

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI

Corso di Laurea in Matematica

Generazione di istanze difficili per problemi di impaccamento e routing

Tesi di Laurea in Ricerca Operativa

Relatore:

Chiar.mo Prof.

ANDREA LODI

Presentata da:

CLAUDIO GAMBELLA

Correlatrici:

Chiar.ma Prof.

ELENA LOLI PICCOLOMINI

Dr.

SOPHIE N. PARRAGH

I Sessione

Anno Accademico 2011/2012

*Alla mia famiglia acquisita
di amici matematici*

Indice

| | |
|---|------------|
| Introduzione | iii |
| 1 Vehicle Routing Problem: descrizione del problema | 1 |
| 1.1 Caratteristiche generali del VRP | 2 |
| 1.2 Capacitated Vehicle Routing Problem (CVRP) | 5 |
| 1.3 Bin Packing e Traveling Salesman Problem | 6 |
| 2 CVRP: Risoluzione con modelli ILP e algoritmi esatti | 11 |
| 2.1 Modelli matematici per CVRP | 12 |
| 2.1.1 Modello 1 | 12 |
| 2.1.2 Modello 2 | 14 |
| 2.1.3 Modello 3 a generazione di colonne | 17 |
| 2.2 Motivazioni del lavoro di tesi | 19 |
| 3 Generazione di istanze BPP difficili | 21 |
| 3.1 Generazione di istanze di perfect packing con un numero di soluzioni ottime limitato | 21 |
| 3.2 Risoluzione istanze con generazione di colonne | 27 |
| 3.3 Conteggio soluzioni non isomorfe di BPP tramite orbital branch- ing | 29 |
| 4 Generazione di istanze CVRP da istanze BPP e risoluzione | 31 |
| 4.1 Generazione di istanze CVRP da istanze BPP | 31 |
| 4.2 Risultati algoritmo Roberti | 36 |

| | |
|--|----|
| 4.3 Osservazioni conclusive sui risultati ottenuti | 38 |
|--|----|

| | |
|---------------------|-----------|
| Bibliografia | 41 |
|---------------------|-----------|

Introduzione

Il presente lavoro di tesi nasce dall'intento di verificare alcune ipotesi riguardo ad aspetti degli algoritmi esatti per la risoluzione di problemi relativi al Vehicle Routing Problem (VRP) che li rendono poco efficaci su certi tipi di istanze reali.

La classe di problemi VRP, descritta dettagliatamente nel Capitolo 1, occupa in maniera trasversale i settori del trasporto merci e della gestione di servizi. L'importanza applicativa del problema spiega quindi l'interesse della ricerca nello sviluppo di algoritmi che riescano a risolvere efficientemente un numero via via maggiore di istanze. Tale richiesta si reputa relativamente difficoltosa se si considera che, spesso, ci sono vincoli di varia natura difficili da esprimere ed imporre in un problema VRP.

Nella tesi, ci si è concentrati in particolare sul problema del Capacitated Vehicle Routing (CVRP), che consiste nel servire tramite veicoli, con capacità, i clienti, dotati di domande, di una rete stradale a costo minimo, rispettando le capacità dei veicoli. Ogni istanza di CVRP è quindi costituita da una componente di packing, ovvero di impaccamento ottimale dei clienti nei veicoli, e di routing, l'instradamento dei veicoli per servire tutti i clienti nella rete stradale minimizzando il costo.

Visto che gli algoritmi esatti presenti in letteratura si basano effettivamente sulla costruzione di routes, si è pensato che essi possano tenere meno in considerazione la parte di packing delle istanze CVRP. L'ipotesi è quindi che gli algoritmi esatti incontrino difficoltà di risoluzione sulle istanze in cui il numero di soluzioni ottime di Bin Packing è limitato.

Allo scopo di verificare la validità delle congettura, si è pensato di creare istanze di packing dotate di un numero limitato di soluzioni ottime ed aggiungere loro 3 schemi di coordinate planari che si diversificassero per una supposta difficoltà in termini di routing.

La generazione di istanze di Bin Packing dotate di poche soluzioni ottime non è un problema banale: in letteratura non è mai stato affrontato direttamente. Sono stati quindi condotti alcuni test cercando configurazioni di perfect packing come illustrato nel capitolo 3. Per le istanze di Bin Packing create sono presenti, attualmente, solamente delle indicazioni di difficoltà che dovrebbero essere verificate con un effettivo conteggio tramite tecniche di orbital branching, che sono ancora in fase di elaborazione.

L'aggiunta, nelle modalità indicate nel capitolo 4, della componente di routing alle istanze di Bin Packing definisce delle istanze CVRP; tali istanze sono state testate sull'algoritmo esatto del professor Roberti, ottenendo i risultati riportati in sezione 4.2.

Capitolo 1

Vehicle Routing Problem: descrizione del problema

La classe di problemi relativi al servizio di un insieme di clienti tramite veicoli, inizialmente localizzati in uno o più depositi, che si muovono in una rete stradale viene identificata con il nome di Vehicle Routing Problem (*VRP*).

Il problema è stato proposto da Dantzig e Ramser nel 1959. I due studiosi descrissero un'applicazione reale riguardo la consegna di gasolio a stazioni di servizio. Il VRP è uno dei problemi di ottimizzazione combinatoria più studiati; le motivazioni del suo interesse scaturiscono dalla sua importanza pratica e dalla sua difficoltà di risoluzione.

I molteplici risvolti pratici del Vehicle Routing Problem si incontrano nei settori inerenti al trasporto merci, come ad esempio:

- vendite online;
- servizi di visite mediche a domicilio;
- servizio scuolabus;
- raccolta di posta nelle cassette postali;

- visite di manutenzione di impianti;
- raccolta rifiuti;
- trasporto in ambito urbano di persone affette da handicap;
- distribuzione di prodotti alimentari di una regione.

Le difficoltà che si possono incontrare nella risoluzione delle istanze di VRP sono di varie tipologie.

Ad esempio, la necessità di affrontare problemi reali con un elevato numero di clienti rende difficoltosa la risoluzione esatta in tempi di calcolo brevi.

Un altro elemento che complica la risoluzione è la mole di vincoli a cui devono essere soggette alcune soluzioni di problemi di VRP dipendenti da molti parametri.

Dare una classificazione generale dei problemi di VRP non è semplice; ci si limita ora ad esporre le loro principali caratteristiche .

1.1 Caratteristiche generali del VRP

Il servizio dei *clienti* è effettuato distribuendo uno o più tipologie di *merce* attraverso una flotta di *veicoli*, che parte da uno o più *depositi* ed è affidata ad *autisti*, che si muove su una *rete stradale*.

Definiti i parametri distintivi del problema, una soluzione di VRP è un insieme di percorsi effettuati dai veicoli, i quali partono ed arrivano nei depositi rispettando le richieste specifiche del problema in merito ai clienti coinvolti nel circuito e al veicolo stesso. La soluzione deve inoltre minimizzare una misura del costo totale del servizio.

Tra i problemi di VRP si possono riconoscere varie tipologie a seconda delle

differenti modalità operative della situazione applicativa presa in esame.

Si illustrano ora le diverse caratteristiche distintive di clienti, merce, veicoli, depositi e autisti di un problema VRP.

Ciascun cliente è caratterizzato da:

- la sua posizione nella rete stradale in cui si spostano i veicoli;
- la quantità (*domanda*) e il tipo di merce che deve essere consegnata oppure raccolta presso il cliente;
- il tempo di scarico o carico della merce, che può diventare molto rilevante in situazioni come il trasporto di auto da parte di una bisarca;
- eventuali restrizioni di natura logistica sull'insieme di veicoli che può servire ciascun cliente;
- un'eventuale finestra temporale entro cui ciascun cliente deve essere servito (ad esempio un orario di apertura, oppure un orario di accesso ad una Z.T.L.);
- eventuali priorità sul servizio di clienti nel caso in cui non sia possibile raggiungerli tutti;
- eventuali penalità associate alla parziale o totale mancanza di servizio.

I veicoli che servono i clienti si caratterizzano per:

- il deposito di appartenenza, che può non coincidere con il deposito di ritorno;
- la capacità di carico, in termini di peso, volume, oppure unità di imballo delle tipologie di merce trasportabile;
- eventuale suddivisione in scompartimenti, ognuno caratterizzato dalla sua capacità e dal tipo di merce trasportabile (ad esempio celle frigorifere e vani non refrigerati in uno stesso veicolo);

- insieme dei collegamenti della rete stradale attraversabili;
- idoneità al trasporto di alcune tipologie di merce;
- costo di utilizzo del veicolo, per unità di distanza percorsa e/o di tempo impiegato.

La flotta di veicoli può essere omogenea se è costituita da veicoli identici ai fini del problema, oppure eterogenea se i veicoli presentano caratteristiche diverse.

Gli autisti che guidano i veicoli sono soggetti a vincoli sindacali riguardanti, ad esempio, l'orario di lavoro ed il numero e la durata della pause tra i turni lavorativi. Di fatto tali vincoli si associano direttamente al veicolo.

La rete stradale in cui è ambientato il problema è solitamente planare e quindi può essere considerata su un grafo piano.

