

## Università della Calabria

Dipartimento di Elettronica, Informatica e Sistemistica

Dottorato di Ricerca in: *Ricerca Operativa*

Ciclo XXIII

Settore Scientifico Disciplinare: Ricerca Operativa (MAT/09)

# Metodi euristici per la soluzione dello Stochastic Dial a Ride Problem

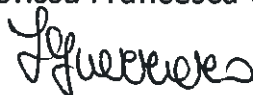
Dottoranda:

Maria Grazia Grande



Supervisore:

Prof.ssa Francesca Guerriero



Coordinatore del corso di Dottorato:

Prof. Lucio Grandinetti



Anno Accademico: 2010/2011

INDICE	Pag.
Introduzione.....	3
1. Dial a Ride Problem: Stato dell'Arte.....	6
1.1. Introduzione.....	6
1.2. Caratteristiche principali del DARP.....	8
1.3. Single Vehicle DARP (DARP mono-veicolo).....	9
1.3.1. Caso statico.....	10
1.3.2. Caso dinamico.....	11
1.4. DARP multi veicolo.....	12
1.4.1. Caso statico.....	12
1.4.2. Caso dinamico.....	16
1.5. DARP stocastico.....	21
2. La Programmazione Stocastica.....	24
2.1. Introduzione.....	24
2.2. Modellazione di un problema di Programmazione Stocastica.....	26
2.2.1. Programmazione stocastica a due stadi con ricorso.....	26
2.2.2. Problema di Programmazione stocastica multistadio.....	31
2.2.3. L'approccio a scenari.....	35
2.2.3.1. Classificazione degli scenari.....	36
2.2.3.2. Analisi e valutazione degli scenari.....	38
2.2.4. Modelli con vincoli probabilistici.....	39
3. Il DARP in condizioni di incertezza.....	40
3.1. Definizione del problema DARP.....	40
3.2. Formulazione deterministica del problema DARP.....	42
3.3. Definizione del DARP stocastico (Stochastic DARP) a due stadi.....	43

<b>3.4. Formulazione stocastica del DARP.....</b>	<b>48</b>
<b>4. Algoritmo risolutivo per lo Stochastic DARP.....</b>	<b>52</b>
<b>4.1. Schema Euristica 1.....</b>	<b>52</b>
<b>4.2. Descrizione della Procedura per la fusione delle soluzioni.....</b>	<b>54</b>
<b>4.3. Esempio operativo.....</b>	<b>56</b>
<b>4.4. Motivazione generale sulla scelta euristica e sul blocco clienti.....</b>	<b>63</b>
<b>5. Descrizione esperimenti, risultati e valutazioni.....</b>	<b>64</b>
<b>5.1. Introduzione.....</b>	<b>64</b>
<b>5.2. Validazione della qualità della soluzione euristica ottenuta su piccole istanze.....</b>	<b>64</b>
<b>5.3. Validazione dell'euristica su istanze tratte dalla letteratura: confronto con un'altra euristica "base".....</b>	<b>70</b>
<b>5.4. Risultati Euristica 1.....</b>	<b>73</b>
<b>5.5 Risultati dell' Euristica di confronto.....</b>	<b>79</b>
<b>5.6. Confronto delle due euristiche (Euristica 1, Euristica 2) .....</b>	<b>84</b>
<b>6. Conclusioni.....</b>	<b>86</b>
<b>Appendice.....</b>	<b>88</b>
<b>Bibliografia.....</b>	<b>96</b>

## Introduzione.

L'attività di ricerca presentata in questa tesi è focalizzata sullo studio del Dial A Ride Problem (DARP) in condizioni di incertezza. La versione deterministica di tale problema appartiene alla classe dei problemi relativi alla ottimizzazione dei servizi di recapito veloce su brevi distanze. (e.g. un servizio di fattorini che consegnano delle merci in una regione urbana). Più specificatamente, nei problemi di tipo DARP è necessario soddisfare  $N$  richieste, dove ogni richiesta consiste nel ritirare della merce da un nodo  $i$ ,  $i \in 1, \dots, N$ , e nel consegnarla al nodo  $j$ ,  $j \in N + 1, \dots, 2N$ , utilizzando una flotta di  $K$  veicoli. Ogni veicolo parte dal deposito (nodo  $0$ ), ed eseguirà una ed una sola rotta  $r_k$ , dove  $r_k$  consiste in una successione di nodi di carico e scarico (per ogni richiesta il nodo di carico deve essere visitato prima del corrispettivo nodo di scarico).

I vincoli operativi solitamente considerati nel DARP sono i seguenti: ogni veicolo può trasportare al più una quantità di merce pari a  $q_k$ , ogni corsa non può durare oltre il tempo limite  $T$ , ogni nodo deve essere visitato all'interno di una finestra temporale  $[et, st]$ , ed infine la merce non può viaggiare per un tempo superiore a  $W$ . Obiettivo del DARP è la minimizzazione dei costi di trasporto.

A partire da tale ambito, l'attività di ricerca si è focalizzata su una variante del problema in esame maggiormente attinente ai casi reali, in cui la natura stocastica delle richieste è presa in considerazione. Tale problema sarà nel seguito denominato "Stochastic DARP".

Lo Stochastic DARP è una variante del DARP in cui un sottoinsieme delle richieste non è noto con certezza. La soddisfazione delle nuove richieste può quindi richiedere la modifica delle rotte quando queste sono già in corso. Uno dei possibili modi per rappresentare la probabilità congiunta associata alla comparsa di una serie di richieste è la definizione di un set di scenari futuri  $S = \{s_1, \dots, s_n\}$ ; ogni scenario corrisponde ad un set di richieste che si manifesteranno dopo il tempo  $H$ ; ad ogni scenario è inoltre associata una probabilità  $p_s$  di realizzazione.

Il DARP stocastico può essere rappresentato mediante il framework della programmazione stocastica a due stadi con ricorso. La soluzione di primo stadio corrisponde a  $K$  rotte che servono i vertici già conosciuti a priori. Le soluzioni di secondo stadio corrispondono invece alle modifiche da effettuare alla soluzione di primo stadio a seconda dello scenario che effettivamente si realizzerà. L'obiettivo del DARP stocastico è la minimizzazione del costo atteso, dove per costo atteso si intende il costo della soluzione di primo stadio sommato al costo delle modifiche da effettuare in ogni scenario  $s_s$  pesato per la rispettiva probabilità  $p_s$  (costo atteso).

Per risolvere in modo efficiente il problema descritto è stata ideata ed implementata una euristica basata su due passi successivi: la determinazione prima di una soluzione di primo stadio ed in seguito la determinazione delle relative soluzioni di secondo stadio. Tramite una procedura di "fusione delle soluzioni", ideata ad hoc, l'algoritmo cerca di ottenere una soluzione di primo stadio che sia quanto più "baricentrica" possibile rispetto alle soluzioni di secondo stadio.

Al fine di confrontare la bontà dell'euristica proposta, è stato considerato un altro approccio euristico risolutivo che ha permesso di confrontare le soluzioni ottenute sia in termini di costo atteso, sia in termini di requisiti computazionali.

L'approccio euristico risolutivo alternativo trova soluzioni di primo stadio non baricentriche in quanto costruite senza tener conto degli scenari. La modifica della rotta di primo stadio  $SL_0^0$  dunque risulta maggiormente costosa nel momento in cui uno scenario si realizzi (il costo atteso è maggiore).

La tesi è strutturata nel seguente modo: il capitolo 1 delinea lo stato dell'arte sui metodi risolutivi per il DARP (e alcune sue varianti). Il capitolo 2 descrive la programmazione stocastica che si occupa dei problemi di ottimizzazione in cui intervengono in modo esplicito componenti caratterizzate da incertezza.

Il capitolo 3 descrive la formulazione del DARP stocastico oggetto di questa tesi con la relativa formulazione matematica. Il capitolo 4 descrive nel dettaglio l'euristica utilizzata e specifica la motivazione della scelta fatta rispetto ad approcci già esistenti. Il capitolo 5 descrive nel dettaglio la fase sperimentale: viene presentato il metodo utilizzato per la creazione delle istanze, il protocollo sperimentale adottato e i

risultati delle sperimentazioni effettuate. In particolare, i risultati dell'euristica sono stati confrontati su piccole istanze con i risultati ottenuti mediante il solver Cplex, mentre su istanze di maggiore dimensione con i risultati ottenuti dall'euristica di confronto senza fusione delle soluzioni.

# 1. Dial a Ride Problem: Stato dell'Arte

## 1.1. Introduzione

Alcuni sistemi logistici prevedono il servizio di recapito veloce di pacchi o colli su brevi distanze. Un esempio pratico è rappresentato da un servizio di fattorini che consegnano delle merci in una regione urbana.

Tali tipi di servizi di consegna rapida solitamente funzionano nel seguente modo: una centrale operativa riceve le chiamate dei clienti e schedula le consegne nel rispetto dei vincoli temporali dettati dal cliente stesso e dalle possibilità operative del servizio (e.g. numero di fattorini, orario di lavoro, etc.). Ogni cliente indica inoltre un luogo dove la merce deve essere ritirata (carico) e un luogo dove deve essere effettuata la consegna (consegna). Ogni fattorino esegue più ritiri/consegne durante ogni sua corsa; quindi ogni veicolo dovrà visitare un certo numero  $n$  di luoghi da cui ritirare della merce (pacchi, colli o persone), e un egual numero di luoghi dove la merce andrà consegnata. Nel seguito del testo il termine "luogo" verrà sostituito con il più generico termine "vertice", sottintendendo quindi l'esistenza di una rete logistica.

I vincoli operativi sono molteplici:

- ogni veicolo può trasportare solo una limitata quantità di merce;
- ogni corsa non può durare oltre un certo limite (per esempio le 8 ore lavorative giornaliere);
- ad ogni vertice corrisponde una determinata finestra temporale, ovvero la merce può essere ritirata e consegnata solo in determinate fasce orarie.

Nell'ambito della Ricerca Operativa tali servizi e le problematiche gestionali annesse vengono solitamente formulati come Dial a Ride Problem (DARP).

Il Dial-a-Ride Problem (DARP) consiste nel progettare le rotte di veicoli e gli orari per  $n$  utenti che specificano richieste di ritiro e consegna tra le origini e le destinazioni.

Spesso lo stesso utente avrà due richieste durante lo stesso giorno: una richiesta in uscita dalla sede di destinazione (ad esempio, un ospedale), e una richiesta in ingresso per il viaggio di ritorno.

