b}\right|}{2}$
3. ponendo

$$S_l = S_0 \cap \left\{ \left(x^j, y_j\right) \middle| \sum_{k=1}^{j^*} x_{i^b h_{i^b}^k} = 0 \right\}$$

e

$$S_r = S_0 \cap \left\{ \left(x^j, y_j\right) \middle| \sum_{k=j^*+1}^{\left|H_{i^b}\right|} x_{i^b h_{i^b}^k} = 0 \right\}$$

Si assicura che i^b non venga assegnato alle prime j^* mediane in H_{i^b} (S_l) o alle rimanenti $\left|H_{i^b}\right| - j^*$ (S_r). La scelta può essere specializzata per i problemi multirisorse, definendo M_{i^b} come l'insieme delle mediane a cui il nodo i^b è associato nella soluzione di L-RMP e partizionandolo in due insiemi $M_{i^b}^l$ e $M_{i^b}^r$, in modo che $\left|M_{i^b}^l\right| \simeq \frac{\left|M_{i^b}\right|}{2}$ e $M_{i^b}^r = M_{i^b} \setminus M_{i^b}^l$. Anche l'insieme $R_{i^b} = H_{i^b} \setminus M_{i^b}$ può essere partizionato nello stesso modo negli insiemi $R_{i^b}^r$ ed $R_{i^b}^l$:

$$S_l = S_0 \cap \left\{ \left(x^j, y_j\right) \middle| \sum_{k \in M_{i^b}^l \cup R_{i^b}^l} x_{i^b k} = 0 \right\}$$

e

$$S_r = S_0 \cap \left\{ \left(x^j, y_j\right) \middle| \sum_{k \in M_{i^b}^r \cup R_{i^b}^r} x_{i^b k} = 0 \right\}$$

4.6 Column Management

La politica di gestione delle colonne nel RLPM deve essere pianificata con attenzione in quanto scelte sbagliate producono algoritmi poco efficienti e con richieste di memoria troppo elevate. Di seguito sono esposti i criteri di gestione utilizzati in questa tesi.

Scelta delle colonne iniziali

Come spiegato nel capitolo 3, per dare inizio al processo di column generation è necessario che in RLPM vi sia un insieme di colonne che costituiscano una soluzione di base ammissibile; questo problema può essere risolto inserendo in RLPM un opportuno insieme di colonne *dummy*.

Viene, quindi, inserita una colonna che identifica un cluster contenente tutti i nodi:

$$\left(\underbrace{1,\ldots,1}_{N} \right)^T$$

Nel caso di problemi su mediane o facility, si aggiungono M ulteriori colonne:

$$\left(\underbrace{1,\ldots,1}_{N}, \quad \gamma_1^j, \gamma_2^j, \ldots, \gamma_M^j \right)^T, \ \forall j \in M$$

con

$$\gamma_j^j = 1 \text{ e } \gamma_k^j = 0 \quad \forall k \in M, k \neq j .$$

Le colonne *dummy* devono avere costo molto alto per garantire che non siano mai utilizzate nella soluzione ottima; per far ciò è sufficiente porre $c^j \geq \sum_{j \in M} \sum_{i \in N} d_{ij} + \sum_{j \in M} f_j$. Bisogna inoltre conservare questo insieme di colonne nel RLPM di ogni sottoproblema.

Come suggerito da Savelsbergh in [Save97], abbiamo aggiunto al RLPM iniziale alcune colonne generate in modo euristico utilizzando come criterio di ordinamento gli indici delle caratteristiche delle facility: avere a disposizione sin dalle prime iterazioni di column generation un set di colonne che rappresentino dei clusters vantaggiosi riduce notevolmente i tempi di calcolo.

Inserimento di colonne

La soluzione dei problemi di knapsack (uno per ogni potenziale mediana $j \in M$) consente di individuare, ad ogni iterazione, fino a M colonne di costo ridotto negativo. In questo lavoro si è scelto di inserire in RLPM tutte le colonne di costo ridotto negativo generate. Questo criterio di inserimento consente di ridurre il

numero di iterazioni di column generation complessivamente necessarie per la convergenza, a scapito delle dimensione di RLPM.

Rimozione di colonne

Inserendo di volta in volta nuove colonne, il numero di variabili in RLPM può diventare troppo elevato per poter essere gestibile efficientemente e, in ogni caso, dover risolvere il rilassamento lineare di problemi dalle dimensioni elevate può peggiorare sensibilmente le prestazioni dell'algoritmo.

Per contro, avere a disposizione un insieme di clusters "molto buoni" prima di dare inizio alla generazione di nuove colonne può accelerare la convergenza all'ottimo del rilassamento lineare, riducendo il numero di iterazioni di column generation, e quindi il numero di chiamate all'algoritmo per la soluzione del pricing problem. La cancellazione di clusters da RLPM deve essere regolata con cura: in generale, politiche che prevedano un compromesso tra i due aspetti possono fornire le prestazioni migliori, anche se il corretto bilanciamento dei tempi di calcolo dipende dall'efficienza relativa del MIP solver utilizzato per la soluzione del rilassamento lineare e dall'algoritmo per il pricing problem.

In questo lavoro si è scelto di valutare la rimozione di una colonna in funzione del suo costo ridotto. Se il costo ridotto di una colonna è molto positivo, difficilmente tale colonna può, nelle iterazioni successive, entrare in base. Una possibile politica è la rimozione delle colonne aventi, al termine delle iterazioni di column generation, costo ridotto maggiore di una soglia prefissata. Inoltre, più il bound duale si avvicina al primale, minore è il numero di clusters candidati ad entrare in base: se il costo ridotto di una colonna risulta maggiore del gap tra bound primale e bound duale in un problema, il cluster corrispondente non è candidato a far parte della soluzione ottima per i sottoproblemi del problema in esame. Invece di confrontare i costi ridotti delle colonne con un valore costante, si conservano in RLPM solo le colonne aventi costo ridotto inferiore ad una soglia prefissata $s(Z^*, \omega_{CG})$, funzione della differenza tra bound primale (Z^*) e bound duale (ω_{CG}).