Un problema di VRP si pensa quindi come grafo orientato completo e pesato. I suoi vertici sono i punti notevoli della rete stradale, ovvero clienti, depositi e, se è necessaria una rappresentazione in dettaglio, gli incroci tra le strade. Gli archi sono i collegamenti stradali tra questi punti ed hanno a loro associati una direzione di percorrenza ed un costo minimo di transito quantificabile in termini di lunghezza o tempo di passaggio.

Generalmente si assume che la rete stradale sia euclidea, cioè goda della proprietà per cui il costo c_{ij} associato al tratto stradale che collega la città i alla città j è minore o uguale alla somma di costi $c_{ik} + c_{kj}$ per qualsiasi città k del grafo. La disuguaglianza triangolare è naturalmente verificata se il costo di un arco rappresenta la lunghezza del corrispondente tratto stradale.

Gli obiettivi da conseguire in un VRP possono essere molteplici e purtroppo, talvolta, contrastanti.

Si possono richiedere:

- la minimizzazione del costo globale di trasporto, dipendente dalla distanza totale percorsa, dal tempo totale impiegato e dai costi fissi per i veicoli e gli equipaggi;

- la minimizzazione del numero di veicoli e di autisti necessari per servire i clienti;
- il bilanciamento dei percorsi compiuti da ciascun veicolo in relazione al tempo di percorrenze e/o al carico di merce nei veicoli;
- la minimizzazione delle penalità accumulate in seguito al disservizio dei clienti.

Talvolta, non si conosce totalmente a priori la domanda dei clienti, quindi è necessario introdurre dei termini stocastici per poter trattare il problema.

Nel seguito ci si concentra sul Capacitated Vehicle Routing Problem (*CVRP*), che è la versione più comune dei problemi di trasporto merci.

1.2 Capacitated Vehicle Routing Problem (CVRP)

Il CVRP è un problema della classe VRP con le seguenti caratteristiche:

- presenza di un'unica tipologia di merce, che deve essere solamente consegnata (o solamente raccolta) ai clienti;
- domande dei clienti note a priori;
- le domande dei clienti devono essere soddisfatte completamente;
- flotta di veicoli omogenea di cardinalità fissata;
- presenza di un unico deposito.

L'obiettivo è la determinazione, per ogni veicolo, di un circuito di vertice iniziale e finale il deposito, in modo che:

1. ogni cliente sia servito da un solo veicolo, ovvero ciascun vertice sia visitato esattamente da un circuito;

2. la somma delle domande dei clienti serviti da un veicolo non ecceda la capacità del veicolo;
3. la somma dei costi dei circuiti in soluzione, rappresentata come la distanza percorsa, sia minimizzata.

Il CVRP è considerato un problema NP-hard: non si conoscono algoritmi polinomiali per risolverlo e, se se ne trovasse uno allora dovrebbe esserne uno per ogni problema NP-hard.

Il CVRP si presenta come il problema applicato più vicino a due problemi cardine della ricerca operativa: il Bin Packing e il Traveling Salesman Problem.

1.3 Bin Packing e Traveling Salesman Problem

Il Bin Packing Problem (BPP) consiste nella minimizzazione del numero di contenitori (bin) di uguale capacità necessari per inserire (*packing*), senza sovrapposizioni, un insieme di oggetti ciascuno dotato di una certa dimensione.

Il Bin Packing è un problema di ottimizzazione NP-completo in senso forte, ovvero è NP-hard e non esistono algoritmi pseudopolinomiali per risolverlo. Per questo motivo è altamente improbabile che esistano algoritmi veloci in grado di determinarne una soluzione. La ricerca si è quindi rivolta verso la ricerca di algoritmi di approssimazione a prestazione garantita.

Tra le applicazioni del Bin Packing si possono citare la gestione dei processori e della memoria, l'assegnazione di spot pubblicitari in pagine Web o di giornali, il taglio di lastre di vetro, di pannelli o lamine nella produzione industriale.

Si illustrano ora il modello di programmazione lineare intera che risolve il Bin Packing Problem nella sua formulazione naturale di Capacitated Facility

Location Problem e il modello di tipo Set Covering.

Detti m il numero di oggetti, n il numero di contenitori, b la capacità dei bin e d_i la domanda dell' i -esimo oggetto, si definiscono, per $i = 1, \dots, m$ e $j = 1, \dots, n$ le variabili:

$$x_{ij} = \begin{cases} 1 & \text{se l'oggetto } i \text{ è nel contenitore } j \\ 0 & \text{altrimenti} \end{cases}$$

$$y_j = \begin{cases} 1 & \text{se il contenitore } j \text{ è in soluzione} \\ 0 & \text{altrimenti} \end{cases}$$

e si introduce il modello:

$$\min \sum_{j=1}^n y_j$$

$$(1) \quad \sum_{j=1}^n x_{ij} = 1 \quad \forall i = 1, \dots, m$$

$$(2) \quad x_{ij} \leq y_j \quad \forall i = 1, \dots, m, j = 1, \dots, n$$

$$(3) \quad \sum_{i=1}^m d_i x_{ij} \leq b y_j \quad \forall j = 1, \dots, n$$

$$x \in \{0, 1\}^{m \cdot n}, y \in \{0, 1\}^n$$

I vincoli (1) esprimono il fatto che ogni oggetto dev'essere assegnato ad un contenitore; i vincoli (2) assicurano che se l'oggetto i viene assegnato al bin j , allora il bin j viene considerato in soluzione; i vincoli (3) impongono che venga rispettata la capacità dei bin in soluzione.

Questo modello presenta vari svantaggi, fra cui l'elevata simmetria, che è un aspetto penalizzante nel caso di uso in sequenza delle soluzioni che si compie negli algoritmi di branch and bound.

L'altro modello usato per il BPP riguarda la sua formulazione Set Covering. Definiti $S = \{\text{sottoinsiemi di oggetti che possono essere impaccati in un bin}\}$ e le variabili

$$x_s = \begin{cases} 1 & \text{se il sottoinsieme } s \in S \text{ è in soluzione} \\ 0 & \text{altrimenti} \end{cases}$$

il modello è:

$$\begin{aligned}
 & \min \sum_{s \in S} x_s \\
 (1) \quad & \sum_{s \in S, i \in s} x_s \geq 1 \quad \forall i = 1, \dots, m \\
 & x \in \{0, 1\}^{|S|},
 \end{aligned}$$

i cui vincoli ammettono che ciascun oggetto venga assegnato a più bin massimali.

Il CVRP possiede quindi i vincoli di capacità del BPP. I clienti sono oggetti dotati di domande e devono essere serviti dai veicoli rispettandone la capacità: i veicoli hanno il ruolo dei bin.

Pertanto, dato un problema di CVRP si può naturalmente costruire un problema BPP ignorando le posizioni dei clienti nella rete stradale e considerando il numero di veicoli utilizzati come funzione obiettivo da minimizzare anziché come parametro dell'istanza.

Si nota che un problema CVRP ammette una soluzione che utilizza k veicoli se e solo se il problema di Bin Packing associato ha k come soluzione ammissibile.

L'altra componente di vincoli del CVRP riguarda il Traveling Salesman Problem (TSP).

Esso consiste nella determinazione di un circuito chiuso di lunghezza minima che visiti un insieme di città, ciascuna una e una sola volta. Il problema si può generalizzare ad una versione pesata in cui la funzione obiettivo da minimizzare è il costo del circuito costruito, che si calcola come somma dei costi di ciascun collegamento. Nella versione asimmetrica del problema (ATSP) il costo che si compie per andare dalla città i alla città j è in generale differente dal costo del collegamento $j - i$. Il Traveling Salesman Problem si pensa su un grafo pesato avente per vertici le città da visitare e per lati o archi i collegamenti tra città. In base alla presenza di simmetria nei costi

dei collegamenti, il grafo è orientato o non orientato.

A dispetto della semplicità di formulazione, il TSP è uno dei problemi di matematica computazionale più studiati e non se ne conosce un effettivo metodo risolutivo per il caso generale. Gli studi degli ultimi cinquant'anni hanno portato allo sviluppo di vari metodi di soluzione esplorando diversi campi dell'ottimizzazione matematica.

Il TSP ha molteplici applicazioni pratiche nei problemi di trasporti o logistici: ad esempio nelle ottimizzazioni per servizi via cavo, lavori urbani, problemi agricoli, tragitti di autobus scolastici e rotte che congiungono aeroporti.

Si presenta ora il modello di programmazione lineare intera che si usa per la risoluzione di ATSP.

Detti c_{ij} il costo del collegamento della città i con la città j e definite le variabili

$$x_{ij} = \begin{cases} 1 & \text{se l'arco } (i,j) \text{ è in soluzione} \\ 0 & \text{altrimenti} \end{cases}$$

il modello è:

$$\min \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij}$$

$$(1) \quad \sum_{j \in V} x_{ij} = 1 \quad \forall i \in V$$

$$(2) \quad \sum_{i \in V} x_{ij} = 1 \quad \forall j \in V$$

$$(3) \quad \sum_{a \in A(S)} x_a \leq |S| - 1 \quad |S| \subset V, 2 \leq |S| \leq |V| - 2$$

$$x \in \{0, 1\}^{|V| \times |V|},$$

dove con $A(S)$ si intende l'insieme degli archi con entrambi gli estremi in S . I vincoli di grado (1) e (2) assicurano che tutte le città vengano visitate una sola volta.