Nella versione standard, il servizio è fornito da una flotta di  $m$  veicoli identici con sede presso lo stesso deposito. L'obiettivo è quello di pianificare una serie di rotte di costo minimo nel rispetto di una serie di vincoli (soddisfacimento delle richieste, rispetto della capacità di carico dei veicoli, rispetto delle finestre temporali, etc.).

Ogni rotta è composta da una serie di "nodi". Per nodo si intende una sosta presso un determinato vertice; ogni nodo è quindi contraddistinto da:

- vertice visitato;
- istante di arrivo;
- istante di partenza;
- tempo di servizio.

Ovviamente, l'istante di arrivo e di partenza di ogni nodo devono essere inclusi all'interno della finestra temporale del rispettivo vertice visitato, al fine di ottenere una soluzione ammissibile.

Sono possibili diverse formulazioni ed applicazioni alternative per tale problema; l'esempio più comune è rappresentato dai servizi di trasporto porta a porta di persone anziane o disabili (si veda, ad esempio, Madsen et al.1995; Toth e Vigo 1996, 1997; Borndörfer et al. 1997, Colorni e Righini 2001; Diana e Dessouky 2004; Rekiek et al. 2006; Melachrinoudis et al. 2007).

Molti enti locali sono orientati a fornire servizi di dial-a-ride o a potenziare servizi già esistenti in risposta alla crescente domanda. Questo fenomeno può essere attribuito in parte all'invecchiamento della popolazione, ma anche ad una tendenza verso lo sviluppo dei servizi sanitari ambulatoriali. Alcuni sistemi esistenti non riescono a soddisfare adeguatamente la domanda mentre altri si trovano ad affrontare crescenti costi di gestione.



## 1.2. Caratteristiche principali del DARP

I servizi Dial-a-ride possono operare secondo una modalità statica o dinamica. Nel primo caso, tutte le richieste di trasporto sono note in anticipo, mentre nel secondo caso le richieste vengono rese note gradualmente nel corso della giornata e i percorsi dei veicoli sono regolati in tempo reale per soddisfare la domanda. Più recentemente, inoltre, è stato introdotto anche il caso del DARP stocastico; in questo caso alcune caratteristiche del problema non sono note a priori, ma le funzioni di probabilità legate alle quantità non note sono invece conosciute e possono essere utilizzate per ottimizzare il problema. Un esempio comune è il caso in cui non si conoscono le richieste in anticipo, ma sono note le probabilità associate alla realizzazione di ogni richiesta.

Nella maggior parte degli studi sul DARP si assume la disponibilità di una flotta di  $m$  veicoli omogenei con sede presso un unico deposito. Nella realtà possono esistere casi in cui si hanno diversi depositi, in particolare in vaste aree geografiche, e la flotta è eterogenea per poter effettuare tipi di trasporto differenti.

L'obiettivo principale in alcuni problemi è quello di determinare la dimensione della flotta e la composizione in grado di soddisfare tutta la domanda, mentre in altri contesti, l'obiettivo è quello di massimizzare il numero di richieste che possono essere servite con una flotta di dimensioni fisse.

Si possono quindi considerare due possibili problemi in cui gli obiettivi possono essere: minimizzare i costi legati alla piena soddisfazione della domanda e dei vincoli ad essi annessi; massimizzare la soddisfazione della domanda soggetta alla disponibilità dei veicoli e dei vincoli ad essi associati. Un compromesso consiste nel servire alcuni clienti con una flotta di veicoli proprietari e, se necessario, utilizzare veicoli esterni.

Gli elementi di costo più comuni riguardano il dimensionamento ed il funzionamento della flotta, l'uso occasionale degli extra veicoli ed il pagamento dei trasportatori.

La qualità del servizio include diversi criteri quali la durata dell'itinerario, la lunghezza della rotta, il tempo di attesa dei clienti, il tempo di corsa del cliente (ad esempio, il tempo totale trascorso nei veicoli) e

la differenza tra i tempi di consegna reali e desiderati. Alcuni di questi criteri possono essere trattati come vincoli o come parte della funzione obiettivo.

Una tendenza comune nei modelli DARP è quella di consentire agli utenti di imporre una finestra temporale sugli orari di partenza e di arrivo, ma questo può essere eccessivamente vincolante per il trasportatore soprattutto se queste finestre di tempo sono strette.

Jaw et al. (1986) ha affrontato il problema considerando che gli utenti dovrebbero essere in grado di specificare la finestra temporale sul tempo di arrivo del loro viaggio di andata e l'orario di partenza del loro viaggio di ritorno. In tal caso il trasportatore determina quindi un programma sull'orario di partenza per il viaggio di andata e sull'orario di arrivo previsto per il viaggio di ritorno, soddisfacendo il limite superiore del tempo di corsa. Uno dei casi più semplici del DARP è quello in cui tutti gli utenti sono serviti da un singolo veicolo.

Il Dial a Ride Problem è stato estensivamente studiato nel corso degli ultimi trenta anni; le ricerche si sono concentrate sia sulla formulazione di versioni alternative del problema, capaci di modellare diverse tipologie di vincoli operativi, sia sulla progettazione di algoritmi risolutivi (esatti o euristici) per la soluzione delle formulazioni proposte.

### **1.3. Single Vehicle DARP (DARP mono-veicolo)**

Uno dei casi più semplici del DARP è quello in cui tutti gli utenti sono serviti da un singolo veicolo. Per tale caso sono stati ideati algoritmi sia per il caso statico, dove tutte le richieste sono note in anticipo, sia per il caso dinamico, dove le richieste sono gradualmente rivelate in tempo reale.

La maggior parte delle ricerche in tale ambito si sono concentrate sulla versione statica del DARP anche se inizialmente (Wilson et al, 1971;. Wilson e Weissberg, 1976; Wilson e Colvin, 1976) sono state cercate soluzioni in tempo reale per il DARP dinamico.

### **1.3.1. Caso statico**

Psaraftis (1980) ha formulato e risolto il problema con un programma dinamico in cui la funzione obiettivo è la minimizzazione della somma ponderata del tempo di completamento della rotta e della insoddisfazione del cliente. L'insoddisfazione del cliente è espressa come una combinazione ponderata del tempo di attesa prima del pick-up e del tempo di corsa.

Psaraftis (1980, 1983) ha sviluppato un algoritmo esatto di programmazione dinamica per il caso in cui le finestre temporali sono imposte su ogni pick-up and delivery. Il problema per l'utente è controllato attraverso vincoli sulla "massima posizione di spostamento" limitando la differenza tra la posizione di un utente nella lista di richieste e la sua posizione nella rotta del veicolo.

Questo algoritmo è stato poi aggiornato dallo stesso autore (Psaraftis 1983) per gestire finestre temporali specificate dall'utente sugli orari di partenza e arrivo. Come spesso accade nelle formulazioni di programmazione dinamica, l'algoritmo può risolvere solo casi relativamente piccoli in modo ottimale in quanto la procedura ha una complessità  $O(n^2 3^n)$ . Il problema più grande che è stato risolto con questo algoritmo prevede solo nove utenti. Tale approccio potrebbe rivelarsi utile se si considerano sotto rotte in un algoritmo multi-veicolo, a condizione che il numero di gli utenti in ogni rotta rimanga relativamente piccolo.

Sexton (1979) e Sexton and Bodin (1985a, 1985b) hanno affrontato il DARP mono-veicolo come un singolo passo all'interno dell'euristica per la soluzione del DARP multi-veicolo; tale euristica per il DARP multi-veicolo effettua anche un clustering dei clienti. Il loro algoritmo itera risolvendo un problema di routing per mezzo di un inserzione euristica e il problema di scheduling associato basando l'alternanza di questi due passi nel contesto di decomposizione Benders. La funzione obiettivo minimizza i disagi dell'utente considerando la somma pesata di due termini: il primo misura la differenza tra il tempo di viaggio effettivo e il tempo di viaggio diretto di un utente; il secondo termine è la differenza (positiva) tra il tempo di consegna desiderato e tempi di consegna effettiva, sotto l'ipotesi che il primo è almeno grande

quanto il secondo, essendo non consentite consegne in ritardo. I risultati sono stati riportati su diversi set di dati da Baltimora e Gaithersburgh, dove il numero di utenti varia tra i sette e i venti.

Il DARP mono-veicolo è stato riformulato come un problema di programmazione intera da Desrosiers et al.(1986). La formulazione comprende vincoli sulle finestre temporali, sulla capacità dei veicoli e vincoli di precedenza. Tale problema è stato risolto con la programmazione dinamica. Utilizzando un sistema di etichettatura doppia, gli autori sono stati in grado di identificare ed eliminare diversi stati dominati e di transizione. Soluzioni ottimali sono stati ottenuti per  $n = 40$ .

### **1.3.2. Caso dinamico**

Il DARP mono-veicolo dinamico è stato considerato da Psaraftis (1980). In questo problema, nuove richieste si rivelano dinamicamente nel tempo ma nessuna informazione sulle future richieste è disponibile (a differenza di quanto accade nella programmazione stocastica). Quando una nuova richiesta diventa nota al tempo  $t$ , è disponibile una soluzione pianificata. Tutte le richieste in programma prima di  $t$  sono già state trattate e non sono più rilevanti. Il problema è riottimizzare delle rotte considerando le nuove richieste. Ciò è stato fatto applicando un algoritmo di programmazione dinamica sviluppato per il caso statico.

Una delle difficoltà pratiche derivanti da questo approccio è la capacità di risolvere il problema al tempo  $t$  prima dell'arrivo della prossima richiesta, che non può essere fattibile se l'algoritmo è lento e le richieste arrivano in rapida successione.

Un modo per aggirare questa difficoltà, recentemente proposto da Gendreau et al. (2001) nel contesto di delocalizzazione dinamica dell'ambulanza, è quello di precomporre diversi scenari, utilizzando in calcolo parallelo, in previsione delle richieste future.

Nonostante i suoi limiti, Psaraftis, con il suo lavoro su DARP dinamico mono-veicolo, ha contribuito a definire i concetti utilizzati in seguito nella ricerca sui problemi di routing dinamico (si veda, ad esempio, Psaraftis 1988, 1995;. Mitrovi'c-Mini'c et al 2004).

#### **1.4. DARP multi veicolo**

Un'altra versione del DARP, per molti versi più realistica, consiste nel considerare non un singolo veicolo ma una intera flotta di mezzi. Sono state effettuate ricerche sia per il caso statico che dinamico.

##### **1.4.1. Caso statico**

Una delle prime euristiche per il DARP multiple-veicolo statico è stata proposta da Jaw et al. (1986). Il modello considerato da questi autori impone finestre sui tempi di pickup (prelievo) delle richieste in entrata e sui tempi di delivery (consegna) delle richieste in uscita. Per ogni utente si impone un tempo di corsa al massimo, espresso come funzione lineare del tempo di corsa diretta.