5. Esperimenti

5.1 Strumenti e dati utilizzati

Per valutare le prestazioni dell'algoritmo realizzato sono state considerate 20 istanze, già utilizzate in [Osma94], [Bald02] e [Cese05]. I primi 10 problemi sono definiti su grafi di 50 nodi (classe A), nei rimanenti 10 problemi i grafi hanno 100 nodi (classe B). Per i problemi con vincoli di cardinalità è stato posto $p = \frac{N}{10}$, per cui $p_A = 5$ e $p_B = 10$. In entrambi le classi di istanza i costi di apertura per i centri di servizio sono uniformi: $f_j = 120 \quad \forall j \in M$.

L'algoritmo è stato sviluppato secondo il paradigma della programmazione orientata agli oggetti. Il codice è stato scritto in C++ e compilato con gcc/g++, con impostazioni di ottimizzazione di terzo livello. Per risolvere i problemi di PL è stato utilizzato CPLEX 8.1. I test sono stati condotti su PC con processore Pentium 4 da 1,6 GHz, dotato di 500MB di RAM, sistema operativo Linux (distribuzione RedHat). Per tutti gli esperimenti sono stati posti dei limiti al tempo e alla memoria a disposizione: le istanze vengono terminati dopo 1800 secondi di tempo di CPU e hanno a disposizione al più 500MB di memoria. Il MIP solver generico ILOG CPLEX versione 8.1 è stato utilizzato come termine di paragone.
Nelle sezioni seguenti si utilizza la sigla **MIP-S** per identificare il MIP solver CPLEX, e **BP-A** per far riferimento all'algoritmo branch & price.

5.2 Confronto tra Branch & Price e MIP-Solver

Nella Tabella A viene presentato un confronto tra le prestazioni del MIP solver e dell'algoritmo branch & price. La prima colonna mostra i problemi su cui sono stati condotti gli esperimenti. Per ogni problema vengono riportati i valori medi per il gap (differenza in percentuale tra bound primale e bound duale; $gap = \frac{PB-DB}{PB}$), il tempo di risoluzione nei test condotti e la percentuale di istanze risolte. La terza e la quarta colonna contengono, rispettivamente, i risultati sulla classe A e sulla classe B di istanze.

La prima cosa che si nota osservando i risultati nella Tabella A è che MIP-S ha prestazioni migliori negli esperimenti con 50 nodi; viceversa BP-A eccelle nelle istanze su 100 nodi. L'algoritmo BP-A si comporta meglio all'aumentare della complessità del problema; le prestazioni su istanze con pochi nodi sono peggiori in quanto vi è un overhead di inizializzazione dovuto al fatto che sono state implementate strutture dati piuttosto complesse.

Pur comportandosi meglio con il CPMP (si ricorda che l'algoritmo originario era stato sviluppato appositamente per questa classe di problemi) l'algoritmo branch & price dimostra di essere robusto, infatti il gap è piuttosto contenuto in tutte le tipologie di problema.

L'aggiunta di costi fissi incrementa la complessità dei problemi per entrambi i risolutori; la variante concentratori, invece, agevola il processo risolutivo al MIP solver, mentre penalizza l'algoritmo branch & price.

5.3 Confronto tra modelli

Le tabelle B e C mostrano i valori medi dei risultati dei test condotti con BP-A rispettivamente sulle istanze della Classe A e della Classe B. In particolare nelle colonne vengono presentati i valori, sia per la radice che per l'albero di ricerca completo, delle iterazioni di column generation effettuate, del numero di colonne generate, del gap e del tempo impiegato; le ultime due colonne contengono il numero di nodi valutati complessivamente e la percentuale di istanze risolte.

Osservando questi risultati è possibile desumere l'influenza che l'aggiunta del vincolo di cardinalità, dei costi di apertura o l'appartenenza di una centro di

servizio al proprio cluster (variante concentratori) hanno sulle prestazioni dell'algoritmo.

Mediamente l'aggiunta del vincolo di cardinalità migliora i tempi dell'algoritmo. Confrontando i risultati si nota che l'aggiunta del vincolo di cardinalità riduce il numero di iterazioni di column generation ma aumenta le colonne generate nel nodo radice. Pur non causando variazioni sostanziali nel gap, l'aggiunta di questo vincolo semplifica la risoluzione di LPM.

Il parametro che influisce maggiormente è il costo di apertura delle facilities. L'aggiunta dei costi di apertura dei centri di servizio complica i problemi a tal punto da portare la percentuale di istanze della classe B risolte dal 60%-70% al 20%. Nonostante questa discrepanza, il gap rimane sostanzialmente invariato.

Imporre che ciascun centro di servizio appartenga al cluster che descrive comporta un notevole peggioramento sia nei tempi di risoluzione, sia nei gap ottenuti. Nelle istanze utilizzate per condurre i test si ha $d_{jj} = 0 \quad \forall j \in M$; per verificare meglio le prestazioni dell'algoritmo su problemi con concentratori sarebbe utile realizzare i test su istanze in cui $d_{jj} \geq 0 \quad \forall j \in M$.