I vincoli (3) impediscono la formazione di sottocircuiti che non passano per

il deposito.

Il CVRP contiene quindi i vincoli di instradamento propri del TSP. Un problema di TSP si può vedere come caso particolare di un istanza CVRP; infatti, quando si ha a disposizione un solo veicolo e la somma delle domande dei clienti è minore o uguale alla capacità del veicolo, allora il circuito soluzione del CVRP risolve anche il TSP associato.

Capitolo 2

CVRP: Risoluzione con modelli ILP e algoritmi esatti

Si illustrano ora alcuni metodi per risolvere le istanze di CVRP.

Una prima possibilità è quella di impostare e risolvere un problema di programmazione lineare intera. Questo approccio può non essere affrontabile per problemi di grandi dimensioni o per i quali sia necessario avere una soluzione in tempo di calcolo brevi. Tali limiti pratici non possono essere ignorati nella questione della risoluzione; per questo motivo è scaturito l'interesse nello sviluppo di algoritmi esatti.

Gli algoritmi hanno l'obiettivo di fornire una soluzione di buona qualità, ovvero abbastanza vicina alla soluzione esatta, tenendo conto della limitatezza dei tempi di calcolo e della memoria impiegata. Non si può pensare che un algoritmo di questa tipologia possa essere impiegato efficacemente su una qualsiasi istanza di CVRP.

Il presente lavoro di tesi mira proprio ad approfondire alcuni aspetti per i quali i migliori algoritmi esatti per CVRP in circolazione si rivelano scarsamente efficienti per alcuni tipi di istanze.

2.1 Modelli matematici per CVRP

Si illustrano ora tre modelli di programmazione lineare intera per CVRP. Essi vengono effettivamente usati nelle applicazioni a seconda della situazione pratica considerata.

E' necessario introdurre alcune notazioni utili per descrivere i modelli.

Detto n il numero di clienti da servire, si indicherà l'insieme dei vertici del grafo associato con $V = \{0, \dots, n\}$; il vertice 0 indica il deposito, mentre $\{1, \dots, n\}$ è l'insieme dei clienti. Ciascun cliente ha domanda d_j ; un collegamento $a \in A$ della rete stradale ha costo c_a . Dato $S \subseteq V$, si indicano con $\delta^-(S)$ l'insieme degli archi con coda in S e testa in $V \setminus S$ e con $\delta^+(S)$ l'insieme degli archi con testa in S e coda in $V \setminus S$.

Si indica con k il numero di veicoli di capacità b che si vogliono utilizzare per risolvere il problema. Questo parametro deve essere maggiore della soluzione del BPP associato, che si indicherà con $\sigma(V \setminus \{0\})$. Una stima in difetto di questo valore è il lower bound geometrico del Bin Packing $\frac{\sum_{i=1}^n d_i}{b}$. Esso si ottiene nel rilassamento continuo del problema immaginando di poter servire i clienti in modo parziale e con più veicoli: in questo modo, a parte l'ultimo bin, la capacità dei bin in soluzione viene sfruttata completamente.

2.1.1 Modello 1

Si definiscono, per $a \in A$ le variabili

$$x_a = \begin{cases} 1 & \text{se l'arco } a \text{ è in soluzione} \\ 0 & \text{altrimenti} \end{cases}$$

e si considera il modello:

$$\begin{aligned}
& \min \sum_{a \in A} c_a x_a \\
(1) \quad & \sum_{a \in \delta^-(i)} x_a = 1 \quad \forall i = 1, \dots, n \\
(2) \quad & \sum_{a \in \delta^+(i)} x_a = 1 \quad \forall i = 1, \dots, n \\
(3) \quad & \sum_{a \in \delta^-(0)} x_a = k \\
(4) \quad & \sum_{a \in \delta^+(0)} x_a = k \\
(C1) \quad & \sum_{a \in \delta^+(S)} x_a \geq \sigma(S) \quad \forall S \subseteq V \setminus \{0\} \\
& x \in \{0, 1\}^{|A|}
\end{aligned}$$

I vincoli (1) – (4) curano la parte di routing del problema: i primi due assicurano che ogni cliente venga servito da un veicolo una e una sola volta, mentre i vincoli (3) e (4) assicurano che i k circuiti in soluzione partano e arrivino al deposito 0.

I vincoli (C1) curano la componente di packing del problema e impediscono la creazione di sottocircuiti che non passino per il vertice 0. In tali vincoli si precisa che $\sigma(S)$ indica il valore della soluzione ottima del problema di Bin Packing con $|S|$ oggetti con dimensioni date dalle domande d_j e capacità dei bin data da b . La cardinalità esponenziale dei vincoli (C1) rende necessario un modello di programmazione lineare intera per la loro separazione. Si dimostra però che un tale modello non esiste, quindi i vincoli vengono sostituiti dai vincoli (C2):

$$(C2) \quad \sum_{a \in \delta^+(S)} x_a \geq \frac{\sum_{i=1}^n d_i}{b} \quad \forall S \subseteq V \setminus \{0\}$$

che risultano ugualmente validi per il problema e ammettono il seguente modello ILP per la loro separazione.

Indicata con x^* la soluzione del modello ridotto, si cerca $\bar{S} \subseteq V \setminus \{0\}$ tale che $\sum_{a \in \delta^+(S)} x_a < \frac{\sum_{i=1}^n d_i}{b}$.

Pertanto si definiscono le variabili, per $i \in V$ e $a \in A$:

$$y_i = \begin{cases} 1 & \text{se } i \in \bar{S} \\ 0 & \text{altrimenti} \end{cases}$$

$$z_a = \begin{cases} 1 & \text{se } a \in \delta^+(S) \\ 0 & \text{altrimenti} \end{cases}$$

e si considera il modello:

$$\min b \sum_{a \in A} x_a^* z_a - \sum_{i \in V} d_i y_i$$

$$y_0 = 0$$

$$z_{ij} \leq y_i \quad \forall (i, j) \in A$$

$$z_{ij} \leq 1 - y_j \quad \forall (i, j) \in A$$

$$z_{ij} \geq y_i - y_j \quad \forall (i, j) \in A$$

$$y \in \{0, 1\}^n, z \in \{0, 1\}^{|A|}$$

Se il valore della soluzione è negativo, allora la soluzione corrisponde ad un vincolo violato.

Tipicamente, si imposta il modello con i vincoli (C2) e si inizia la generazione di righe. Quando si è abbastanza vicini alla soluzione si usa il modello con i vincoli (C1) perchè essi hanno un rilassamento continuo più forte rispetto ai (C2).

2.1.2 Modello 2

Il seguente modello è preferibile al precedente nel caso in cui i veicoli siano eterogenei e che ogni cliente sia dotato di una specifica finestra temporale

all'interno della quale deve essere visitato.

Si definiscono, per $a \in A$, $h \in \{1, \dots, k\}$, $i \in \{1, \dots, n\}$ le variabili:

$$x_a^h = \begin{cases} 1 & \text{se l'arco } a \text{ appartiene al circuito } h \\ 0 & \text{altrimenti} \end{cases}$$

$$y_i^h = \begin{cases} 1 & \text{se il vertice } i \text{ è visitato dal circuito } h \\ 0 & \text{altrimenti} \end{cases}$$

Il modello è:

$$\min \sum_{h=1}^k \sum_{a \in A} c_a x_a^h$$

$$(1) \quad \sum_{a \in \delta^-(i)} x_a^h = y_i^h \quad \forall i = 1, \dots, n, \forall h = 1, \dots, k$$

$$(2) \quad \sum_{a \in \delta^+(i)} x_a^h = y_i^h \quad \forall i = 1, \dots, n, \forall h = 1, \dots, k$$

$$(3) \quad \sum_{h=1}^k y_i^h = 1 \quad \forall i = 1, \dots, n$$

$$(4) \quad \sum_{a \in \delta^-(0)} x_a^h = 1 \quad \forall h = 1, \dots, k$$

$$(5) \quad \sum_{a \in \delta^+(0)} x_a^h = 1 \quad \forall h = 1, \dots, k$$

$$(6) \quad \sum_{i=1}^n d_i y_i^h \leq b \quad \forall h = 1, \dots, k$$

$$(7) \quad \sum_{a \in \delta^+(S)} x_a^h \geq y_i^h \quad \forall S \subseteq V \setminus \{0\}, i \in S, \forall h = 1, \dots, k$$

$$x \in \{0, 1\}^{|A|} \times \{0, 1\}^k, y \in \{0, 1\}^n$$

I vincoli (1) e (2) esprimono il fatto che nei percorsi compiuti da ciascun veicolo i clienti vengono visitati una e una sola volta. Il vincolo (3) afferma che ciascun vertice è visitato da un sottocircuito. I vincoli (4) e (5) assicurano che ciascun veicolo parta ed arrivi al deposito. I vincoli di tipo (6) assicurano

il rispetto della capacità dei veicoli e possono essere adattati in modo immediato al caso di veicoli eterogenei per capacità. I vincoli (7) impediscono la creazione di sottocircuiti che non passino per 0; essi sono di cardinalità esponenziale.