Inoltre, i veicoli non possono essere inattivi durante il trasporto dei passeggeri. Per valutare la qualità delle soluzioni è stata utilizzata una funzione obiettivo non-lineare combinando diversi tipi di disutilità. L euristica seleziona gli utenti in ordine di tempo e gradualmente li inserisce nelle rotte del veicolo in modo da produrre il meno possibile un incremento della funzione obiettivo. L'algoritmo è stato testato su istanze artificiali che coinvolgono 250 utenti e su un set di dati reali con 2617 utenti e 28 veicoli.

Una tecnica comunemente utilizzata in tali problemi consiste nel definire gruppi di utenti da servire dallo stesso veicolo, prima della fase di routing. Questa idea è sfruttata da Bodin e Sexton (1986) che hanno sviluppato un metodo basato sui clusters, utilizzando un inserimento euristico sequenziale per formare cluster di utenti. Un miglioramento di questo approccio è stato presentato da Dumas et al. (1989), con una euristica a due stadi basata su una tecnica di raggruppamento di richieste e una procedura di routing. Questo approccio a due fasi avviene creando cosiddetti "mini-gruppi" di utenti, cioè gruppi di utenti da servire all'interno della stessa area approssimativamente nello stesso tempo. Questi mini-gruppi sono poi combinati in modo ottimale per formare possibili rotte dei veicoli, utilizzando una tecnica di generazione di colonne. Infine, ogni rotta è ottimizzata nuovamente attraverso l' algoritmo del mono-veicolo di Desrosiers et al. (1986). Gli autori hanno risolto con successo casi derivati dalla vita reale attraverso i dati provenienti da tre città canadesi: Montreal, Sherbrooke e Toronto. Istanze con un

massimo di 200 utenti sono state risolte facilmente, mentre per grandi casi si richiede l'uso di una tecnica di decomposizione temporale. La fase di mini-raggruppamento è stata in seguito migliorata da Desrosiers et al. (1991) che hanno presentato i risultati su una serie di dati costituiti da quasi 3000 utenti.

Infine, Ioachim et al. (1995) hanno mostrato vantaggio, in termini di qualità soluzione, di ricorrere ad una tecnica di ottimizzazione per costruire i cluster. Un problema reale sorto a Bologna è stato affrontato da Toth e Vigo (1996). In tale problema gli utenti specificano le finestre temporali sull'origine e sulla destinazione. Un limite proporzionale alla distanza diretta è imposto sul tempo di corsa ed il trasporto è fornito da una flotta di minibus e automobili speciali. In più occasioni, possono essere usati i taxi, ma poiché questi non sono la migliore modalità di trasporto per le persone disabili, è imposta una penalità sul loro uso. L'obiettivo è minimizzare il costo totale del servizio.

Il problema è stato risolto con una euristica costituita da una procedura di inserimento in parallelo, ed i risultati sono stati riportati su istanze che coinvolgono tra 276 e 312 richieste.

Toth e Vigo (1997) hanno sviluppato una euristica composta da una prima assegnazione delle richieste alle rotte per mezzo di una procedura di inserzione parallela, e poi dall'esecuzione di scambi intra-rotta e inter-rotta. I test effettuati hanno evidenziato significativi miglioramenti rispetto alle precedenti soluzioni.

Inoltre, Borndorfer et al. (1997) hanno riportato i risultati di uno studio per il trasporto di portatori di handicap a Berlino. Il metodo proposto applica il branch-and-cut per risolvere due formulazioni di partizionamento del problema: uno per la creazione di cluster e un altro per la concatenazione di questi gruppi nelle rotte dei veicoli. Un cluster è definito come una "sotto rotta massima tale che il veicolo non è mai vuoto." Nella prima fase, viene costruito un insieme di "buoni" cluster ed è risolto un problema di set partitioning per selezionare un sottoinsieme di cluster, servendo ogni utente esattamente una volta.

Nella seconda fase, sono elencate le rotte ammissibili dalla combinazione di cluster ed è risolto un secondo problema di partizionamento per selezionare le migliori rotte che coprono ogni cluster esattamente una volta. Entrambi i problemi di partizionamento sono risolti da un algoritmo di branch-and-

cut. I risultati sono stati riportati per casi reali che coinvolgendo tra 859 e 1765 richieste di trasporto al giorno.

Cordeau e Laporte (2003a) hanno proposto una euristica per il DARP statico multi-veicolo applicando la tabu search. Gli utenti specificano le finestre temporali di arrivo e di partenza e un tempo di corsa massimo è associato ad ogni richiesta. Tale tempo può essere lo stesso per tutte le richieste, o calcolato utilizzando un fattore di scostamento massimo dal tempo di corsa più diretta di ogni particolare rotta.

Sui veicoli sono imposti vincoli sulla capacità e la massima lunghezza della rotta. L'algoritmo di ricerca rimuove iterativamente una richiesta di trasporto e la reinserisce in un'altra rotta.

Come è consuetudine negli schemi tabu (Gendreau et al 1994, Cordeau et al 2001), soluzioni intermedie non ammissibili sono consentite attraverso l'utilizzo di una funzione obiettivo penalizzata.

L'algoritmo è stato testato su istanze generati casualmente ( $24 \leq n \leq 144$ ) e su sei set di dati ( $n = 200$  e  $295$ ) forniti da un trasportatore danese.

Nell'articolo di Aldaihani e Dessouky (2003) è stato preso in esame un DARP ibrido in cui sono considerati due tipi di servizi offerti: servizi di trasporto porta a porta e servizi di trasporto da e verso una fermata dell'autobus. La scelta tra le due modalità è effettuata all'interno dell'algoritmo. Gli autori hanno sviluppato una euristica in due parti che consiste in un inserimento-reinserimento nella fase di discesa, seguita dalla tabu search. Sono perseguiti due obiettivi: minimizzazione della distanza percorsa dai veicoli e del tempo di viaggio dei passeggeri. In entrambi i casi la funzione obiettivo è considerata nelle diverse parti degli algoritmi. Gli autori hanno riportato i risultati utilizzando i dati di Lancaster County in California.

Diana and Dessouky (2004) hanno applicato una euristica parallela del tipo "regret insertion" basata su una procedura di inserzione per risolvere una versione del DARP in cui l'obiettivo è una somma ponderata di distanza, tempo di corsa in eccesso sul tempo di corsa diretto, tempo di inattività del veicolo. Sono stati riportati risultati su istanze di dimensioni 500 e 1000.

Nel 2006, sono state pubblicate quattro euristiche sul DARP multi-veicolo. Rekiek et al. (2006), hanno affrontato il problema con l'obiettivo di minimizzazione del numero di veicoli usati. Gli autori hanno

proposto un algoritmo genetico per la fase di clustering e un meccanismo di inserimento per la fase di routing. Sono stati generati buoni risultati sui dati forniti dal Comune di Bruxelles ( $100 \leq n \leq 164$ ).

Xiang et al. (2006) hanno risolto una versione elaborata della DARP in cui l'obiettivo è la minimizzazione di una combinazione dei costi fissi dei veicoli, dei costi variabili del veicolo, dei costi dei trasportatori, dei tempi di attesa e dei tempi di assistenza in base ai vincoli operativi. Nell'algoritmo proposto sono utilizzati inserimenti e scambi intra-rotta per costruire i percorsi. Gli autori hanno introdotto un elemento di diversificazione nel meccanismo di ricerca utilizzando una funzione obiettivo secondaria focalizzata sui tempi di inattività. Sono state risolte istanze contenenti tra 50 e 2000 richieste.

L'obiettivo considerato da Wong e Bell (2006) è la minimizzazione di una combinazione lineare del tempo operativo totale, del tempo di corsa dei passeggeri e dei costi dei veicoli esterni per le richieste non assegnate. Gli autori hanno considerato inoltre diversi tipi di veicolo e la durata massima della rotta. Nel problema studiato alcuni veicoli sono specificatamente utilizzati per l'accesso ai disabili. Gli autori hanno proposto una procedura parallela di inserimento. Gli utenti vengono prima classificati in base ad un indice che misura la difficoltà e i disagi causati ad altre richieste quando sono inserite in una rotta. Le inserzioni vengono effettuate considerando prima le richieste più difficili. Dopo l'inserzione si considera una fase di post-ottimizzazione. L'algoritmo è stato testato su casi non reali che coinvolgono 150 richieste.

Wolfler Calvo e Colorni (2006) hanno ideato un metodo euristico per una versione del DARP in cui è fissato il numero di veicoli disponibili e sono imposte finestre temporali sui tempi al ritiro e di consegna. È utilizzata una funzione obiettivo gerarchica: algoritmo prima prova a servire più clienti possibile e poi minimizza il disagio per gli utenti espresso come la somma dei tempi di attesa e del tempo di permanenza in corsa in eccesso. L'euristica prima costruisce una serie di  $m$  rotte ed una serie di sotto-rotte risolvendo un problema di assegnazione; poi esegue una fase per inserire le sotto-rotte nelle  $m$  rotte e schedula nuovamente i vertici all'interno delle rotte. I test sono stati effettuati su casi che coinvolgono tra i 10 e 180 utenti.



Cordeau (2006) e Röpke et al. (2007) hanno formulato il problema come un problema di programmazione lineare intera (con alcune variabili continue nel primo caso). Diverse famiglie di disuguaglianze valide sono proposte per ciascun modello. Inizialmente, i modelli sono risolti rilassando alcuni vincoli. Durante il processo di ramificazione, sono applicati gli algoritmi di separazione per individuare i vincoli violati tra quelli che inizialmente erano rilassati o tra le disuguaglianze valide. Questi vincoli sono poi introdotti e il processo si conclude quando l'albero di ricerca è stato analizzato secondo le regole del branch-and-bound. La più grande istanza risolta con il primo algoritmo è costituita da 36 richieste, mentre con il secondo l'algoritmo è costituita da 96 richieste.

Più recentemente, Melachrinoudis et al. (2007) hanno proposto un metodo euristico per una versione del DARP con finestre temporali flessibili provenienti da una organizzazione sanitaria. In questo problema, l'obiettivo è minimizzare una combinazione lineare dei costi di trasporto e dei disagi degli utenti. La loro euristica utilizza la tabu search con reinserimenti di richieste. Sono state risolte istanze contenenti fino a 50 richieste.

Infine, Jørgensen et al. (2007) hanno lavorato con una funzione obiettivo multi-critero contenente sette termini. La loro euristica alterna un meccanismo di ricerca genetica per costruire cluster e una procedura modificata dell'intorno più vicino per costruire le rotte. Sono stati risolti casi contenenti richieste tra 24 e 144.