Nelle tabelle D, E, F, G, H e I, vengono presentati nel dettaglio, per ogni problema, i risultati degli esperimenti condotti con l'algoritmo branch & price. Queste tabelle sono composte da tre blocchi: il primo blocco indica il tipo di istanze, il secondo contiene i risultati sperimentali per il nodo radice mentre il terzo blocco fa riferimento all'intero albero di ricerca.

Per il nodo radice vengono riportati il numero di iterazioni di column generation necessarie per raggiungere l'ottimo (CG it.), il numero di colonne generate (cols), il miglior upper bound trovato (UB), il valore di LPM (LB), il gap tra bound primale e bound duale (gap) e il tempo di CPU utilizzato (cpu).

Per l'albero di ricerca vengono riportati il numero medio di iterazioni di column generation e il numero medio di colonne generate in ogni nodo dell'albero di ricerca, l'upper bound e il lower bound finali (F.UB e F.LB), l'approssimazione ottenuta (gap), il tempo di CPU utilizzato per l'ottimizzazione (cpu) e il numero di nodi valutati nell'albero di ricerca (ev.nodes).

Osservando i valori all'ottimo si può notare che, per le istanze utilizzate, l'utilizzo della variante concentratori non comporta alcuna variazione. Inoltre vi sono due istanze (istanze 1 e 10 della Classe A) in cui il vincolo di cardinalità è stringente e comporta un peggioramento nel valore della soluzione ottima.
A fronte di queste considerazioni si ha che il problema la cui risoluzione ha le prestazioni migliori è il CPMP; in presenza di costi fissi di apertura è invece preferibile formulare il problema come SS-CPFLP

5.4 Conclusioni

In questo tesi è stata esposta nel dettaglio la realizzazione di un framework in grado di trattare in maniera uniforme problemi di localizzazione varianti del *Single Source Capacitated Facility Location Problem*, a partire da un algoritmo branch & price per il *Capacitated P-Median Problem*. Per trattare tutte le classi di problema è stato necessario generalizzare diverse componenti del metodo originario. I risultati mostrano che il metodo utilizzato è robusto e consente di ottenere gap inferiori all'1% in tutte le classi di problema valutate.
La struttura dell'algoritmo consente di adattarlo per problemi con vincoli aggiuntivi, quali i *regional constraints* [Murr97]. Si renderebbe necessario, oltre alle relative modifiche a IPM, lo sviluppo di un'euristica primale che consenta di individuare soluzioni buone ammissibili.

		Classe A		Classe B	
		MIP-S	**BP-A**	**MIP-S**	**BP-A**
SS-CFLP	*gap medio*	0,00%	0,00%	3,94%	0,56%
	tempo (s)	36,519	7,209	-	72,57
	perc. istanze risolte	100,00%	100,00%	0,00%	20,00%
CCLP	*gap medio*	0,00%	0,09%	1,79%	0,69%
	tempo (s)	14,725	23,253	838,45	43,73
	perc. istanze risolte	100,00%	90,00%	10,00%	20,00%
CPMP	*gap medio*	0,00%	0,09%	4,55%	0,30%
	tempo (s)	34,553	6,426	-	119,56
	perc. istanze risolte	100,00%	90,00%	0,00%	60,00%
CPCLP0	*gap medio*	0,00%	0,31%	0,82%	0,41%
	tempo (s)	40,749	5,183	686,99	325,79
	perc. istanze risolte	100,00%	80,00%	60,00%	70,00%
SS-CPFLP	*gap medio*	0,00%	0,01%	3,68%	0,46%
	tempo (s)	65,763	2,476	-	45,89
	perc. istanze risolte	100,00%	90,00%	0,00%	20,00%
CPCLP	*gap medio*	0,00%	0,12%	1,53%	0,53%
	tempo medio (s)	29,996	3,492	1767,11	65,30
	perc. istanze risolte	100,00%	80,00%	10,00%	20,00%

Tabella A - Confronto tra Branch & Price e MIP Solver

Instance	**Root**				**Search Tree**					
Classe A	*CG it.*	*cols*	*gap*	*time (s)*	*CG it.*	*cols*	*gap*	*time (s)*	*ev.nodes*	*inst.solved*
SS-CFLP	11,00	1494,10	0,47%	1,24	13,00	177,60	0,00%	7,21	50,80	100,00%
CCLP	11,10	1531,90	0,46%	1,34	11,40	139,00	0,09%	25,84	353,90	90,00%
CPMP	8,80	3797,40	1,03%	2,09	9,20	109,60	0,09%	6,43	677,00	90,00%
CPCLP0	8,60	3636,00	1,02%	1,79	10,55	122,18	0,31%	5,18	625,80	80,00%
SS-CPFLP	6,90	2790,70	0,60%	0,63	9,10	108,60	0,01%	2,48	1236,00	90,00%
CPCLP	6,80	2693,90	0,59%	0,84	9,30	114,60	0,12%	3,49	803,70	80,00%

Tabella B - Confronto tra modelli sulle istanze della classe A

Instance	**Root**				**Search Tree**					
Classe B	*CG it.*	*cols*	*gap*	*time (s)*	*CG it.*	*cols*	*gap*	*time (s)*	*ev.nodes*	*inst.solved*
SS-CFLP	17,00	3889,50	1,13%	11,65	18,10	536,40	0,56%	72,57	2595,80	20,00%
CCLP	17,10	4103,10	1,18%	11,92	21,00	592,30	0,69%	43,73	2078,60	20,00%
CPMP	13,30	9235,60	1,15%	11,80	15,70	391,20	0,30%	119,56	557,10	60,00%
CPCLP0	11,00	6525,90	1,15%	8,28	19,50	504,30	0,41%	325,79	586,90	70,00%
SS-CPFLP	10,60	5591,50	1,03%	3,73	17,40	440,70	0,46%	45,89	2658,40	20,00%
CPCLP	10,80	6056,70	1,08%	4,09	19,80	521,50	0,53%	65,30	2327,70	20,00%