Data una soluzione \bar{x} e \bar{y} e fissati $h \in \{1, \dots, k\}$ e $i \in V \setminus \{0\}$, il problema di separazione per i vincoli (7) richiede di determinare, se esiste, un sottoinsieme $\bar{S} \subseteq V \setminus \{0\}$ tale che $\sum_{a \in \delta^+(\bar{S})} \bar{x}_a^h < \bar{y}_i^h$.

Si risolve quindi l'ILP con variabili, per $j \in \{1, \dots, n\}$ e $a \in A$:

$$z_a = \begin{cases} 1 & \text{se } a \in \delta^+(\bar{S}) \\ 0 & \text{altrimenti} \end{cases}$$

$$w_j = \begin{cases} 1 & \text{se il vertice } j \text{ appartiene ad } \bar{S} \\ 0 & \text{altrimenti} \end{cases}$$

e modello:

$$\min \sum_{a \in A} \bar{x}_a^h z_a$$

$$w_0 = 0$$

$$w_i = 0$$

$$z_{lj} \leq w_l \quad \forall (l, j) \in A$$

$$z_{lj} \leq 1 - w_j \quad \forall (l, j) \in A$$

$$z_{lj} \geq w_l - w_j \quad \forall (l, j) \in A$$

$$w \in \{0, 1\}^n, z \in \{0, 1\}^{|A|}$$

Se la soluzione non è minore di \bar{y}_i^h , allora si fissa un altro vertice i e si imposta un nuovo modello; se si fallisce con tutti i vertici i , allora si cambia il veicolo h . Nel caso peggiore si risolvono kn modelli ILP.

2.1.3 Modello 3 a generazione di colonne

L'ultimo modello illustrato per il Vehicle Routing è un modello a generazione di colonne: prevalentemente, viene utilizzato quando ci sono vincoli di natura complessa da imporre sui circuiti.

Si definiscono:

- C l'insieme dei circuiti ammissibili per un veicolo;
- γ_C costo del circuito $C \in C$, pari alla somma dei costi degli archi nel circuito;
- per $i \in V \setminus \{0\}$, C_i l'insieme dei circuiti ammissibili che visitano i .

Si introducono, per $C \in C$ le variabili:

$$x_C = \begin{cases} 1 & \text{se il circuito } C \text{ è in soluzione} \\ 0 & \text{altrimenti} \end{cases}$$

e si considera il modello:

$$\begin{aligned} & \min \sum_{C \in C} \gamma_C x_C \\ (a) \quad & \sum_{C \in C} x_C = k \\ (b) \quad & \sum_{C \in C_i} x_C = 1 \quad \forall i \in \{1, \dots, n\} \\ & x_C \in \{0, 1\}^{|C|} \end{aligned}$$

Il vincolo (a) assicura che vengano costruiti k circuiti, mentre i vincoli (b) affermano che ogni cliente viene visitato esattamente da un veicolo. Visto che vale la disuguaglianza triangolare, visitare un cliente più di una volta non è conveniente ai fini della minimizzazione della funzione obiettivo; per questo motivo, i vincoli (b) possono essere rilassati sostituendo all'uguaglianza la disuguaglianza $\sum_{C \in C_i} x_C \geq 1$.

Il modello presenta un numero esponenziale di variabili, per cui si deve attuare la generazione di colonne. Detti I_C l'insieme dei vertici visitati da un

circuito $C \in \mathcal{C}$, z la variabile associata al vincolo (a) e y_1, \dots, y_n le variabili associate ai vincoli (b), il problema duale è:

$$\begin{aligned} \max \quad & kz + \sum_{i \in V \setminus \{0\}} y_i \\ z + \sum_{i \in I_C} y_i &\leq \gamma_C \quad \forall C \in \mathcal{C} \end{aligned}$$

Il problema di separazione consiste, dati \bar{z} e \bar{y} di determinare, se esiste, un circuito $\bar{C} \in \mathcal{C}$ tale che $\bar{z} + \sum_{i \in I_{\bar{C}}} \bar{y}_i > \gamma_{\bar{C}}$.

A tale scopo si definisce un modello simile al secondo visto per la risoluzione di CVRP, ma adattato alla ricerca di un unico sottocircuito.

Si definiscono, per $a \in A$, $h \in \{1, \dots, k\}$, $i \in \{1, \dots, n\}$ le variabili:

$$x_a = \begin{cases} 1 & \text{se l'arco } a \text{ è in soluzione} \\ 0 & \text{altrimenti} \end{cases}$$

$$w_i = \begin{cases} 1 & \text{se il vertice } i \text{ è in soluzione} \\ 0 & \text{altrimenti} \end{cases}$$

$$\begin{aligned} \max \quad & \sum_{i \in V \setminus \{0\}} \bar{y}_i w_i - \sum_{a \in A} c_a x_a \\ (1) \quad & \sum_{a \in \delta^-(i)} x_a = w_i \quad \forall i = 1, \dots, n \\ (2) \quad & \sum_{a \in \delta^+(i)} x_a = w_i \quad \forall i = 1, \dots, n \\ (3) \quad & \sum_{a \in \delta^-(0)} x_a = 1 \\ (4) \quad & \sum_{a \in \delta^+(0)} x_a = 1 \\ (4) \quad & \sum_{i=1}^n d_i w_i \leq b \\ (6) \quad & \sum_{a \in \delta^+(S)} x_a \geq w_i \quad \forall S \subseteq V \setminus \{0\}, i \in S \end{aligned}$$

$$x \in \{0, 1\}^{|A|}, w \in \{0, 1\}^n$$

2.2 Motivazioni del lavoro di tesi

Il presente lavoro di tesi nasce dall'intento di comprendere la complessità pratica delle istanze di CVRP in relazione alla componente di packing in modo da far luce sui punti deboli degli algoritmi esatti ed euristici che costruiscono soluzioni del CVRP basandosi sulla formulazione Set Partitioning del problema. Il modello di Set Partitioning è quello visto nella sezione 2.1.3 con i vincoli di uguaglianza.

In linea generale, gli algoritmi di questa categoria costruiscono alcuni tipi di circuiti ammissibili per il problema e ne testano l'ottimalità risolvendo il problema di pricing associato utilizzando anche bound validi.

A titolo di esempio, si può prendere l'algoritmo di Baldacci, Mingozzi e Roberti descritto nell'articolo del 2010 *New route relaxation and pricing strategies for the vehicle routing problem*: esso costituisce uno dei migliori algoritmi esatti per la risoluzione dei problemi di CVRP e VRP con finestre temporali. L'algoritmo si caratterizza per alcune procedure di bounding nel problema duale che si realizzano grazie all'uso di route elementari e route non elementari, denominate q -path e q -routes, e per la costruzione di particolari circuiti chiamati ng -route.

Negli algoritmi che usano l'approccio Set Partitioning si osserva, dunque, un aspetto comune che riguarda la costruzione di circuiti. L'attenzione alla parte di routing del problema VRP potrebbe portare a pensare che gli algoritmi incontrino delle difficoltà su istanze in cui la componente di packing è "difficile". In altre parole, sulla base delle considerazioni esposte, si è portati ad ipotizzare che, se un'istanza di VRP è dotata di poche soluzioni di valore ottimo per il Bin Packing associato, allora l'algoritmo potrebbe faticare nel trovare una soluzione ottima di VRP.

Il lavoro di tesi è quindi consistito nella generazione di istanze dotate di poche soluzioni di valore ottimo, a cui è stata aggiunta la componente di routing: le istanze di CVRP così composte sono state poi testate sull'algoritmo di Baldacci, Mingozzi e Roberti grazie alla collaborazione del dr. Roberti.

Capitolo 3

Generazione di istanze BPP difficili

Nel presente capitolo, si illustra il lavoro svolto per creare istanze di Bin Packing dotate di un numero limitato di soluzioni di valore ottimo. Il conteggio delle soluzioni non deve tenere conto della numerazione sui bin: i veicoli che servono i clienti nel problema CVRP sono indistinguibili tra loro, quindi non ha senso considerare diverse due soluzioni ottime che differiscono per lo scambio di due o più bin.

Allo scopo di generare questo tipo di istanze, ci si è concentrati sulle disposizioni di perfect packing, ovvero quando tutti i bin in soluzione vengono riempiti completamente dalle domande dei clienti. Il fatto che nei bin in soluzione non ci siano residui liberi di capacità potrebbe suggerire, infatti, una certa difficoltà in termini di packing.

3.1 Generazione di istanze di perfect packing con un numero di soluzioni ottime limitato

La costruzione di istanze di perfect packing dotate di pochissime soluzioni di valore ottimo non equivalenti è stata compiuta seguendo un suggerimento del professor Cor Hurkens.