#### **1.4.2. Caso dinamico**

Come notato da Borndörfer et al. (1997), la distinzione tra DARP statico e dinamico è spesso confusa, in quanto le domande sono molte volte annullate e, di conseguenza, i trasportatori possono consentire l'introduzione di nuove richieste in una soluzione progettata per un problema statico.

Inoltre, il DARP dinamico raramente esiste in una forma pura dal momento che un numero di richieste sono spesso note all'inizio della pianificazione.

Un interessante DARP multi-veicolo dinamico (*Multi Vehicle Dinamic DARP*) è stato descritto da Madsen et al. (1995) che hanno risolto un problema reale sui servizi per le persone anziane e disabili a Copenhagen. In tale problema gli utenti possono specificare o il tempo di pick-up desiderato o il tempo di consegna, ma non entrambi. Veicoli di diversi tipi sono utilizzati per fornire servizi, non tutti disponibili in ogni momento. Le richieste arrivano in modo dinamico tutto il giorno e le velocità dei veicoli sono variabili. Gli autori hanno sviluppato un algoritmo di inserimento, chiamato REBUS, basato sulla procedura precedentemente sviluppata da Jaw et al. (1986). Le nuove richieste sono dinamicamente inserite nelle rotte del veicolo tenendo conto delle loro difficoltà di inserimento nella rotta già esistente. L'algoritmo è stato testato su un esempio con 300 clienti e 24 veicoli. Gli autori hanno evidenziato che l'algoritmo è in grado di generare soluzioni di buona qualità con tempi di calcolo brevi.

Dumas et al. (1991) hanno esteso l'algoritmo esatto per il singolo veicolo al caso di veicolo multiplo applicandolo alle istanze con  $n \leq 55$ . Un altro caso con diversi veicoli eterogenei basati su un unico deposito è stato affrontato da Teodorovic e Radivojevic (1998). Nel problema da loro preso in esame quando si verificano nuove richieste, la prima fase consiste nell'assegnare ogni richiesta ad un veicolo. In una seconda fase è ottimizzato lo scheduling della rotta selezionata. Entrambi gli algoritmi sono basati sulla logica fuzzy. Il problema risolto da Teodorovic e Radivojevic (2000) considera un set di tre criteri che devono essere minimizzati: la distanza totale percorsa, i tempi di attesa dei veicoli e i tempi di corsa dei passeggeri. Ogni volta che arriva una nuova richiesta, questa è inserita in una delle rotte dei veicoli secondo una delle regole con cui si valuta se l'inserimento della richiesta in una determinata rotta produrrebbe un "piccolo", "medio" o "grande" aumento della distanza percorsa o del tempo di attesa di un veicolo. A seconda dell'esito, l'algoritmo assegna una preferenza "molto debole", "debole," media, "forte" o "molto forte" a ciascun veicolo. Tutti i numeri utilizzati nella valutazione sono calcolati mediante funzioni fuzzy. Regole simili sono state usate per determinare la posizione della richiesta nel percorso selezionato. Il sistema è stato testato su istanze artificiali contenenti 900 richieste.

Colorni e Righini (2001) hanno considerato tre diversi obiettivi: la massimizzazione delle richieste di manutenzione, la massimizzazione del livello di servizio percepito dagli utenti e la minimizzazione della distanza percorsa. Il loro sistema presuppone che una trattativa con gli utenti venga posta al fine di dissuaderli da finestre temporali troppo strette. Il meccanismo di inserimento si alterna tra una fase di clustering e una fase di routing. L'algoritmo di routing applica il branch-and-bound per una serie di richieste il cui le finestre temporali sono non troppo lontane nel futuro. La fase di clustering funziona con gli scambi. Gli autori hanno effettuato esperimenti con il loro sistema (DARIA) a Crema e Verbania, situata nel nord Italia, ma non è stato riportato nessun risultato computazionale.

Di recente è stato proposto un algoritmo da Coslovich et al. (2006) che segue una strategia in due fasi per l'inserimento di una nuova richiesta in una rotta già esistente. Prima viene utilizzata una fase off-line per creare un intorno ammissibile della rotta corrente attraverso un meccanismo 2-opt; poi viene utilizzata una fase on-line per inserire la nuova richiesta con l'obiettivo di minimizzare l'insoddisfazione degli utenti.

Lo stato dell'arte per il DARP dinamico è particolarmente vasto. L'euristica sviluppata da Madsen et al. (1995) si applica al DARP dinamico del trasporto di persone disabili. Oltre ai vincoli sulle finestre temporali e a una flotta eterogenea, sono considerati vincoli di capacità multi-dimensionale e le priorità dei clienti. Gli autori propongono un algoritmo basato su una euristica di inserimento, chiamata REBUS, per risolvere il problema in un ambiente dinamico con un tempo di calcolo breve.

In Gendreau et al.(1999), hanno affrontato il problema considerando che tutti i clienti non sono noti in anticipo e solo alcuni di essi si aggiungono on-line. Rispetto alla versione originale dell'algoritmo, un'applicazione parallela basata su Badeau et al., (1997) è stata utilizzata per tener conto del contesto dinamico. L'idea principale per interagire con l'ambiente quando non si verifica l'evento è di ottimizzare le rotte attraverso la tabu search. Quando si verifica un evento, i passi da applicare sono i seguenti: fermare l'ottimizzazione, l'aggiornare la memoria e le soluzioni a seconda del caso trattato e riavviare

l'ottimizzazione. Solo due eventi sono considerati: l'arrivo di un nuovo cliente o la fine del servizio in un cliente destinazione.

Ichoua et al. (2000) hanno cercato di risolvere il DARP dinamico introducendo delle deviazioni nelle rotte. Queste deviazioni permettono di migliorare i risultati. Il miglioramento consiste nel deviare un veicolo lontano dalla sua destinazione prevista per servire un cliente vicino, che si è appena rivelato.

Horn et al. (2002) hanno proposto un sistema di pianificazione denominato *L2sched*: quando si verifica una nuova richiesta, questa è inserita in una rotta corrente utilizzando uno schema di inserzione a costo minimo. Le rotte sono inoltre migliorate grazie ad una ricerca locale.

Un caso più generale del DARP è stato presentato da Attanasio et al. (2004). L'obiettivo del problema è massimizzare il numero totale di richieste accettate arrivate on-line. Per risolvere questo problema, gli autori hanno confrontato diverse modalità di parallelizzazione di una tabu search ispirata da Cordeau e Laporte, (2003).

Coslovich et al. (2005) hanno esaminato un caso speciale in cui le richieste sono emesse "*on the fly*" da parte dei clienti alle fermate dei veicoli. Il conducente del veicolo accetta o rifiuta la richiesta. Gli autori hanno sviluppato un metodo di soluzione a due fasi: la prima fase consiste nel testare l'inserimento di un nuovo cliente nel percorso attuale in modo che il guidatore possa rispondere a questa domanda in fretta; la seconda fase è gestita off-line quando il veicolo si muove tra due fermate successive ed è volta a creare un intorno ammissibile di rotta corrente.

Un'altra estensione è stata presentata da Gendreau et al. (2006). In questa estensione, il cliente è rappresentato da una domanda che deve essere servita da una location di pick-up ad una di delivery. Tale problema è una tipica applicazione di servizio di un corriere in ambiente dinamico.

Melachrinoudis et al. (2007) hanno considerato il DARP in un contesto di organizzazione sanitaria. In questo problema, lo stesso veicolo viene utilizzato due volte per lo stesso paziente. Gli autori hanno proposto una formulazione basata sulla programmazione intera e un algoritmo di ricerca tabu.

Un ulteriore studio sul DARP dinamico è stato proposto da Xiang et al. (2008). Gli autori hanno preso in considerazione diversi tipi di eventi possibili (nuove esigenze, la cancellazione di richieste, i veicoli ripartizione, ecc.) e hanno risolto questo problema costruendo un algoritmo basato su un caso statico del problema studiato precedentemente (Xiang et al., 2006).

Beaudry et al. (2008) hanno sviluppato un algoritmo euristico per un DARP dinamico. L'obiettivo è l'inserimento di nuove richieste in percorsi già esistenti rispettando alcuni vincoli, quali l'isolamento del paziente o il requisito di articoli medicali. Gli autori hanno presentato una euristica ad inserimenti più un algoritmo di tabu search.

Hanne et al. (2009) hanno considerato il problema focalizzandosi sulla gestione del servizio di trasporto di alcuni pazienti tra diversi ospedali e tra diversi edifici dello stesso ospedale, utilizzando sia dei veicoli (con diversi pazienti per veicolo, e configurazioni diverse dei veicoli) sia il trasporto a piedi (barelle, letti, o sedie a rotelle). Le richieste hanno le stesse caratteristiche contemplate da Beaudry et al. (2008) (finestre temporali, differenti modi di trasporto, attrezzature necessarie, ecc) in un ambiente dinamico. Le soluzioni sono state valutate sulla base dei disagi dei pazienti e i costi ospedalieri. Gli autori hanno presentato quindi un sistema di pianificazione dei trasporti chiamato Opti-TRANS come soluzione del problema considerato. Questo sistema permette di prenotare, pianificare, spedire, controllare e segnalare le richieste di trasporto. Alcuni algoritmi di ottimizzazione sono adoperati in Opti – TRANS: due metodi per l'assegnazione della domanda, tre metodi per il routing e la programmazione delle decisioni e un algoritmo evolutivo. Con Opti-TRANS, gli autori hanno notato un miglioramento della soddisfazione del paziente e una riduzione dei costi di trasporto.

Per ulteriori approfondimenti, segnaliamo che Cordeau e Laporte (2007) che con i loro studi sul caso offrono un'ampia analisi sullo stato dell'arte di modelli e algoritmi per il DARP nelle sue varie versioni, mentre Berbeglia et al. (2010) presentano una rassegna della letteratura di problemi di pickup and delivery.

### 1.5. DARP stocastico

Più recentemente, le ricerche si sono orientate al caso in cui tutte o parte delle richieste non sono note in anticipo, ma vengono rivelate durante la giornata lavorativa. Tali problemi possono essere formulati come DARP stocastico oppure come DARP dinamico. Nel caso del DARP dinamico, le richieste aggiuntive arrivano casualmente durante l'arco della giornata, e non si ha conoscenza a priori della loro possibile distribuzione. Nel caso del DARP stocastico le probabilità di realizzazione delle data richiesta (richieste considerate singolarmente o in gruppi) è nota a priori, e questa informazione può essere utilizzata durante la soluzione del problema.