Tabella C - Confronto tra modelli sulle istanze della classe B

Tabella D –Risultati di BP-A su SS-CFLP

Instance		Root						Search Tree						
N	P	CG it.	cols	UB	LB	gap	time (s)	CG it.	cols	F.UB	F.LB	gap	time (s)	ev.nodes
50,00	5,00	9,00	1212,00	1311,00	1300,33	0,81%	0,706	18,00	257,00	1311	1311,00	0,00%	5,000	29,00
		3,00	480,00	1340,00	1340,00	0,00%	0,389	0,00	0,00	1340	1340,00	0,00%	0,420	1,00
		11,00	1876,00	1351,00	1349,00	0,15%	1,060	10,00	236,00	1351	1351,00	0,00%	1,500	3,00
		2,00	267,00	1251,00	1250,98	0,00%	0,252	0,00	0,00	1251	1251,00	0,00%	0,300	1,00
		22,00	3728,00	1264,00	1260,50	0,28%	1,774	41,00	593,00	1264	1264,00	0,00%	5,050	5,00
		2,00	244,00	1378,00	1377,68	0,02%	0,316	0,00	0,00	1378	1378,00	0,00%	0,360	1,00
		16,00	2119,00	1387,00	1378,25	0,63%	1,902	13,00	163,00	1387	1387,00	0,00%	5,510	25,00
		14,00	1699,00	1394,00	1370,72	1,67%	1,578	16,00	197,00	1391	1391,00	0,00%	31,450	209,00
		18,00	1912,00	1315,00	1312,02	0,23%	2,624	18,00	209,00	1315	1315,00	0,00%	4,770	9,00
		13,00	1404,00	1417,00	1403,85	0,93%	1,755	14,00	121,00	1417	1417,00	0,00%	17,730	225,00
	Average	11,00	1494,10			0,47%	1,236	13,00	177,60			0,00%	7,209	50,80
100,00	10,00	18,00	4418,00	2213,00	2189,66	1,05%	10,619	0,00	0,00	2213	2213,00	0,00%	1801,000	2,00
		21,00	4968,00	2145,00	2120,06	1,16%	14,545	25,00	1240,00	2145	2121,38	1,10%	1801,000	5,00
		22,00	5160,00	2226,00	2219,36	0,30%	15,153	24,00	515,00	2226	2226,00	0,00%	93,680	63,00
		15,00	3472,00	2186,00	2149,82	1,66%	9,612	19,00	560,00	2186	2165,78	0,92%	1801,000	3820,00
		17,00	3895,00	2276,00	2271,09	0,22%	11,362	16,00	292,00	2275	2275,00	0,00%	51,450	55,00
		17,00	3686,00	2157,00	2137,49	0,90%	12,203	21,00	526,00	2154	2148,13	0,27%	1801,000	4574,00
		17,00	3945,00	2234,00	2208,80	1,13%	11,263	19,00	513,00	2234	2222,68	0,51%	1801,000	3917,00
		15,00	3433,00	2251,00	2210,33	1,81%	10,296	21,00	721,00	2251	2226,22	1,10%	1801,000	2221,00
		14,00	2918,00	2237,00	2215,62	0,96%	11,220	17,00	431,00	2237	2230,10	0,31%	1801,000	5815,00
		14,00	3000,00	2220,00	2172,39	2,14%	10,203	19,00	566,00	2220	2188,75	1,41%	1801,000	5486,00
	Average	17,00	3889,50			1,13%	11,648	18,10	536,40			0,56%	1455,313	2595,80
	Overall average	14,00	2691,80			0,80%	6,442	15,55	357,00			0,28%	731,261	1323,30