Hurkens ipotizzava che la probabilità che in una sequenza casuale di 100 numeri interi distribuiti uniformemente tra 1 e $10^{10} - 1$ la somma di un gruppo di 10 numeri consecutivi sia uguale a $5 \cdot 10^{10}$ fosse molto bassa. Si pensa inoltre che sia poco probabile che si riesca a partizionare il gruppo dei 100 numeri in sottoinsiemi da 8, 9, 11 o 12 elementi in modo che la somma dei numeri in ciascun sottoinsieme sia pari a $5 \cdot 10^{10}$.

Sulla base di queste considerazioni si può generare un'istanza di packing nel modo seguente:

- si scelgono 90 numeri interi distribuiti in maniera uniforme tra 1 e $10^{10} - 1$;
- si dividono i 90 numeri, se possibile, in 10 gruppi da 9 in modo che la somma dei numeri in ciascun gruppo sia minore di $5 \cdot 10^{10}$;
- si inserisce un numero in ciascun gruppo in modo che la somma dei numeri in ciascun gruppo diventi esattamente $5 \cdot 10^{10}$

L'istanza generata è di perfect packing e sembra altamente improbabile che ammetta altre soluzioni ottime.

Un'indicazione della veridicità di questa congettura si può ottenere dal teorema del limite centrale: data una somma di variabili aleatorie X_i indipendenti e identicamente distribuite con media μ e varianza σ^2 si ha che $\lim_{n \rightarrow \infty} \sum_{i=1}^n X_i$ si approssima con $N(n \cdot \mu, n \cdot \sigma^2)$ dove N è la distribuzione normale. Nel caso della distribuzione discreta uniforme nell'intervallo $[a, b]$, la media è $\mu = \frac{a+b}{2}$ e la varianza è $\sigma^2 = \frac{(a-b)^2 \cdot (n+1)}{12 \cdot (n-1)}$. Si riporta nella figura 3.1 la funzione di densità di $N(n \cdot \mu, n \cdot \sigma^2)$ con $n = 10$: $f(x) = \frac{1}{\sqrt{2\pi n \sigma^2}} e^{-\frac{(x-n\mu)^2}{2n\sigma^2}}$ per $x \in \mathbb{R}$.

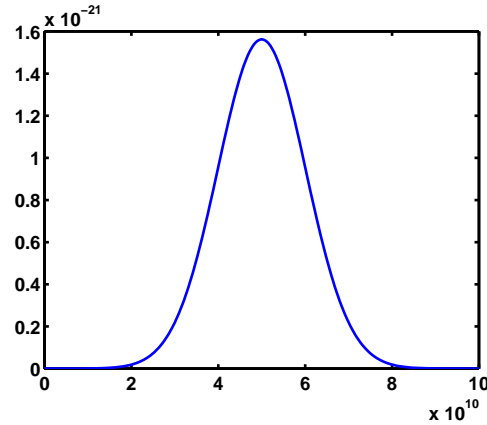


Figura 3.1: Funzione di densità relativa all'esempio di Hurkens

Dunque la probabilità che $\sum_{i=1}^n X_i$ assuma valori vicini a $D = 5 \cdot 10^{10}$ si stima calcolando l'integrale della funzione di densità in un intervallo relativamente piccolo centrato in D , ad esempio $[D - 10, D + 10]$: il risultato è $3.1252 \cdot 10^{-20}$.

L'istanza è stata poi risolta con il software *IBM CPLEX* integrato in *Matlab*. E' stato caricato il modello naturale di Bin Packing con l'aggiunta dei vincoli di assegnazione del primo oggetto al primo bin e di attivazione di un bin solo se il precedente bin è stato attivato. Il *solve* di *Cplex* trova la soluzione di perfect packing costruita.

L'interesse principale del lavoro era quello di creare istanze difficili per il packing con capacità dei bin di valori tra 100 a 200, affinché potessero essere confrontabili con le istanze proposte in letteratura e per testarle con algoritmi esatti. Tale obiettivo potrebbe essere raggiunto se si riuscissero a generalizzare le stime di probabilità compiute nell'esempio di Hurkens per capacità e intervalli di definizione delle domande degli oggetti qualsiasi.

Considerare capacità molto più contenute limita però il campo di variabilità delle domande degli oggetti e conseguentemente si aumenta il rischio di costruire istanze per cui la soluzione di perfect packing non sia l'unica soluzione ottima: infatti, appena si creano due o più domande in un bin la cui somma

è pari a quella di un altro gruppo di oggetti in un altro bin, allora si crea una nuova soluzione ottima invertendo i gruppi di oggetti con stessa somma di domande. Per questo motivo, il numero totale di oggetti difficilmente potrà essere alto (superiore a 100). Tale restrizione rende inaffidabile l'approssimazione fornita dal teorema del limite centrale.

La costruzione delle istanze difficili di perfect packing con capacità contenuta è stata condotta nel seguente modo:

- scelta della capacità b (valori da 140 a 200);
- scelta del numero k di bin nella soluzione di perfect packing, del numero n di oggetti in un bin (costante indipendentemente dal bin) e del parametro D che definisce la variabilità delle domande in modo da incrementare la difficoltà in termini di packing dell'istanza;
- generazione delle domande di $k \cdot (n - 1)$ oggetti distribuite uniformemente su $[1, D]$;
- suddivisione (se consentita) dei $k \cdot (n - 1)$ oggetti di cui è stata definita la domanda in k gruppi da $n - 1$ oggetti in modo che la somma delle domande delle capacità non ecceda b ;
- inserimento degli ultimi k oggetti in modo che l'istanza sia costituita da k gruppi di n oggetti con somma di domande pari a b ;

Si noti che, dopo aver scelto la capacità, ci sono vari punti di indeterminazione nella scelta di alcuni parametri che vanno trattati tenendo presente diverse considerazioni:

- conviene tenere n basso (3, 4, 5) e k più alto allo scopo di riuscire a raggruppare gli oggetti senza eccedere la capacità;
- il parametro D va invece scelto tenendo conto di due esigenze: più ristretto è l'intervallo da cui vengono generate le domande e più è facile compiere i k raggruppamenti; allargando invece l'intervallo di

variabilità si aumentano le probabilità di generare un'istanza con poche soluzioni ottime.

La questione del conteggio delle soluzioni ottime verrà trattata nella sezione 3.3, ma si possono comunque avere delle indicazioni sulla difficoltà in termini di packing. E' importante rilevare che, per tali istanze, il *solve* di *Cplex* è estremamente lento nell'esplorazione dell'albero di branch and bound e che la soluzione di perfect packing viene trovata dopo molto tempo o non viene trovata affatto. Si allegano nella seguente tabella i parametri scelti per le istanze generate.

| b | D | n | k |
|-----|-----|-----|-----|
| 200 | 167 | 3 | 15 |
| 200 | 100 | 3 | 15 |
| 180 | 82 | 5 | 25 |
| 160 | 80 | 4 | 30 |
| 150 | 70 | 3 | 20 |
| 140 | 125 | 3 | 15 |

Tabella 3.1: Parametri libreria 1 istanze

Sulla base dell'esempio di Hurkens, si è generata poi un'altra istanza di perfect packing che si suppone essere difficile dal punto di vista del packing. Si riportano le operazioni applicate alle domande degli oggetti dell'istanza di Hurkens:

- divisione di domande e capacità per 10^7 ;
- correzione della domanda dell'ultimo oggetto del bin se la somma delle domande degli oggetti nel bin non è 5000;
- agli oggetti con domanda minore di 100 la domanda viene aumentata di 100 e conseguentemente si corregge la domanda dell'ultimo oggetto del relativo bin;

- contate le occorrenze di ciascuna domanda nell'insieme delle domande degli oggetti, si eliminano i doppioni modificando le domande di un'unità e di conseguenza si cambia la domanda dell'ultimo oggetto del bin.

Le operazioni compiute hanno lo scopo di riportare l'istanza di Hurkens a numeri più piccoli cercando di mantenere alta la variabilità delle domande, che è strettamente legata alla difficoltà in termini di packing.

Le 7 istanze così create vengono considerate nella Libreria 1.

Un'altra modalità per generare istanze difficili di perfect packing, che andranno in Libreria 2, si ottiene studiando il seguente lemma di divisibilità.

Lemma 1.

Sia $c \in \mathbb{N}$. Siano a, b divisori di c , $(a, b) = 1$.

Allora tutti e soli i multipli di c sono i multipli comuni tra $\frac{c}{a}$ e $\frac{c}{b}$.

Dimostrazione 1.

Per assurdo, esistono $k < a, l < b$ tali che $k\frac{c}{a} = l\frac{c}{b}$.

Quindi $bkc = alc$, da cui $bk = al$.

Allora $b | al$ e $a | bk$. Poichè $(a, b) = 1$ si ha $b | l$ e $a | k$.

Dunque esistono k', h' tali che: $bak' = abl'$ ed allora $k' = l' =: h$.

In conclusione: $k = ha < a, l = hb < b$, che è assurdo.

La procedura seguente consente di generare istanze di perfect packing con un'unica soluzione ottima.

- Si sceglie la capacità b dei bin.
- Si scelgono alcuni divisori di b primi tra loro.