Per quanto concerne il DARP stocastico, il numero di lavori presenti in letteratura è considerevolmente limitato. Il primo contributo in questo ambito è rappresentato da Swihart e Papastavrou (1999). Nel loro modello gli autori hanno considerato il caso in cui un singolo veicolo che deve servire delle richieste che vengono rese note dinamicamente secondo un processo poissoniano ed i cui nodi di pick up e delivery sono uniformemente distribuiti su una regione.

Fu (2002) ha proposto invece un modello in cui i tempi di percorrenza delle strade sono modellati secondo una data funzione di probabilità. In tal modo il modello è capace di rappresentare casi di congestione stradale. Il problema è stato risolto tramite un algoritmo euristico mutuato dai problemi di scheduling.

Un altro algoritmo ispirato allo scheduling è stato proposto da Xiang et al. (2008); nel loro modello sono stati considerati differenti eventi stocastici, come ad esempio nuove richieste o rotture dei veicoli e le performances del modello rispetto ai diversi tipi di eventi stocastici sono state illustrate attraverso una completa gamma di sperimentazioni su dati simulati.

Un algoritmo risolutivo totalmente differente è invece stato adottato da Cremers et al. (2009); nel loro lavoro gli autori hanno considerato un modello ispirato dai servizi di trasporto di pazienti anziani, in cui le richieste possono arrivare inaspettatamente (anche se si suppone che l'aleatorietà delle richieste sia

in qualche modo modellabile). Per risolvere il problema è stato proposto un algoritmo basato su clustering e algoritmi genetici.

Un esempio di problema maggiormente attinente allo scopo della presente tesi è dato da Haugland et al. (2009); in questo lavoro è stato considerato il classico problema DARP in cui ad ogni richiesta è stata associata una probabilità di realizzazione. È stato proposto e valutato un algoritmo euristico basato sulla ispezione dell'intorno.

Hyytia et al. (2010) hanno presentato un interessante problema in cui è considerato un solo veicolo e dove i tempi di servizio di ogni cliente possono essere alterati (le alterazioni seguono una data funzione di probabilità) in seguito alla accettazione di un nuovo cliente. Il modello è stato risolto applicando una tecnica risolutiva nota sviluppata per il "*Erlang loss problem*".

Infine, nel caso di Heilporn et al. (2011) la stocasticità è invece stata considerata nel tempo di arrivo dei clienti alle loro sedi di pick up; uno specifico metodo basato sulla tecnica "L-shaped" è stato proposto per cercare la soluzione del modello.

Il problema affrontato in questa tesi è innovativo sotto diversi aspetti rispetto allo stato dell'arte. Nella formulazione qui proposta, parte delle richieste sono note in anticipo, mentre le rimanenti richieste (dette "*richieste aleatorie*") vengono rese tutte note solo in un dato momento della giornata. Si suppone inoltre che la probabilità di realizzazione delle richieste aleatorie sia nota e modellata introducendo il concetto di "*scenario*". Uno scenario consiste in una possibile realizzazione del vettore di richieste aleatorie, vettore al quale viene associata una probabilità di realizzazione.

Per comprendere appieno l'utilità di tale modello, si pensi ad una azienda di trasporti con un ampio portafoglio di clienti abituali; tale azienda potrebbe non conoscere completamente gli ordini da eseguire durante la giornata, ma sulla base della propria esperienza passata può essere in grado di prevedere, almeno parzialmente, quali ordini verranno ulteriormente richiesti.

Si noti nuovamente la differenza con il problema DARP dinamico, dove tutte le richieste arrivano in maniera casuale e non si hanno informazioni a priori rispetto alla loro distribuzione.

Infine, rispetto al presente stato dell'arte questa tesi presenta anche una innovativa metodologia euristica di risoluzione, basata sulla applicazione della tabu search. Per quanto la tabu search sia stata già ampiamente sperimentata nell'ambito del DARP, in questo lavoro si propone una nuova procedura in grado di tenere conto della stocasticità del problema attraverso la fusione delle soluzioni fornite dalla tabu search.

L'ipotesi che si vuole confermare in questo lavoro consiste nel dimostrare che tale procedura di fusione consente di trovare delle soluzioni migliori della semplice applicazione della euristica tabu. Per maggiori dettagli sull'euristica proposta, si veda il Capitolo 4.

Prima di descrivere nel dettaglio il problema preso in esame, è opportuno parlare della programmazione stocastica trattata in maniera approfondita nel Capitolo 2 che segue.



## 2. La Programmazione Stocastica

### 2.1. Introduzione

La programmazione stocastica si occupa dei problemi di ottimizzazione in cui intervengono in modo esplicito delle componenti caratterizzate da incertezza. L'incertezza viene delineata in termini probabilistici attraverso la definizione di variabili aleatorie di cui si suppone nota la funzione di distribuzione di probabilità. Un elemento chiave nella descrizione dei modelli di decisione in condizione di incertezza è rappresentato dalla descrizione della struttura informativa ovvero dal processo che governa l'arrivo di nuove informazioni e il loro utilizzo all'interno del processo decisionale. Tale struttura fornisce la relazione, fondamentale all'interno di un processo di decisione multiperiodale in condizione di incertezza, tra la componente dinamica e la componente stocastica.

A tale riguardo è possibile individuare almeno tre categorie di problemi che si caratterizzano per una diversa interazione tra aspetto dinamico e aspetto stocastico:

- Modelli anticipativi
- Modelli adattivi
- Modelli con revisione.

Assumiamo che la componente stocastica del modello possa essere riassunta in un vettore di variabili casuali indicate con  $\omega$ .

Nei modelli di tipo anticipativo (ad esempio quelli statici), le decisioni relative a tutti gli istanti successivi vengono assunte prima che si renda disponibile qualsiasi informazione futura e mantenute nel medio periodo. La decisione deve essere assunta prima che sia possibile alcuna osservazione sulla variabile casuale  $\omega$ ; essa non dipende dunque dalle future realizzazioni ma cerca di anticiparle. La decisione che si ottiene risulta quindi "ottima" in relazione alla capacità di analisi dei possibili sviluppi futuri. Una formulazione, frequente in questi modelli, coinvolge vincoli di tipo probabilistico.

A questa prima categoria di modelli si contrappongono i modelli di tipo adattivo nei quali l'arrivo dell'informazione avviene in modo progressivo e la politica ottima risulta da un processo del tipo decisione-osservazione.

In particolare, nei modelli adattivi l'ottimizzazione avviene in un ambiente in cui l'incertezza può essere parzialmente ridotta dall'osservazione delle realizzazioni delle quantità aleatorie.

Infine nei modelli con revisione si ottiene una integrazione delle due precedenti tipologie. L'ottimizzazione in questo caso avviene in un ambiente caratterizzato da incertezza e da apprendimento progressivo. In particolare ogni decisione deve essere assunta condizionatamente alle informazioni già disponibili e in modo tale da anticipare possibili realizzazioni future.

In questo caso si parla anche di recourse problems, nei quali le decisioni dei periodi successivi dipendono dalle informazioni che via via divengono disponibili.

Una attività di revisione basata sulla riapplicazione di modelli statici può rivelarsi insufficiente per gestire l'incertezza. In particolare le decisioni ottime su un orizzonte uniperiodale possono risultare miopi rispetto alle possibili evoluzioni future e quindi subottimali su un orizzonte multiperiodale.

Si rende quindi chiaramente necessaria l'introduzione di modelli dinamici che permettano di cogliere l'evoluzione del sistema. La struttura temporale con cui l'informazione si rende disponibile costituisce un primo fondamentale elemento nella descrizione dei modelli intertemporali, sulla base di tale struttura si possono descrivere modelli decisionali diversi.

Nella struttura generale del problema con revisione multiperiodale la struttura di dipendenza temporale tra decisioni e realizzazioni delle variabili casuali può essere espressa nel seguente modo

$$(x_1, \varepsilon_1) \rightarrow (x_2(x_1, \varepsilon_1), \varepsilon_2) \rightarrow \dots \rightarrow x_T(x_1, \dots, x_{T-1}, \varepsilon_1, \dots, \varepsilon_{T-1})$$

dove  $x_t$  ed  $\varepsilon_t$  rappresentano rispettivamente il vettore delle variabili decisionali ed il vettore delle realizzazioni delle variabili aleatorie al tempo  $t$ , con  $t = 1, \dots, T$ .

In ciascun periodo la decisione ottima dipende dalle decisioni precedenti e dalle realizzazioni della componente stocastica fino a quel momento osservate.

## 2.2. Modellazione di un problema di Programmazione Stocastica

Dato uno spazio di probabilità  $(\Omega, F, P)$ , con  $\omega \in \Omega$ , un problema di programmazione stocastica lineare (PSL) può essere scritto in come:

$$\min c^T(\omega) x \quad (1)$$

$$\text{s.v.} \quad T(\omega)x = h(\omega) \quad (2)$$

$$x \in X_1 \quad (3)$$

dove  $X_1$  è l'insieme dei vincoli deterministici (in forma compatta); (2) è l'insieme dei vincoli stocastici, con i coefficienti della matrice  $T$  e del vettore dei termini  $h$  dipendenti dall'esito dell'evento aleatorio  $\omega$  che rappresenta la componente stocastica del problema. La funzione obiettivo prevede la minimizzazione dei costi attesi (o la massimizzazione dei rendimenti attesi).

Si tratta di una formulazione generale dalla quale possono essere ottenuti i problemi di programmazione stocastica più noti in letteratura sia statici che dinamici, con vincoli probabilistici e non, e con revisione.

In letteratura esistono essenzialmente due approcci: i modelli con ricorso e i modelli con vincoli probabilistici.

### 2.2.1. Programmazione stocastica a due stadi con ricorso

Frequentemente l'approccio seguito per modellare un problema stocastico coinvolge una scomposizione in almeno due stadi decisionali. Si fissa la prima decisione e dopo che i parametri incerti sono conosciuti si prende la seconda decisione. Certe decisioni devono essere prese al presente nonostante l'incertezza, dopo che l'incertezza sarà risolta si potranno assumere azioni correttive.

Le decisioni del primo stadio sono rappresentate dal vettore  $x$  mentre le decisioni del secondo stadio dal vettore  $y$  oppure  $y(\omega, x)$ . La successione degli eventi e delle decisioni è sintetizzata attraverso il seguente sistema:

$$x \rightarrow \text{osservazione di } \omega (\xi(\omega)) \rightarrow y(\omega, x)$$

Le variabili  $x$ , dette di primo stadio, rappresentano le decisioni anticipative (fase anticipativa), mentre le variabili  $y$ , dette di secondo stadio, indicano le decisioni adattative o di ricorso, ovvero le decisioni prese per correggere quelle anticipative una volta noto il risultato dell'evento aleatorio (fase adattativa).