Tabella E –Risultati di BP-A su CCLP

Instance		Root						Search Tree						
N	P	CG it.	cols	UB	LB	gap	time (s)	CG it.	cols	F.UB	F.LB	gap	time (s)	ev.nodes
50,00	5,00	8,00	1257,00	1311,00	1300,33	0,81%	0,681	19,00	234,00	1311	1311,00	0,00%	15,510	85,00
		4,00	533,00	1340,00	1339,70	0,02%	0,669	0,00	0,00	1340	1340,00	0,00%	0,710	1,00
		18,00	2769,00	1351,00	1349,00	0,15%	1,901	12,00	153,00	1351	1351,00	0,00%	2,360	3,00
		3,00	462,00	1251,00	1251,00	0,00%	0,455	0,00	0,00	1251	1251,00	0,00%	0,490	1,00
		18,00	3075,00	1264,00	1260,50	0,28%	1,548	19,00	283,00	1264	1264,00	0,00%	60,010	3,00
		4,00	524,00	1378,00	1377,99	0,00%	0,662	0,00	0,00	1378	1378,00	0,00%	58,570	1,00
		14,00	1770,00	1387,00	1378,25	0,63%	1,742	13,00	156,00	1387	1387,00	0,00%	65,510	51,00
		13,00	1597,00	1392,00	1370,72	1,53%	1,537	22,00	253,00	1392	1380,00	0,86%	1801,000	3064,00
		17,00	1838,00	1315,00	1312,02	0,23%	2,679	15,00	171,00	1315	1315,00	0,00%	3,920	11,00
		12,00	1494,00	1417,00	1403,85	0,93%	1,508	14,00	140,00	1417	1417,00	0,00%	25,450	319,00
	Average	11,10	1531,90			0,46%	1,338	11,40	139,00			0,09%	203,353	353,90
100,00	10,00	20,00	5309,00	2213,00	2189,72	1,05%	11,463	22,00	692,00	2213	2202,19	0,49%	1801,000	2166
		18,00	3828,00	2139,00	2120,06	0,89%	12,435	28,00	691,00	2139	2130,02	0,42%	1801,000	1857
		17,00	4222,00	2226,00	2219,36	0,30%	11,080	19,00	371,00	2226	2226,00	0,00%	40,810	55
		17,00	3945,00	2188,00	2149,82	1,74%	12,693	18,00	572,00	2188	2164,22	1,09%	1801,000	3003
		17,00	4137,00	2281,00	2271,09	0,43%	11,056	17,00	408,00	2275	2275,00	0,00%	46,650	61
		17,00	4057,00	2157,00	2137,49	0,90%	13,418	27,00	551,00	2157	2144,59	0,58%	1801,000	2502
		19,00	4766,00	2242,00	2208,80	1,48%	13,387	22,00	827,00	2242	2217,20	1,11%	1801,000	1784
		17,00	4067,00	2251,00	2210,33	1,81%	11,677	20,00	711,00	2251	2225,67	1,13%	1801,000	2293
		14,00	3168,00	2236,00	2215,62	0,91%	11,413	17,00	474,00	2236	2227,13	0,40%	1801,000	3596
		15,00	3532,00	2223,00	2172,58	2,27%	10,608	20,00	626,00	2223	2186,09	1,66%	1801,010	3469
	Average	17,10	4103,10			1,18%	11,923	21,00	592,30			0,69%	1449,547	2078,60
	Overall average	14,10	2817,50			0,82%	6,631	16,20	365,65			0,39%	826,450	1216,25

Tabella F –Risultati di BP-A su CPMP

Instance		Root						Search Tree						
N	P	CG it.	cols	UB	LB	gap	time (s)	CG it.	cols	F.UB	F.LB	gap	time (s)	ev.nodes
50,00	5,00	11,00	3926,00	713,00	705,00	1,12%	2,016	20,00	160,00	713	713,00	0,00%	9,060	23,00
		5,00	3651,00	740,00	739,70	0,04%	1,669	0,00	0,00	740	740,00	0,00%	1,760	1,00
		13,00	5433,00	751,00	749,00	0,27%	2,974	15,00	247,00	751	751,00	0,00%	4,550	3,00
		7,00	4006,00	651,00	651,00	0,00%	1,557	0,00	0,00	651	651,00	0,00%	1,650	1,00
		5,00	3336,00	664,00	663,85	0,02%	1,431	0,00	0,00	664	664,00	0,00%	1,510	1,00
		4,00	3040,00	778,00	777,54	0,06%	1,052	0,00	0,00	778	778,00	0,00%	1,130	1,00
		11,00	3586,00	787,00	778,25	1,11%	2,236	13,00	159,00	787	787,00	0,00%	17,380	45,00
		9,00	3060,00	821,00	771,67	6,01%	1,956	21,00	254,00	821	814,00	0,85%	1801,000	6595,00
		12,00	4172,00	715,00	712,40	0,36%	2,794	13,00	182,00	715	715,00	0,00%	4,060	5,00
		11,00	3764,00	829,00	817,87	1,34%	3,198	10,00	94,00	829	829,00	0,00%	16,730	95,00
	Average	8,80	3797,40			1,03%	2,088	9,20	109,60			0,09%	185,883	677,00
100,00	10,00	13,00	11390,00	1006,00	1001,07	0,49%	14,359	16,00	324,00	1006	1006,00	0,00%	48,150	29,00
		13,00	9053,00	966,00	958,49	0,78%	11,449	18,00	354,00	966	966,00	0,00%	319,650	243,00
		13,00	8891,00	1026,00	1021,55	0,43%	10,407	41,00	1383,00	1026	1026,00	0,00%	47,920	7,00
		17,00	10963,00	984,00	971,75	1,24%	17,176	0,00	0,00	984	984,00	0,00%	1801,000	2,00
		17,00	9316,00	1092,00	1080,41	1,06%	14,283	0,00	0,00	1092	1088,37	0,33%	1801,000	2,00
		12,00	9471,00	955,00	951,33	0,38%	11,149	14,00	315,00	954	954,00	0,00%	28,730	17,00
		12,00	9302,00	1034,00	1025,28	0,84%	11,143	16,00	250,00	1034	1034,00	0,00%	224,060	249,00
		14,00	10043,00	1045,00	1031,90	1,25%	13,979	21,00	501,00	1045	1041,88	0,30%	1801,000	1123,00
		10,00	7145,00	1033,00	1026,26	0,65%	8,969	14,00	314,00	1031	1031,00	0,00%	48,830	69,00
		12,00	6782,00	1018,00	973,65	4,36%	5,114	17,00	471,00	1018	990,00	2,75%	1801,010	3830,00
	Average	13,30	9235,60			1,15%	11,803	15,70	391,20			0,34%	792,135	557,10
	Overall average	11,05	6516,50			1,09%	6,95	12,45	250,40			0,21%	489,009	617,05