- Detto div uno di tali divisori, si creano div copie di domande $\frac{b}{div}$ in modo che riempiano completamente un bin.

L'istanza così creata è costituita da gruppi di oggetti che si impaccano perfettamente in un bin se hanno la stessa domanda. Per il lemma 1, non ci sono altre disposizioni degli oggetti che permettano di usare un numero di bin pari a quello della soluzione ottima. Se si considerano capacità vicine a quelle della Tabella 3.1, le istanze generate in questo modo hanno un numero basso di bin (2 o 3) e la cardinalità dell'insieme di oggetti per bin varia da bin a bin.

In tabella 3.2 si presentano le capacità e i relativi divisori scelti.

| b | Divisori |
|-----|----------|
| 220 | 4, 5, 11 |
| 204 | 3, 4, 17 |
| 198 | 2, 3, 11 |
| 182 | 2, 7, 13 |
| 156 | 3, 4, 13 |

Tabella 3.2: Parametri libreria 2 istanze

3.2 Risoluzione istanze con generazione di colonne

Costruite le istanze della libreria 1, nasce l'esigenza di avere indicazioni sulla loro difficoltà in termini di packing.

Visto che nella precedente sezione si è evidenziata la difficoltà compiuta dal *solve* di CPLEX nel cercare la soluzione di perfect packing delle istanze proposte, si è pensato di usare la formulazione Set Covering del problema per risolvere il BPP come avviene nelle applicazioni reali; il modello di Set Covering è quello visto nella sezione 2.1.3.

Il modello è stato eseguito solo per le istanze della libreria 1, perchè non si è sicuri dell'unicità della soluzione ottima del BPP associato a tali istanze. L'algoritmo usato è un algoritmo esatto che procede nel modo seguente: inizializza alcune colonne in maniera euristica; cerca soluzioni di valori via via più piccoli finchè trova soluzioni ammissibili; effettua il Branch & Price per esibire una soluzione del valore calcolato.

Le soluzioni delle istanze della libreria 1 secondo il metodo a generazione di colonne sono quelle di perfect packing note e il tempo di esecuzione è inferiore ad 1 secondo.

La velocità di esecuzione può suggerire, quindi, che per un algoritmo esatto sia abbastanza semplice generare le colonne relative alla soluzione di perfect packing.

Ci si è chiesti, poi, se, dato come insieme di colonne del primale ristretto le colonne relative alla soluzione di perfect packing nota, fosse possibile trovare nel duale ristretto dei vincoli violati o se, invece, la particolare configurazione della soluzione impedisse di aggiungere nuove colonne. Questa ipotesi, in generale, non è vera, come si può vedere nell'esempio seguente.

Esempio 3.1.

Si hanno 4 oggetti con domande,rispettivamente, 3,6,5,4 e capacità dei bin pari a 9.

L'insieme S dei bin massimali è quindi: $\{s1 = \{3,6\}, s2 = \{3,5\}, s3=\{3,4\}, s4=\{5,4\}\}$.

L'unica soluzione di bin packing, che è anche di perfect packing, è costituita dai bin $s1$ e $s4$.

Se le colonne del primale sono quelle che riguardano questa soluzione, allora il problema duale è:

$$\max y_1 + y_2 + y_3 + y_4$$

$$y_1 + y_2 \leq 1$$

$$y_3 + y_4 \leq 1$$

$$y \in \{0,1\}^4$$

La soluzione di questo problema potrebbe essere $y_1 = y_3 = 1$, ma allora il bin s_2 sarebbe da aggiungere come colonna nel primale.

In conclusione, si può quindi ritenere che le istanze siano "facili" per il column generation.

3.3 Conteggio soluzioni non isomorfe di BPP tramite orbital branching

La lentezza dell'esplorazione dell'albero di branch & bound da parte del *solve* di CPLEX nella formulazione naturale del Bin Packing è dovuta anche all'elevata simmetria del modello visto in sezione 1.3.

Si è provato ad usare la procedura *populate* di CPLEX, che è stata appunto creata per contare le soluzioni ottime o vicine all'ottimo di un problema di Programmazione Intera Mista. In tale procedura non sono previste delle tecniche che permettano di riconoscere le soluzioni simmetriche di un problema, e quindi anche quelle del Bin Packing in formulazione naturale: l'algoritmo di *populate* tenta quindi di contare tutte le soluzioni ottime del problema. Un'enumerazione di questo tipo, però, non è effettivamente gestibile dal calcolatore, perchè la quantità di soluzioni ottime cresce in maniera più che esponenziale con l'aumentare del numero di bin della soluzione ottima: se n è il numero di bin a disposizione nel modello, per ogni soluzione che sfrutta k bin, ci sono $\binom{n}{k}k!$ soluzioni equivalenti.

In letteratura, sono state sviluppate delle tecniche, denominate di *orbital branching*, di esplorazione dell'albero delle soluzioni, che si applicano a modelli che presentano un'elevata simmetria. Tali tecniche si basano sul calcolo dei *gruppi di simmetria* del problema.

Detti n la dimensione del problema, Π^n il gruppo delle permutazioni su $\{1, \dots, n\}$ e F l'insieme delle $x \in \{0, 1\}^n$ soluzioni ammissibili, si definisce il gruppo di simmetria $G = \{\pi \in \Pi^n \mid \pi(x) \in F \forall x \in F\}$. Si può poi considerare il gruppo delle orbite $orb(G, z) = \{\pi(z) \mid \pi \in G\}$. Nel caso del Bin

Packing, il gruppo di simmetria conterrà le permutazioni sugli oggetti che spostano il contenuto di uno o più bin tra loro.

Il calcolo del gruppo di simmetria di un problema di programmazione lineare intera è NP-hard, per cui, nelle applicazioni pratiche, si deve mirare ad esplorarne solamente un suo sottogruppo, scelto in modo opportuno in base alla formulazione del problema. Determinate tali simmetrie, si imposta un algoritmo di branching tenendo presente che, dato un vincolo $\alpha^T x \leq \beta$ con $(\alpha, \beta) \in \{0, 1\}^{n+1}$, allora si verifica una delle due eventualità: una forma simmetricamente equivalente di $\alpha^T x \leq \beta$ vale per uno dei vettori di $orb(G, \alpha)$, oppure vale $\alpha^T x \geq \beta + 1$ per tutti tali vettori.

La costruzione di un albero delle soluzioni di questo tipo consente di raggruppare le soluzioni in termini di simmetria rispetto alla formulazione del problema. Questa proprietà viene spesso usata per esplorare in modo oculato l'albero e risolvere velocemente il Bin Packing; spiegazioni dettagliate su questa tipologia di algoritmi si possono trovare nell'articolo in referenza 7.

In letteratura sono stati messi a punto efficacemente degli algoritmi in cui la proprietà di simmetria viene sfruttata esclusivamente per trovare una soluzione ottima del problema. Attualmente sono allo stato di elaborazione alcune tecniche che impieghino l'orbital branching per contare tutte le soluzioni non isomorfe di un problema di programmazione lineare intera. Funzionalità di questo tipo avrebbero un interesse pratico notevole, come evidenziato nel presente lavoro di tesi, e per questo andrebbero integrate negli algoritmi di enumerazione delle soluzioni come il *populate* di CPLEX.

Capitolo 4

Generazione di istanze CVRP da istanze BPP e risoluzione

Dopo aver creato le due librerie di istanze di perfect packing come descritto nel capitolo 3, è necessario dotare gli oggetti di coordinate spaziali, in modo da aggiungere la componente di routing e produrre così istanze CVRP. Le reti stradali in cui vengono inserite le istanze sono euclidee, in quanto il costo per muoversi tra due punti della rete è pari alla loro distanza planare.

4.1 Generazione di istanze CVRP da istanze BPP

Data un'istanza di Bin Packing sono state generate tre istanze di CVRP diverse tra loro per la parte di routing.

L'obiettivo era quello di verificare nelle istanze CVRP il legame tra difficoltà in termini di packing e difficoltà di risoluzione VRP con gli algoritmi esatti relativi all'approccio Set Partitioning. Per questo, ad ogni istanza BPP delle due librerie sono stati aggiunti tre insiemi di coordinate spaziali differenti tra loro in termini di difficoltà per la componente di routing. Si è pensato che un algoritmo che privilegia l'aspetto di routing tenda a costruire route costituite da clienti vicini tra loro, se la capacità dei veicoli lo consente.

Ritenere che l'istanza abbia poche soluzioni ottime non isomorfe di packing e conoscere una di tali soluzioni di packing rende quindi possibile costruire la parte di routing ritenuta, in questo senso, difficile disponendo i clienti di uno stesso bin lontani tra loro. Viceversa, per facilitare la risoluzione delle istanze CVRP, si posizionano i clienti di uno stesso bin in soluzione vicini tra loro. Un grado intermedio di difficoltà si ottiene invece da una disposizione casuale dei clienti nella rete stradale.

Descrizione dell'aggiunta della componente routing alle istanze di packing

Gli oggetti delle istanze di routing sono stati distribuiti nel quadrato Q di vertici $(0, 0)$, $(50, 0)$, $(50, 50)$ e $(0, 50)$; il deposito da cui partono i veicoli è situato nel punto $(25, 25)$.