Il problema di programmazione stocastica a due stadi può essere così rappresentato:

$$\begin{aligned} \min \quad & c^T x + E[Q(x, \xi(\omega))] & (4) \\ \text{s.v.} \quad & Ax = b \\ & x \in \mathfrak{R}_+^{n1} \end{aligned}$$

$Q(x, \xi(\omega))$  rappresenta il valore ottimo del problema di secondo stadio, per un dato  $\omega$ , e può essere così esplicitato:

$$Q(x, \xi(\omega)) = \min\{q(\omega)^T y \mid W(\omega)y = h(\omega) - T(\omega)x, y \in \mathfrak{R}_+^{n2}\} \quad (5)$$

Nella formulazione (4) si ha:

- $x \in \mathfrak{R}^{n1}$                       decisioni di primo stadio;
- $c \in \mathfrak{R}^{n1}$                         vettore dei coefficienti di costo nel primo stadio;
- $\omega \in \Omega$                          evento casuale del secondo stadio;
- $q(\omega) \in \mathfrak{R}^{n2}$                  funzione di costo del secondo stadio;
- $y(\omega) \in \mathfrak{R}^{n2}$                  decisioni del secondo stadio;
- $A \in \mathfrak{R}^{m1 \times n1}$                  matrice dei coefficienti nel primo stadio sui vincoli deterministici;
- $T(\omega) \in \mathfrak{R}^{m2 \times n1}$             matrice dei coefficienti nel secondo stadio;
- $W(\omega) \in \mathfrak{R}^{m2 \times n2}$            matrice di ricorso;

- $h(\omega) \in \mathfrak{R}^{m_2}$  vettore nel secondo stadio.

Raggruppando le componenti aleatorie  $h, q, T, W$  si ottiene il vettore:

$$\xi^T = \xi^T(\omega) = (q(\omega)^T, h(\omega)^T, T_1(\omega), \dots, T_{m_2}(\omega), W_1(\omega), \dots, W_{m_2}(\omega))$$

dove  $T_i(\omega)$  e  $W_i(\omega)$  rappresentano rispettivamente la  $i$ -esima riga delle matrici  $T(\omega)$  e  $W(\omega)$ . Il vettore delle variabili  $\xi$  ha potenzialmente componenti  $N = (n_2 + m_2 + m_2 * (n_1 + n_2))$ .

Tale formulazione è valida sia per variabili aleatorie discrete che continue anche se le prime sono quelle più ampiamente utilizzate. Nel caso di variabili discrete la funzione di secondo stadio diventa:

$$Q(x, \xi_l) = \min_{y^l} \{q_l^T y^l \mid W^l y^l = h^l - T^l x, y^l \geq 0\}$$

Il problema (4) può essere riformulato come:

$$\begin{aligned} \min \quad & c^T x + \sum_{l=1}^N p_l q_l^T y^l \quad (6) \\ \text{s.v.} \quad & Ax = b \\ & T^l x + W^l y^l = h^l, \quad l = 1, \dots, N \\ & x \geq 0 \\ & y^l \geq 0, \quad l = 1, \dots, N \end{aligned}$$

In tale formulazione si assume che:

- le variabili aleatorie abbiano una distribuzione di probabilità discreta con un numero finito  $N$  di realizzazioni  $\xi_l = (q_l, h^l, T^l, W^l), l = 1, \dots, N$  chiamati scenari con le corrispondenti probabilità  $p_l$ .
- $y^l$  rappresentino le variabili di secondo stadio associate allo scenario  $l$ .
- $T^l$  sia la matrice delle tecnologie.
- $W^l$  sia la matrice del ricorso.

Da notare che se la matrice  $W$  è costante il modello è chiamato problema con ricorso fisso, mentre se  $W = (I, -I)$ , dove  $I$  è la matrice identità, il problema è denominato con ricorso fisso semplice.

Una delle caratteristiche più importanti della formulazione dei modelli di programmazione stocastica è la non anticipatività. Tale concetto riflette la restrizione in base alla quale sotto incertezza, le decisioni di primo stadio devono essere implementate prima delle realizzazioni delle variabili aleatorie. In assenza di tale restrizione il problema (6) potrebbe essere separato in  $N$  sottoproblemi di dimensione inferiore, uno per ogni scenario  $l$ .

L'ipotesi di distribuzioni discrete per la componente stocastica del modello permette di trasformare problemi di programmazione stocastica multi periodale in problemi di grandi dimensioni, nello specifico in problemi di programmazione lineare di larga scala, per la risoluzione dei quali possono, almeno in linea di principio essere utilizzate degli algoritmi generali.

Nella pratica tuttavia tali vie si dimostrano del tutto impraticabili poiché le dimensioni dei problemi che risultano dalla trasformazione deterministica equivalente crescono esponenzialmente all'aumentare del numero di periodi considerati e del numero di possibili realizzazioni delle quantità aleatorie per ogni periodo diventando presto ingestibili.

Per questo sono state cercate altre tecniche di risoluzione al fine di trarre vantaggio dalla struttura del problema; una delle tecniche utilizzate è decomporre il problema stesso in sottoproblemi più facilmente trattabili da un punto di vista numerico.

Diverse tecniche numeriche sono state sviluppate per la risoluzione di problemi di programmazione stocastica multistadio.

I metodi proposti possono essere raggruppati in due grandi categorie: *metodi di decomposizione* e *metodi a punti interni*, di seguito analizzati.

### *Metodi di decomposizione*

I metodi di decomposizione si fondano sull'idea di suddividere il problema originario in sottoproblemi di dimensione ridotta e quindi più facilmente risolvibili. Sono classificati in metodi primali e duali. Nei primali la decomposizione è fatta rispetto allo stadio. Il metodo più noto di decomposizione primale è l'algoritmo *L-Shaped* introdotto nel 1969 da Van Slyke e Wets. Si tratta di un metodo per la risoluzione dei problemi di programmazione stocastica a due stadi attraverso una linearizzazione esterna del problema basata sulla *decomposizione di Benders*.

L'idea fondamentale alla base della *decomposizione di Benders* è quella di approssimare il termine nonlineare della funzione obiettivo del problema con revisione.

Nei metodi duali la decomposizione è fatta rispetto agli scenari; in questo caso il problema deterministico equivalente viene suddiviso in un numero di sottoproblemi pari al numero degli scenari considerati ciascuno risolto in modo indipendente, le soluzioni scenario-dipendenti vengono quindi aggregate sulla base di un opportuno criterio.

Uno degli algoritmi utilizzati per la decomposizione duale è quello di *Copertura Progressiva* ideato da Rockafellar e Wets. Tale metodo si basa sulla costruzione di una funzione lagrangiana aumentata ottenuta con l'introduzione di vincoli di non-anticipatività all'interno della funzione obiettivo opportunamente pesati attraverso un parametro di penalità.

Il notevole vantaggio associato a questa tecnica deriva dal fatto che il problema relativo ad uno scenario risulta completamente deterministico in tutti i suoi elementi e non presenta quindi particolari difficoltà risolutive. Di contro si pone il problema della convergenza delle singole soluzioni verso una soluzione unica che sia implementabile.

Il problema principale con cui si scontrano tali tecniche deriva dalla presenza di vincoli di non-anticipatività che legano tra loro le soluzioni in diversi scenari.

### *Metodi a punti interni*

I metodi a punti interni operano direttamente sul problema di origine riconoscendone la particolare struttura duale blocco-angolare. L'applicazione di tale metodo per problemi di PLS a due stadi è motivata dalla efficienza di tali algoritmi per la risoluzione di problemi di grande dimensione caratterizzati da una particolare struttura della matrice dei vincoli.

### **2.2.2. Problema di Programmazione stocastica multistadio**

Il modello a due stadi può essere esteso ad un modello multistadio considerando un orizzonte temporale discreto e finito. Ciò può essere intuitivamente schematizzato come segue:

$$\dots \text{decisione } x_{t-1} \rightarrow \text{osservazione di } \xi_t \rightarrow \text{decisione } x_t \rightarrow \dots$$

dove:

- $x \in \mathfrak{R}^{nt}$ ,  $t = 1, \dots, T$  rappresentano le decisioni prese al  $t$ -esimo stadio;
- $\xi_t$  indicano i vettori aleatori le cui realizzazioni divengono note all'inizio dello stadio  $t$

La formulazione di un problema di programmazione stocastica multistadio include in modo esplicito la componente di natura aleatoria che determina l'evoluzione nel tempo del sistema che si analizza, assegnando una struttura di probabilità. Si comprende quindi come sia fondamentale fornire al modello una rappresentazione il più possibile accurata dell'evoluzione del sistema.

La difficoltà di ottenere delle previsioni puntuali attendibili per le grandezze stocastiche da un lato, e l'impossibilità pratica di considerare delle distribuzioni continue per tali grandezze, possono essere in parte superate considerando un approccio all'incertezza basato sugli scenari.

Si definisce uno scenario come una possibile traiettoria in  $(0, T)$  del processo stocastico discreto, cioè una possibile realizzazione del processo  $\xi_t$  che descrive il comportamento della componente aleatoria del problema.



La struttura delle possibili evoluzioni future del sistema può essere agevolmente rappresentata mediante una struttura ad albero. Un esempio di tale rappresentazione è illustrato in Figura 1. Ogni scenario è rappresentato come un ramo con origine nel nodo-radice ( $I$ ) dell'albero e si sviluppa nel tempo attraverso una successione di nodi che corrispondono agli stadi decisionali. Il nodo radice ( $I$ ) si riferisce al primo periodo e corrisponde ai dati deterministici. I nodi ai livelli  $t \geq 1$  corrispondono a possibili realizzazioni di  $\xi_t$ . I nodi che non hanno successori sono detti foglie dell'albero. Uno scenario è definito come un cammino dal nodo radice ad un nodo foglia e rappresenta una realizzazione congiunta dei parametri del problema per tutti gli stadi  $t = 1, \dots, T$ .

In particolare l'albero considerato si estende sull'orizzonte temporale  $[0,3]$ , con dodici scenari. Il nodo A è un nodo relativo al primo stadio decisionale mentre i nodi B e C sono relativi al secondo.

L'albero rappresenta quindi una struttura con sei scenari nell'intervallo  $[0,2]$  e dodici scenari nell'intervallo  $[0,3]$ .