Tabella G – Risultati di BP-A su CPCLP0

Instance		Root						Search Tree						
N	P	CG it.	cols	UB	LB	gap	time (s)	CG it.	cols	F.UB	F.LB	gap	time (s)	ev.nodes
50,00	5,00	12,00	3951,00	713,00	705,00	1,12%	1,009	17,50	167,76	713	713,00	0,00%	11,280	71,00
		5,00	2908,00	740,00	739,99	0,00%	1,336	0,00	0,00	740	740,00	0,00%	1,430	1,00
		11,00	4221,00	751,00	749,00	0,27%	2,369	16,00	264,00	751	751,00	0,00%	4,080	5,00
		7,00	4370,00	651,00	650,47	0,08%	0,000	0,00	0,00	651	651,00	0,00%	1,950	1,00
		6,00	4309,00	664,00	663,91	0,01%	2,045	0,00	0,00	664	664,00	0,00%	2,130	1,00
		7,00	3766,00	778,00	777,86	0,02%	1,684	0,00	0,00	778	778,00	0,00%	1,770	1,00
		9,00	3411,00	787,00	778,25	1,11%	2,492	14,00	168,00	787	787,00	0,00%	14,070	45,00
		11,00	3204,00	820,00	771,67	5,89%	2,598	21,00	306,00	820	796,20	2,90%	1801,010	3695,00
		10,00	3170,00	715,00	712,40	0,36%	2,147	12,00	149,00	715	715,00	0,00%	4,750	13,00
		8,00	3050,00	829,00	817,87	1,34%	2,232	25,00	167,00	829	827,00	0,24%	1801,010	2425,00
	Average	8,60	3636,00			1,02%	1,791	10,55	122,18			0,31%	364,348	625,80
100,00	10,00	11,00	5881,00	1006,00	1001,07	0,49%	8,461	16,00	347,00	1006	1006,00	0,00%	74,790	63,00
		11,00	5991,00	966,00	958,49	0,78%	7,292	17,00	359,00	966	966,00	0,00%	344,450	351,00
		13,00	6284,00	1026,00	1021,55	0,43%	7,880	30,00	825,00	1026	1026,00	0,00%	44,300	17,00
		14,00	8103,00	984,00	971,75	1,24%	11,342	32,00	1041,00	984	976,38	0,77%	1801,000	587,00
		12,00	7941,00	1092,00	1080,41	1,06%	10,537	17,00	322,00	1091	1091,00	0,00%	1449,970	1155,00
		11,00	6264,00	955,00	951,33	0,38%	7,701	17,00	482,00	954	954,00	0,00%	23,920	17,00
		11,00	9396,00	1034,00	1025,28	0,84%	10,081	15,00	266,00	1034	1034,00	0,00%	304,750	381,00
		9,00	5876,00	1048,00	1031,90	1,54%	6,015	20,00	563,00	1048	1041,20	0,65%	1801,000	1205,00
		8,00	4779,00	1033,00	1026,26	0,65%	5,709	14,00	319,00	1031	1031,00	0,00%	38,320	47,00
		10,00	4744,00	1015,00	973,67	4,07%	7,781	17,00	519,00	1015	987,82	2,68%	1801,000	2046,00
	Average	11,00	6525,90			1,15%	8,280	19,50	504,30			0,41%	768,350	586,90
	Overall average	9,80	5080,95			1,09%	5,036	15,03	313,24			0,36%	566,349	606,35

Tabella H – Risultati di BP-A su SS-CPFLP

Instance		Root						Search Tree						
N	P	CG it.	cols	UB	LB	gap	time (s)	CG it.	cols	F.UB	F.LB	gap	time (s)	ev.nodes
50,00	5,00	10,00	3320,00	1313,00	1305,00	0,61%	0,682	17,00	183,00	1313	1313,00	0,00%	3,560	21,00
		2,00	1504,00	1340,00	1339,01	0,07%	0,187	0,00	0,00	1340	1340,00	0,00%	0,230	1,00
		12,00	4448,00	1351,00	1349,00	0,15%	0,934	17,00	240,00	1351	1351,00	0,00%	1,530	3,00
		2,00	2245,00	1251,00	1250,60	0,03%	0,194	0,00	0,00	1251	1251,00	0,00%	0,240	1,00
		4,00	2341,00	1264,00	1263,93	0,01%	0,466	0,00	0,00	1264	1264,00	0,00%	0,510	1,00
		4,00	2515,00	1378,00	1377,87	0,01%	0,395	0,00	0,00	1378	1378,00	0,00%	0,450	1,00
		8,00	3289,00	1387,00	1378,25	0,63%	0,695	14,00	182,00	1387	1387,00	0,00%	5,410	25,00
		11,00	2561,00	1421,00	1371,67	3,47%	1,099	20,00	224,00	1420	1418,33	0,12%	1801,000	12197,00
		9,00	3525,00	1315,00	1312,02	0,23%	0,817	12,00	154,00	1315	1315,00	0,00%	1,500	5,00
		7,00	2159,00	1429,00	1417,87	0,78%	0,801	11,00	103,00	1429	1429,00	0,00%	8,850	105,00
	Average	6,90	2790,70			0,60%	0,627	9,10	108,60			0,01%	182,328	1236,00
100,00	10,00	12,00	6314,00	2207,00	2189,66	0,79%	4,289	5,00	69,00	2207	2190,66	0,74%	1801,000	3,00
		10,00	5542,00	2139,00	2120,06	0,89%	3,446	22,00	408,00	2139	2135,82	0,15%	1801,000	3463,00
		11,00	5919,00	2226,00	2219,36	0,30%	3,795	17,00	340,00	2226	2226,00	0,00%	52,010	55,00
		11,00	5021,00	2182,00	2149,82	1,47%	3,960	18,00	515,00	2182	2165,37	0,76%	1801,000	3435,00
		12,00	6201,00	2286,00	2271,09	0,65%	4,263	18,00	568,00	2275	2275,00	0,00%	39,760	39,00
		10,00	6482,00	2154,00	2137,49	0,77%	3,653	21,00	475,00	2154	2147,94	0,28%	1801,000	4342,00
		11,00	5368,00	2234,00	2208,80	1,13%	3,717	20,00	543,00	2234	2222,26	0,53%	1801,000	3178,00
		11,00	5834,00	2247,00	2210,33	1,63%	3,717	20,00	670,00	2247	2226,00	0,93%	1801,000	2083,00
		9,00	5103,00	2234,00	2215,62	0,82%	3,250	15,00	327,00	2231	2229,66	0,06%	1801,000	4914,00
		9,00	4131,00	2214,00	2172,39	1,88%	3,178	18,00	492,00	2214	2189,11	1,12%	1801,000	5072,00
	Average	10,60	5591,50			1,03%	3,727	17,40	440,70			0,46%	1449,977	2658,40
	Overall average	8,75	4191,10			0,82%	2,177	13,25	274,65			0,23%	816,153	1947,20