Si analizza ora il procedimento seguito per aggiungere le coordinate spaziali agli oggetti per ottenere la configurazione difficile per il routing.

La procedura cambia leggermente tra istanze della libreria 1 e quelle della libreria 2, perchè ci sono delle modifiche da apportare nel caso in cui il numero di oggetti in ogni bin non sia lo stesso in ogni bin. Si considera dapprima il caso delle istanze della libreria 1, in cui il numero di oggetti per bin è costante.

- Partizione di Q in una griglia con un numero di rettangoli pari al numero di oggetti in ciascun bin della soluzione di perfect packing.
- Partizione di ciascun rettangolo della griglia in tanti rettangoli verticali quanti sono i bin in soluzione.
- Disposizione di ciascuno degli oggetti di uno stesso bin in un diverso rettangolo della griglia in maniera casuale in rettangoli verticali corrispondenti.

La costruzione della griglia dipende quindi dal numero di righe in cui si vuole che sia divisa e da quanti rettangoli compaiono in una riga.

L'inserimento degli oggetti di uno stesso bin in rettangoli verticali corrispondenti serve per evitare che gli oggetti si dispongano in modo ravvicinato se sono posizionati in rettangoli adiacenti della griglia principale.

Nella figura 4.1 si riporta la disposizione dei clienti nell'istanza 1 della libreria 1. L'istanza ha 45 clienti e 15 bin nella soluzione di perfect packing. Al centro del quadrato si trova il deposito; i clienti sono contrassegnati con lo stesso simbolo se compaiono nello stesso bin. Si constata visivamente che i clienti che devono essere serviti con lo stesso veicolo sono relativamente lontani.

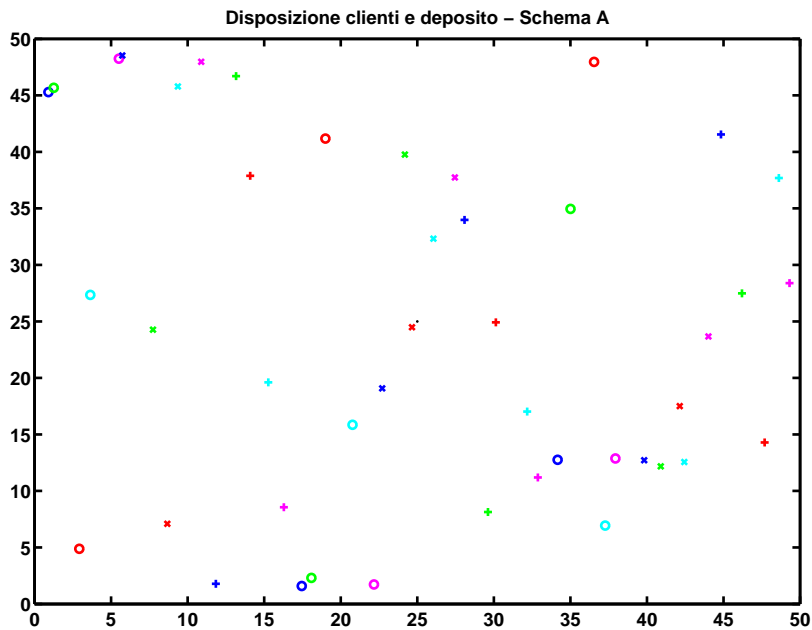


Figura 4.1: Istanza 1 libreria 1 - Difficile per il routing

Nel caso della libreria 2 di istanze il numero di oggetti per bin non è costante. Per ciascun bin della soluzione di perfect packing è stata creata una griglia con un numero di rettangoli pari al numero di oggetti per bin e sono stati disposti gli oggetti del bin nei diversi rettangoli della griglia in maniera casuale.

La figura 4.2 riporta l'istanza 1 di questa libreria in disposizione difficile per

il routing.

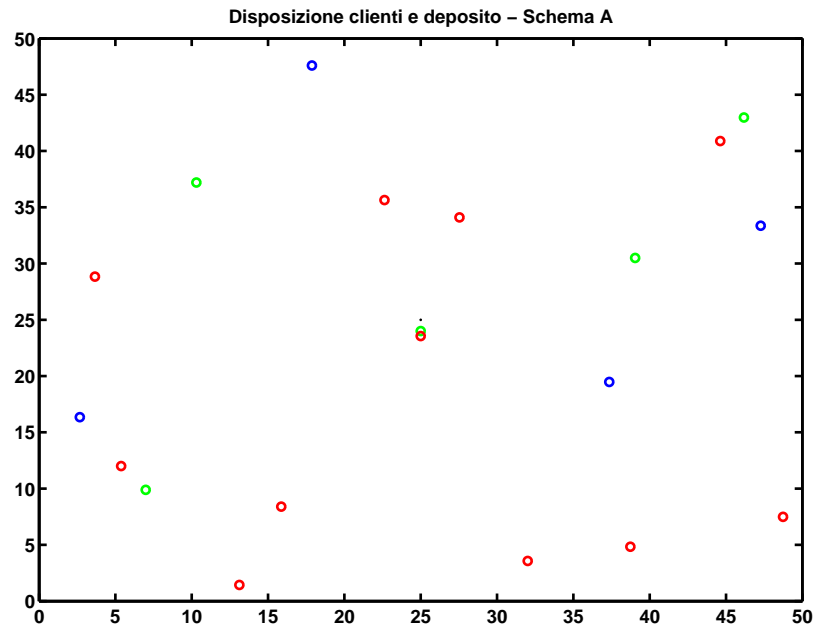


Figura 4.2: Istanza 1 libreria 2 - Difficile per il routing

La disposizione di difficoltà intermedia si ottiene disponendo i clienti in maniera casuale nel quadrato Q . In figura 4.3 si riporta, a titolo di esempio, la configurazione dei clienti dell'istanza 1 della libreria 1 secondo questo livello di difficoltà.

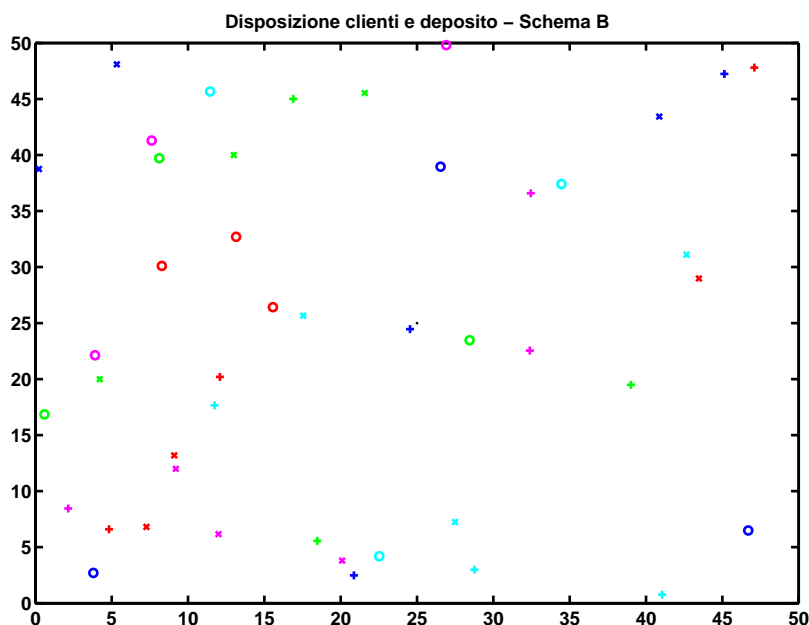


Figura 4.3: Istanza 1 libreria 1 - Difficoltà intermedia per il routing

Il procedimento seguito per aggiungere le coordinate spaziali agli oggetti ed ottenere una configurazione facile per il routing è stato il seguente:

- Partizione di Q in una griglia dotata di tanti rettangoli quanti sono i bin della soluzione di perfect packing.
- Disposizione degli oggetti di uno stesso bin in un unico rettangolo della griglia in maniera casuale.

Come per la configurazione difficile per il routing, c'è un punto di indeterminazione dovuto alla scelta del numero di righe e colonne della griglia.

In figura 4.4 si presenta la disposizione facile per il routing costruita dagli oggetti dell'istanza 1 della libreria 1. Si notano gli agglomerati di clienti raggruppati per veicolo.