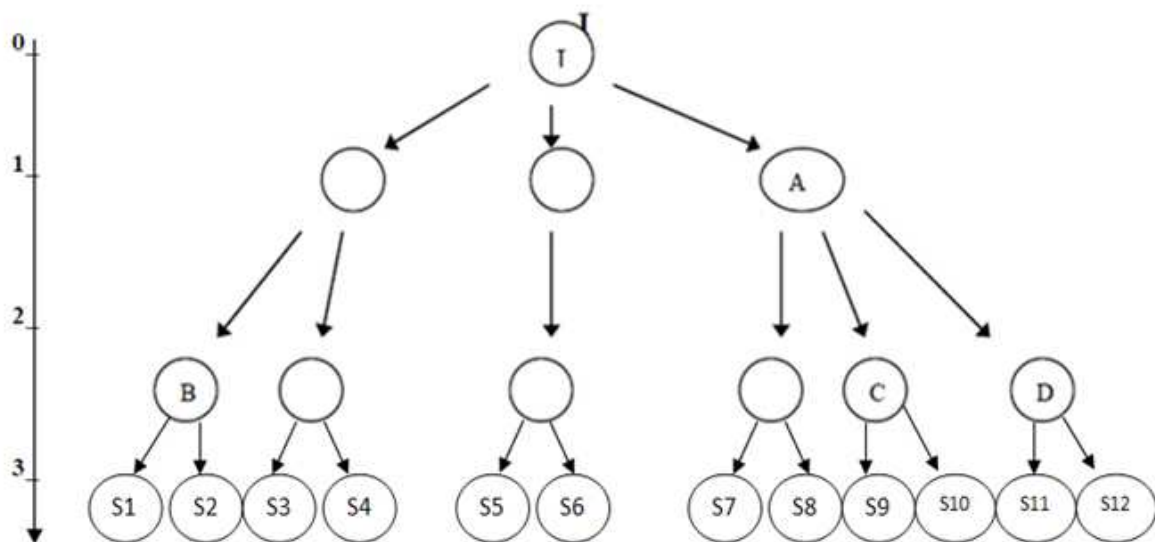


Figura 1: Struttura ad albero

La struttura ad albero è particolarmente interessante perchè garantisce una adeguata flessibilità nel modellare le dipendenze intertemporali tra le variabili casuali. La struttura e l'estensione dell'albero determinano una distribuzione di probabilità discreta che si assume, ai fini della risoluzione del problema, rappresentativa di tutte le possibili realizzazioni future.

Sia  $\mathfrak{N}$  l'insieme dei nodi dell'albero. Si consideri un generico nodo foglia  $i$  a livello  $T$  e siano  $i_1, i_2, \dots, i_T$  i nodi facenti parte del cammino dalla radice al nodo considerato. La probabilità associata allo scenario è definita dal seguente prodotto:

$$p_{i_1 i_2} * p_{i_2 i_3} * \dots * p_{i_{(T-1)} i_T}$$

dove la generica  $p_{ai}$  rappresenta la probabilità di spostarsi dal nodo  $a$  (nodo predecessore di  $i$ ) al nodo  $i$ .

Si consideri:

- un generico nodo  $n \in \mathfrak{N}$ ;
- $p_n$  la probabilità associata al nodo  $n$ ;
- $x_n$  il corrispondente vettore delle variabili di decisione;
- $c_n, h_n, T_n, W_n$  i corrispondenti vettori e matrici già precedentemente indicate.

Il modello di programmazione stocastica multistadio può essere formulato come segue:

$$\min \sum_{n \in \mathfrak{N}} p_n c_n x_n \quad (7)$$

s.v.

$$Ax_1 = b$$

$$T_n x_{pre(n)} + W_n x_n = h_n, \quad n \in \mathfrak{N}$$

$$x_1 \geq 0$$

$$x_n \geq 0, \quad n \in \mathfrak{N}$$

dove  $A$  è la matrice associata alle variabili di primo stadio  $x_1$ .

Si indichi con :

- $S_n$  l'insieme di scenari che passano attraverso il nodo  $n$ ;
- $\{y_{tl}\}_{t=1}^T$  la sequenza di decisioni associate allo scenario  $l$  e  $t$  indica un generico stadio del problema decisionale;
- $t(n)$  stadio nel nodo  $n$ .

I vincoli di non anticipatività possono essere rappresentati come:

$$x_n - y_{t(n)l} = 0, \quad \forall l \in S_n$$

Uno dei vantaggi derivanti dall'inclusione esplicita dei vincoli deriva dal fatto che ogni scenario può essere trattato in maniera indipendente. Per ogni scenario  $l = 1, \dots, N$ , si indichi con  $(c_{tl}, h_{tl}, T_{tl}, W_{tl})$  i corrispondenti vettori e matrici e sia  $p_l$  la probabilità associata. Il modello multistadio può essere riformulato come:

$$\min \sum_{l=1}^N p_l [ c_1 y_{1l} + \sum_{t=2}^T c_{tl} y_{tl} ]$$

s.v.

$$A y_{1l} = b, \quad l = 1, \dots, N$$

$$T_{(t-1)l} y_{(t-1)l} + W_{tl} y_{tl} = h_{tl}, \quad t = 2, \dots, T, \quad l = 1, \dots, N$$

$$x_n - y_{t(n)l} = 0, \quad \forall l \in S(n), \quad \forall n \in \mathfrak{N}$$

$$y_{tl} \geq 0, \quad t = 1, \dots, T, \quad l = 1, \dots, N$$

### 2.2.3. L'approccio a scenari

L'analisi a scenari è ampiamente applicata in tutti quegli ambiti decisionali in cui sia necessario assumere decisioni in condizioni di incertezza e non si disponga di previsioni puntuali attendibili.

Gli scopi dell'analisi a scenario possono essere molteplici, tra i principali ricordiamo:

- sviluppare un insieme di previsioni sul comportamento di un sistema in modo strumentale al processo decisionale;
- consentire la valutazione e il confronto tra diverse strategie rispetto a diversi sviluppi futuri del sistema.

Non esiste in letteratura una definizione univoca di scenario. In generale uno scenario può essere visto come una realizzazione, sull'orizzonte temporale considerato, del processo stocastico che descrive il comportamento delle grandezze aleatorie. Tale processo può essere a tempo discreto oppure a tempo continuo con monitoraggio discreto, dove le epoche di monitoraggio possono corrispondere, ad esempio, agli istanti di decisione.

Nella rappresentazione mediante albero dell'insieme degli scenari ciascun nodo rappresenta uno stato del sistema in corrispondenza di un istante temporale. Ogni scenario è rappresentato da un sentiero che collega l'origine dell'albero (nodo radice) con un nodo all'istante finale (nodo foglia), che rappresenta un possibile stato del sistema alla fine dell'orizzonte temporale considerato.

La struttura ad albero permette di evidenziare il legame di ciascuna realizzazione con le realizzazioni che la precedono e che la seguono. In particolare ad ogni nodo dell'albero sono associati un nodo padre, o generatore, e un numero, che può essere variabile, di nodi discendenti, o figli.

La generazione degli scenari è un passo cruciale per la corretta specificazione del problema di ottimizzazione stocastica e riveste un ruolo fondamentale nella attendibilità delle soluzioni ottenute. La struttura dei possibili sviluppi futuri del sistema (albero degli scenari) può essere determinata in diversi modi in relazione anche alla natura e alla struttura del problema considerato.

### 2.2.3.1. Classificazione degli scenari

Gli scenari possono essere classificati sulla base delle ipotesi e delle finalità per le quali sono costruiti. Di seguito sono riportate alcune tra le più importanti classificazioni presentate in letteratura.

- *Scenari esplorativi*: l'analisi parte dalla situazione attuale, si cercano di individuare i possibili sviluppi futuri del sistema prevedendo le conseguenze legate alla situazione presente. In questo caso sono importanti le relazioni di causa-effetto; la direzione ideale secondo cui è condotta l'analisi è quella *forward*.
- *Scenari anticipativi*: sono costruiti cercando le possibili cause che possono portare ad una certa situazione futura, l'analisi si focalizza sugli obiettivi da raggiungere. La direzione seguita idealmente è quella *backward*.

Un'altra possibile classificazione riguarda l'ampiezza degli sviluppi considerati dagli scenari. In particolare si può distinguere tra:

- *Scenari a trend* che estendono l'attuale andamento del sistema nel futuro includendo pochissimi o nessun elemento inatteso.
- *Scenari periferici* che assegnano valori estremi alle variabili creando evoluzioni inattese del sistema.

Infine una ulteriore distinzione può essere fatta sulla base del tipo di analisi condotta, in particolare si può distinguere tra approccio *qualitativo* e *quantitativo*.

Nel primo caso il processo di generazione degli scenari è organizzato in più passi secondo uno schema generale di questo tipo:

1. Identificazione degli obiettivi.
2. Individuazione dell'orizzonte temporale.
3. Individuazione dei fattori chiave.
4. Costruzione degli scenari combinando le diverse possibili realizzazioni dei fattori chiave.

In queste tecniche rivestono notevole importanza le relazioni causa-effetto e gli elementi di anticipazione che già sono presenti al momento della generazione degli scenari.

Per quanto riguarda l'approccio quantitativo si possono individuare diversi tipi di analisi:

- *Trend-impact analysis*
- *Forecast-adjusted model*
- *Cross-impact analysis*

L'analisi di trend si pone come ponte tra la generazione di scenari e la previsione, creando degli sviluppi futuri per il sistema che non si discostano in misura rilevante dal percorso previsto.

I modelli di previsione corretta, nella loro versione più semplice, rappresentano una combinazione di previsione statistica e di 'correzioni' inserite per tenere conto di particolari eventi e fattori che possono avere un impatto sul trend.

In una analisi di trend-impact i fattori che hanno un impatto graduale e continuo possono essere meglio modellati con metodi statistici, mentre quei fattori che hanno un effetto improvviso e discontinuo (cambi strutturali) sono meglio descritti da aggiustamenti del trend.

Le tecniche di cross-impact analysis, invece, focalizzano l'attenzione sulle interdipendenze tra le variabili considerate nel determinare gli sviluppi futuri del sistema, in particolare le interazioni che diversi fattori possono produrre sulle variabili considerate. Ciò implica la necessità di individuare un insieme di probabilità condizionate che indichino come ciascun evento influenzi le realizzazioni degli altri eventi.

Il tipo di processo analizzato e le finalità per cui gli scenari vengono generati determinano quali tra le caratteristiche presentate saranno dominanti.