Instance		Root						Search Tree						
N	P	CG it.	cols	UB	LB	gap	time (s)	CG it.	cols	F.UB	F.LB	gap	time (s)	ev.nodes
50,00	5,00	12,00	3208,00	1313,00	1305,00	0,61%	0,995	14,00	168,00	1313	1313,00	0,00%	3,100	19,00
		3,00	2539,00	1340,00	1340,00	0,00%	0,388	0,00	0,00	1340	1340,00	0,00%	0,420	1,00
		10,00	3427,00	1351,00	1349,00	0,15%	0,753	12,00	146,00	1351	1351,00	0,00%	1,080	3,00
		2,00	1743,00	1251,00	1250,11	0,07%	0,342	0,00	0,00	1251	1251,00	0,00%	0,390	1,00
		4,00	2405,00	1264,00	1264,00	0,00%	0,772	0,00	0,00	1264	1264,00	0,00%	0,580	1,00
		3,00	2425,00	1378,00	1377,51	0,04%	0,399	0,00	0,00	1378	1378,00	0,00%	0,440	1,00
		7,00	2297,00	1387,00	1378,25	0,63%	1,016	14,00	157,00	1387	1387,00	0,00%	9,120	57,00
		10,00	3591,00	1420,00	1371,67	3,40%	1,361	25,00	356,00	1420	1402,62	1,22%	1801,000	7782,00
		8,00	2688,00	1315,00	1312,02	0,23%	0,854	17,00	212,00	1315	1315,00	0,00%	1,890	7,00
		9,00	2616,00	1429,00	1417,87	0,78%	1,543	11,00	107,00	1429	1429,00	0,00%	14,410	165,00
	Average	6,80	2693,90			0,59%	0,842	9,30	114,60			0,12%	183,243	803,70
100,00	10,00	12,00	7034,00	2206,00	2189,72	0,74%	4,205	22,00	475,00	2206	2202,15	0,17%	1801,000	2238,00
		12,00	5608,00	2140,00	2120,06	0,93%	4,306	23,00	499,00	2140	2133,47	0,31%	1801,000	2760,00
		9,00	5790,00	2226,00	2219,36	0,30%	3,656	19,00	323,00	2226	2226,00	0,00%	79,220	119,00
		12,00	5510,00	2186,00	2149,82	1,66%	4,580	23,00	648,00	2186	2162,10	1,09%	1801,000	2646,00
		12,00	6459,00	2286,00	2271,09	0,65%	4,338	15,00	520,00	2275	2275,00	0,00%	51,370	79,00
		11,00	6898,00	2155,00	2137,49	0,81%	4,603	22,00	509,00	2155	2145,71	0,43%	1801,000	2982,00
		11,00	6380,00	2234,00	2208,80	1,13%	4,203	19,00	560,00	2234	2219,92	0,63%	1801,000	2669,00
		10,00	6244,00	2248,00	2210,33	1,68%	3,568	19,00	665,00	2248	2225,86	0,98%	1801,000	2272,00
		8,00	5560,00	2233,00	2215,62	0,78%	3,318	17,00	419,00	2233	2226,93	0,27%	1801,000	3906,00
		11,00	5084,00	2219,00	2172,58	2,09%	4,123	19,00	597,00	2219	2186,89	1,45%	1801,000	3606,00
	Average	10,80	6056,70			1,08%	4,090	19,80	521,50			0,53%	1453,859	2327,70
	Overall average	8,80	4375,30			0,83%	2,466	14,55	318,05			0,33%	818,551	1565,70

Tabella I – Risultati di BP-A su CPCLP

Bibliografia

[Ambr03] D. Ambrosino, A. Sciomachen, Solutions of the Capacitated Concentrator Location Problem in Distribution Networks, AIRO Conference, Conference Program, 33 (2003)

[Avel03] P. Avella, A. Sassano, I. Vasilév, Computational study of large scale p-median problems, Technical report (2003)

[Bald02] Baldacci, R., E. Hadjiconstantinou, V. Maniezzo, A. Mingozzi, A new method for solving capacitated location problems based on a set partitioning approach, *Computers & Operations Research*, 29, 365-386 (2002)

[Beas85] Beasley, J.E., A note on solving large p-median problems, *European Journal of Operational Research*, 21, 270-273 (1985)

[Bram95] J.B. Bramel, D. Simchi-Levi, A location based heuristic for the delivery problem, *Operation Research* 43, 649-660 (1995)

[Cese05] A. Ceselli, G. Righini, A branch-and-price algorithm for the capacitated p-median problem (accettato per la pubblicazione su *Networks*)

[Chri81] Christofides, N., J.E. Beasley, A tree search algorithm for the p-median problem, *European Journal of Operational Research*, 10, 196-204 (1981)