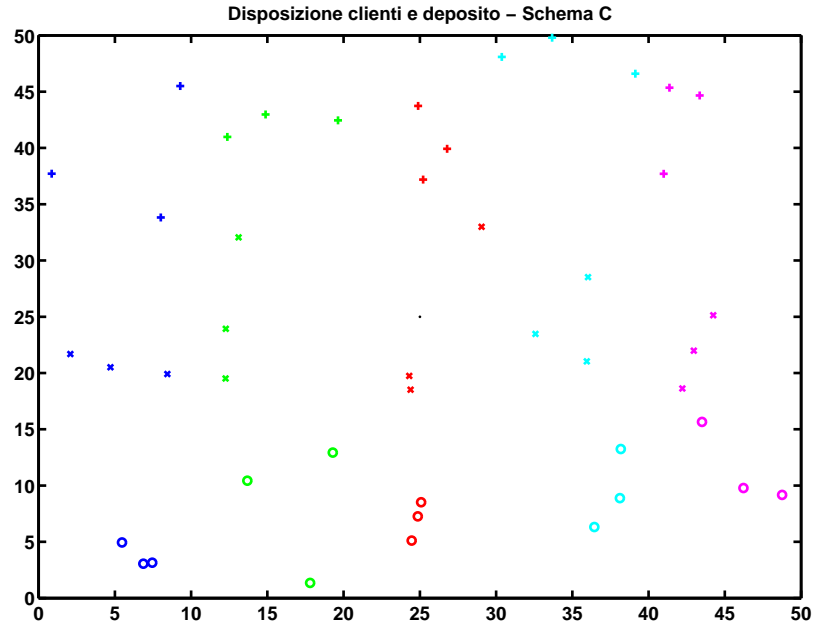


Figura 4.4: Istanza 1 libreria 1 - Facile per il routing

4.2 Risultati algoritmo Roberti

Si riportano ora alcuni dati sulla risoluzione delle istanze delle librerie 1 e 2 con l'algoritmo di Roberti.

Nelle seguenti tabelle si presentano il nome dell'istanza, il valore della soluzione del CVRP, un'indicazione del fatto che la soluzione trovata corrisponda a quella di perfect packing nota, una descrizione dei tipi di route che hanno consentito di trovare la soluzione ottima o dei bound ed il tempo di esecuzione.

| Istanza | Ottimo | Soluzione | Risoluzione con | Tempo (s) |
|-----------|--------|------------------------|----------------------------------|-----------|
| Ist1Lib1A | 1175 | Non nota | q -path | 2.137 |
| Ist1Lib1B | 1228 | Non nota, diversa da A | q -routes | 2.169 |
| Ist1Lib1C | 747 | Nota | q -routes | 2.168 |
| Ist2Lib1A | 1060 | Non nota | Bound routes elementari: 1050 | 3.010 |
| Ist2Lib1B | 1169 | Non nota, diversa da A | Bound routes elementari: 1156.83 | 3.526 |
| Ist2Lib1C | 720 | Nota | Routes elementari | 2.122 |
| Ist5Lib1A | 1435 | Non nota | Bound routes elementari: 1420 | 11.408 |
| Ist5Lib1B | 1232 | Non nota, diversa da A | Bound routes elementari: 1220 | 4.021 |
| Ist5Lib1C | 911 | Nota | Routes elementari | 4.021 |
| Ist6Lib1A | 927 | Non nota | Bound routes elementari: 913 | 3.693 |
| Ist6Lib1B | 880 | Non nota, diversa da A | Bound routes elementari: 867 | 3.535 |
| Ist6Lib1C | 737 | Nota | Routes elementari | 2.212 |
| Ist1Lib2A | 395 | Nota | Routes elementari | 30.155 |
| Ist1Lib2B | 328 | Nota | Routes elementari | 7.675 |
| Ist1Lib2C | 232 | Nota | Routes elementari | 1.856 |
| Ist2Lib2B | 349 | Nota | Routes elementari | 9.877 |
| Ist2Lib2C | 222 | Nota | Routes elementari | 3.042 |
| Ist3Lib2A | 377 | Nota | Routes elementari | 1.996 |
| Ist3Lib2B | 294 | Nota | Routes elementari | 1.825 |
| Ist3Lib2C | 242 | Nota | q -routes | 1.794 |
| Ist4Lib2A | 428 | Nota | Routes elementari | 4.826 |
| Ist4Lib2B | 321 | Nota | Routes elementari | 1.879 |
| Ist4Lib2C | 267 | Nota | Routes elementari | 6.229 |
| Ist5Lib2A | 304 | Nota | q -routes | 1.908 |
| Ist5Lib2B | 370 | Nota | Routes elementari | 6.413 |
| Ist5Lib2C | 213 | Nota | Routes elementari | 1.993 |

Tabella 4.1: Risultati Roberti

Si può affermare che, sia per le istanze della libreria 1 che per quelle della libreria 2, non c'è una grossa differenza in termini di difficoltà di risoluzione

tra schemi di routing A,B e C di una stessa istanza di packing, come invece ci si aspettava. In molti casi, lo schema A, pensato come difficile per il routing, non richiede molto più tempo delle configurazioni B e C relative alla stessa istanza di packing.

Nel complesso, le istanze si sono dimostrate facili per l'algoritmo: le route elementari o i q -path e i q -routes consentono di trovare la soluzione ottima o comunque dei bound molto buoni. Si ritiene quindi che l'algoritmo di generazione di colonne riesca a costruire facilmente colonne relative alla soluzione ottima di CVRP, proprio come accadeva per il column generation per le istanze BPP. E' stato, però, rilevato dal professor Roberti che la generazione delle routes elementari nelle istanze Ist3Lib2 non è stata banale.

Si osserva che, per le istanze delle libreria 1 con schemi A e B di routing, la soluzione nota di perfect packing non viene selezionata come soluzione CVRP. Le istanze BBP create nel capitolo 3 devono quindi ammettere varie soluzioni di perfect packing; vengono così indebolite le convinzioni sulle difficoltà in termini di packing delle istanze.

Le istanze Ist3Lib1, Ist4Lib1 e Ist7Lib1 non sono state risolte perchè il loro elevato numero di clienti (superiore a 100) richiederebbe al codice di Roberti l'uso di funzionalità aggiuntive che allo stato attuale non si sono riuscite a sfruttare. Si ritiene comunque che per queste istanze il bound dato dal Set Partitioning dovrebbe lasciare un gap, piuttosto elevato, dell'1 - 2%.

4.3 Osservazioni conclusive sui risultati ottenuti

Sulla base dei risultati visti nella precedente sezione, si pensa che il motivo della "facilità" delle istanze per l'algoritmo di Roberti possa essere il seguente: nella costruzione delle route, siccome il vincolo di capacità non è rilassato, ma anzi molto stringente, sia più conveniente costruire routes massimali per capacità. Infatti, la generazione di routes non massimali, nel caso di istanze con soluzioni di perfect packing, imporrebbe di selezionare un numero di veicoli più grande del valore di input dell'algoritmo; questo procedimento

dovrebbe venire quindi immediatamente accantonato dall'algoritmo, che si concentrerebbe sulla creazione di route massimali per capacità, promettenti per essere soluzioni del CVRP.

Si deve anche considerare che per le istanze della Libreria 1 il numero di oggetti per bin è basso (3 o 4) e questo potrebbe facilitare ulteriormente la generazione delle routes. Si spiegherebbe anche la maggiore difficoltà di costruzione delle route elementari segnalata nella Ist3Lib2: per le istanze della libreria 2 il numero di oggetti per bin è variabile e talvolta supera le 10 unità.

La questione del gap si può spiegare con il fatto che, quando il numero di clienti è elevato, selezionando le colonne in maniera frazionaria aumenta la probabilità di costruire delle routes non elementari di costo inferiore a quello della soluzione ottima.

Bibliografia

- [1] Baldacci, Mingozzi, and Roberti: Route Relaxation and Pricing Strategies for VRP, *Operations Research* 59(5), pp. 1269-1283 (2011).
- [2] James Ostowski, Jeff Linderoth, Fabrizio Rossi, Stefano Smeriglio, *Operations Research* 39(2011), pp. 127-131.
- [3] Appunti sul Vehicle Routing Problem, Marco Pranzo - Corso di Ottimizzazione su Reti Università degli Studi di Siena Anno Accademico 2007/2008.
- [4] Dispense del Corso di Metodi e Modelli di Supporto alle Decisioni, Alberto Caprara.
- [5] Il Vehicle Routing Problem: Modelli e Algoritmi, Daniele Vigo, D.E.I.S - Università di Bologna
- [6] Algoritmi euristici per il Vehicle Routing Problem, Michele Aldà
- [7] James Ostowski, Jeff Linderoth, Fabrizio Rossi, Stefano Smeriglio, *Constraint Orbital Branching* (2007)

Ringraziamenti

Desidero ringraziare il professore Andrea Lodi per la costante disponibilità e l'impegno profuso nel seguire l'attività di tesi con chiaro interesse e partecipazione attiva. Ringrazio l'ingegnere Enrico Malaguti per il prezioso aiuto fornito nell'utilizzo dell'algoritmo a generazione di colonne per il Bin Packing, nonostante la mia inesperienza con l'ambiente usato. Ringrazio inoltre il dr. Roberto Roberti e il dr. Jeff Linderoth per la collaborazione offerta per i risultati dell'algoritmo per le istanze CVRP e per i tentativi di impiego delle tecniche di orbital branching.

Grazie ai miei genitori, a Luca e Silvia, per avermi permesso di proseguire i miei studi assecondando i miei interessi ed inclinazioni personali e per avermi sempre offerto un sicuro appoggio morale nei momenti difficili.

Grazie a Diego, Giovanni, Alessio, Marco, Andrea, Vale, Francesca e Alice per tutto quello che avete voluto condividere insieme con me in questi anni.