[Chri83] Christofides, N., J.E. Beasley, Extensions to a Lagrangean relaxation approach for the capacitated warehouse location problem, *European Journal of Operational* Research, 12, 19-28 (1983)

[Corn77] Cornueojols, G., M.L. Fisher, G.L. Nemhauser, Location of bank accounts to optimize float: an analytic study of exact and approximate algorithms, *Management Science*, 23, 8, 789-810 (1977)

[Corn91] Cornueojols, G., R. Sridharan, J.M. Thizy, A comparison of heuristics and relaxations for the capacitated plant location problem, *European Journal of Operational Research*, 50, 280-297 (1991)

[Cort03] M.J. Cortinhal, M.E. Captivo, Upper and lower bounds for the single source capacitated location problem, *European Journal of Operational Research* 151, 333–351 (2003)

[Dant60] Dantzig, G.B., P. Wolfe, Decomposition principle for linear programs, *Operations Research*, 8, 101-111 (1960)

[DLT] Mirchandani, P.B., The p-median problem and generalizations, in Discrete Location Theory, Wiley & Sons, New York (1990)

[Erle82] Van Roy, T.J., D. Erlenkotter, Dual-based procedure for dynamic facility location, *Management Science* 28, 10 (1982)

[FL] Zvi Drezner Ed., Facility Location, Springer Series in Operations Research, New York (1995)

[Galv79] Galvão R.D., A dual bounded algorithm for the p-median problem, *Operations Research*, 28, 5 (1979)

[Gold71] Goldman, A.J., Optimal center location in simple networks, *Transportation Science* 5, 212-221 (1971)

[Gold86] Golden, B., C. Skiscim, Using Simulated Annealing to solve routing and location problems, *Naval Research Logistic Quarterly*, 33, 261-179 (1986)

[Hanj85] Hanjoul, P., D. Peeters, A comparison of two dual-based procedures for solving the p-median problem, *European Journal of Operational Research*, 20, 387-396 (1985)

[Hans97] Hansen, P., Jaumard, B., Cluster analysis and mathematical programming, *Mathematical Programmin* 79, 191-215 (1997)

[Holm99] K. Holmberg, M. Rönnqvist, D. Yuan, An exact algorithm for the capacitated facility location problems with single sourcing, *European Journal of Operational Research* 113, 544-559 (1999)

[Kari79] Kariv, O., Sl.L. Hakimi, An algorithmic approach to network location problems. Part 2. The p-median., *SIAM Journal of applied Mathematics* 37, 539-560 (1979)

[Klin86] Klincewicz, J.G, H. Luss, A Lagrangean Relaxation heuristic for capacitated facility location with single-source constraints, *Journal of Operational Research Society*, 37, 5, 495-500 (1986)

[Kueh63] Kuehn, A.A., M.J. Hemburger, A heuristic program for locating warehouses, *Management Science*, 9, 643-666 (1963)

[Labb03] M. Labbé, H. Yaman, Polyhedral Analysis for Concentrator Location Problem, Optimization Online *(www.optimization-online.org*), digest 694 (2003)

[Lehr66] Feldman, E., F.A. Lehrer, T.L. Ray, Warehouse locations under continuous economies of scale, *Management Science*, 2 (1966)

[Lore04] L.A.N. Lorena, E.L.F. Senne, A column generation approach to capacitated p-median problems, *Computers & Operations Research*, 31, 863-876 (2004)

[Mani98] Maniezzo, V., A. Mingozzi, R. Baldacci, A Bionomic approch to the capacitated p-median problem, *Journal of Heuristics*, 4, 263-280 (1998)

[MT89] Martello S., P. Toth, Knapsack Problems – Algorithms and Computer Implementations, John Wiley & Sons (1989)

[Murr97] Murray, A.T., R.A. Gerrard, Capacitated service and regional constraints in location-allocation modelling, *Location Science*, 5, 2, 103-118 (1997)

[Mulv84] Mulvey, J.M., P. Beck, Solving capacitated clustering problems, *European Journal of Operational Research*, 18, 339-348 (1984)

[Naru77] Narula, S.C., U.I Ogbu, H.M. Samuelsson, An algorithm for the p-median problem, *Operations Research*, 25, 4 (1977)

[Neeb83] Neebe, A.W, M.R. Rao, An algorithm for the fixed charge assigning users to sources problem, *Journal of the Operational Research Society* 34, 11, 1107-1113 (1983)

[Osma94] Osman, I.H., N. Christofides, Capacitated clustering problems by hybrid Simulated Annealing and Tabu Search, *International Transactions in Operational Research* 13, 317-339 (1994)

[Pirk87] Pirkul, H., Efficient algorithms for the capacitated concentrator location problem, *Computers and Operations Research* 14, 3, 197-208 (1987)

[Pisi95] Pisinger, D., A minimal algorithm for the 0-1 knapsack problem, *Operational Research*, 46, 5, 758-767 (1995)

[Ronn99] M. Rönnqvist, S. Tragantalerngsa, J. Holt, A repeated matching heuristic for the single-source capacitated facility location problem, *European Journal of Operational Research* 116, 51-68 (1999)

[Save97] Savelsbergh, M., A branch-and-price algorithm for the generalized assignment problem, *Operations Research*, 45, 6 (1997)

[Srid95] Sridharan, R., The capacitated plant location problem, *European Journal of Operational Research* 87, 203-213 (1995)

[VRoy86] Van Roy, T.J., A cross decomposition algorithm for capacitated facility location, *Operations Research*, 34, 145-163 (1986)

[Wols98] Wolsey, L.A., Integer Programming, Wiley & Sons (1998)