Sulla base di tale schema generale, utilizzando adeguati metodi di generazione sulla base del campione passato, combinati con tecniche di aggiornamento delle probabilità, è possibile ottenere sofisticati e flessibili modelli per la generazione degli scenari. Il processo di generazione degli scenari comprende dunque un processo di analisi del sistema, l'individuazione delle relazioni più significative che

determinano il comportamento delle grandezze coinvolte e lo studio delle possibili dinamiche di evoluzione futura.

### **2.2.3.2. Analisi e valutazione degli scenari**

Il processo di generazione e valutazione degli scenari non può prescindere dalle finalità per cui tali scenari sono generati e non può essere basato esclusivamente sulla accuratezza di previsione. Bunn e Salo (1993) hanno individuato alcune proprietà che l'insieme di scenari generati dovrebbe rispettare.

Tali criteri, di natura generale e rilevanti nella fase di valutazione degli scenari, sono:

- Ammissibilità: gli scenari generati devono essere plausibili in riferimento allo stato attuale e passato del sistema che si descrive.
- Ampiezza : riguarda il numero degli scenari generati e l'ampiezza dell'analisi effettuata.
- Consistenza : gli scenari generati devono essere coerenti con le ipotesi poste sulle grandezze prese in esame, sulle loro dinamiche e sulle loro relazioni causali. Le realizzazioni di diverse variabili non devono essere in conflitto tra loro e con il passato, i parametri generati non devono violare il modello teorico alla base del sistema che si sta analizzando. La consistenza è una caratteristica fondamentale perché fornisce la chiave per l'interpretazione dei risultati ottenuti.

Questi criteri guida di carattere qualitativo devono essere tradotti in ipotesi da porre sul processo di generazione e sulle variabili in esso coinvolte.

Un secondo aspetto del processo di valutazione degli scenari riguarda l'applicazione di criteri quantitativi che permettano di ottenere delle misure aggregate dell'informazione contenuta negli scenari generati. Tale analisi permette di confrontare scenari generati con differenti metodi e condurre una analisi di sensibilità delle soluzioni ottime di un problema stocastico rispetto a differenti insiemi di scenari.

#### 2.2.4. Modelli con vincoli probabilistici

L'introduzione dell'incertezza nei modelli matematici porta alla possibile violazione dei vincoli a causa di effetti casuali inaspettati. Nei modelli ricorsivi ciò viene risolto attraverso decisioni compensative da prendere nel secondo stadio. In altre circostanze è più auspicabile accettare la violazione dei vincoli facendo sì che la probabilità dell'evento che porta alla violazione sia vincolata a rimanere inferiore rispetto ad un determinato valore di riferimento. In termini di modellazione matematica vengono considerati vincoli probabilistici nella formulazione del problema. Un modello lineare sotto vincoli probabilistici è definito come segue:

$$\begin{aligned} & \min c^T x && (8) \\ \text{s.v.} & P[T(\omega)x \geq h(\omega)] \geq p \\ & x \in X_1 \end{aligned}$$

dove:

- $p \in (0,1)$  rappresenta il valore di affidabilità con il quale si richiede che il sistema funzioni.  
 $p$  viene deciso in base allo specifico problema considerato.

I vincoli probabilistici possono essere di due tipi:

- vincoli probabilistici congiunti;
- vincoli probabilistici singoli.

In particolare, il vincolo del problema (8) è detto vincolo probabilistico congiunto in quanto imposto su tutte le disequazioni del sistema di vincoli stocastici  $T(\omega)x \geq h(\omega)$ . Si parla di vincoli probabilistici singoli di nel caso in cui i vincoli stocastici sono definiti sulle singole disequazioni  $P[T_i(\omega)x \geq h_i(\omega)] \geq p_i$ ,  $i = 1, \dots, m$ ,  $p_i \in [0,1]$  dove  $T_i(\omega)$  rappresenta la  $i$ -esima riga della matrice  $T(\omega)$  e  $h_i(\omega)$  la  $i$ -esima componente del vettore  $h(\omega)$ .

In realtà l'impiego di vincoli singoli o congiunti dipende fortemente dal problema trattato. In particolare l'utilizzo di vincoli singoli deve essere giustificato in quanto potrebbero portare ad una distorsione del problema da modellare



### 3. Il DARP in condizioni di incertezza

In questo capitolo verrà descritta una versione stocastica del DARP in cui si suppone che alcune domande siano note solo in probabilità fino ad un certo istante a partire dal quale le rotte, precedentemente pianificate, dovranno essere opportunamente modificate per includere eventualmente dei clienti aggiuntivi.

Il modello si basa sul framework generale della programmazione stocastica, ed in particolare può essere inquadrato come un modello a due stadi con ricorso. Nel primo stadio si costruiscono delle rotte “robuste” cioè in grado di assorbire possibili variazioni future; tali rotte sono poi modificate attraverso una azione di ricorso nel secondo stadio, una volta note con certezza le domande aggiuntive.

Per poter formalizzare in modo dettagliato il modello, si presenta nella prima parte del capitolo la formulazione e la definizione del problema deterministico di partenza, rimandando alla seconda parte del capitolo la descrizione della variante stocastica.

#### 3.1. Definizione del problema DARP

Il DARP deterministico con finestre temporali e una flotta omogenea di veicoli può essere formulato utilizzando i parametri e le variabili di seguito considerate.

Sia  $G = (V, A)$  un grafo completo, dove  $V = \{v_0, v_1, \dots, v_{2n}\}$  è l'insieme dei vertici e  $A = \{(v_i, v_j): v_i, v_j \in V, i \neq j\}$  è l'insieme di archi. Il vertice  $v_0$  rappresenta il deposito che è la base di una flotta di  $m$  veicoli, ed i rimanenti  $2n$  vertici rappresentano origine e destinazione per le richieste di trasporto. Ogni coppia di vertici  $(v_i, v_{i+n})$  rappresenta una richiesta di trasporto dall'origine  $v_i$  alla destinazione  $v_{i+n}$ . Ad ogni vertice  $v_i \in V$  è associato un carico  $q_i$  (con  $q_0 = 0$ ), una durata di servizio non negativa  $d_i$  (con  $d_0 = 0$ ) e una finestra temporale  $[e_i, l_i]$  dove  $e_i$  ed  $l_i$  sono non negative. Il carico è pari a 1 per i vertici  $v_i, \dots, v_{i+n}$  e a -1 per i vertici  $v_{n+1}, \dots, v_{2n}, \dots$ . La durata di servizio corrisponde al tempo necessario per consentire all'utente di salire o scendere dal veicolo. Il

termine  $T$  denota la fine dell'orizzonte di pianificazione. Ogni arco  $(v_i, v_j)$  ha un costo di routing  $c_{ij}$  non negativo associato e un tempo di viaggio  $t_{ij}$  non negativo. Infine,  $L$  è il massimo tempo di corsa di un utente.

Il DARP consiste nel progettare  $m$  rotte dei veicoli su  $G$  tali che:

1. ogni rotta inizia e termina presso il deposito;
2. per ogni richiesta  $i$ , i vertici  $v_i$  e  $v_{i+n}$  appartengono alla stessa rotta e il vertice  $v_{i+n}$  è visitato dopo il vertice  $v_i$ ;
3. il carico del veicolo  $k$  non supera mai il limite  $Q_k$  preimpostato;
4. la durata totale della rotta  $k$  non supera il limite  $T_k$  preimpostato;
5. il servizio al vertice  $v_i$  inizia nell'intervallo  $[e_i, l_i]$  e ogni veicolo lascia il deposito e torna al deposito nell'intervallo  $[e_0, l_0]$ ;
6. il tempo di corsa di ogni utente non supera  $L$ ;
7. è minimizzato il costo totale di routing di tutti i veicoli.

$A_i$  rappresenta il tempo di arrivo di un veicolo al vertice  $v_i$ , con  $B_i \geq \max\{e_i, A_i\}$ . l'inizio del servizio al vertice  $v_i$ , e con  $D_i = B_i + d_i$  il tempo di partenza dal vertice  $v_i$ . Si assume che l'attesa su ogni vertice  $v_i$  sia permessa prima dell'avvio del servizio ma non sia consentita dopo che il servizio è terminato. Il vincolo sulla finestra temporale al vertice  $v_i$  è violato se  $B_i > l_i$ .

L'arrivo prima  $e_i$  è, tuttavia, permesso e il veicolo subisce poi un tempo di attesa  $W_i = B_i - A_i$ . Il tempo di corsa associato alla richiesta  $i$  è calcolata come  $L_i = B_{i+n} - D_i$ . Il tempo di corsa corrisponde quindi al tempo trascorso tra la fine del servizio al vertice  $v_i$  e l'inizio del servizio al vertice  $v_{i+n}$ . Se non ci fossero i vincoli sul tempo di corsa, sarebbe sempre meglio impostare  $B_i = \max\{e_i, A_i\}$ . Tuttavia, a

volte, può essere proficuo ritardare l'inizio del servizio al vertice  $v_i$ , in modo da ridurre il tempo di attesa non necessario al vertice  $v_{i+n}$  (o su ogni altro vertice visitato tra il  $v_i$  e  $v_{i+n}$ ) e, quindi, il tempo di corsa associato alla richiesta  $i$ .

### 3.2. Formulazione deterministica del problema DARP

Formalmente il DARP statico può essere matematicamente formulato come segue:

$$\text{Min } \sum_{k \in K} \sum_{i \in P} \sum_{j \in V} c_{ij}^k x_{ij}^k + K * \sum_{k \in K} (1 - x_{00dest}^k) \quad (1)$$

$$\sum_{k \in K, j \in V} x_{ij}^k = 1 \quad i \in P \quad (2)$$

$$\sum_{i \in V} x_{0i}^k = \sum_{i \in V} x_{i0dest}^k = 1 \quad k \in K \quad (3)$$

$$\sum_{j \in V} x_{ij}^k - \sum_{j \in V} x_{(n+i)j}^k = 0 \quad i \in P, k \in K \quad (4)$$

$$\sum_{j \in V} x_{ji}^k - \sum_{j \in V} x_{ij}^k = 0 \quad i \in P \cup D, k \in K \quad (5)$$

$$u_j^k \geq (u_i^k + d_i + t_{ij})x_{ij}^k \quad i, j \in V, k \in K \quad (6)$$

$$w_j^k \geq (w_i^k + q_j)x_{ij}^k \quad i, j \in V, k \in K \quad (7)$$

$$r_i^k \geq u_{n+i}^k - (u_i^k + d_j) \quad i \in P, k \in K \quad (8)$$

$$u_{0dest}^k - u_0^k \leq T_k \quad k \in K \quad (9)$$

$$e_i \leq u_i^k \leq l_i \quad i \in V, k \in K \quad (10)$$

$$t_{i,n+i} \leq r_i^k \leq L \quad i \in P, k \in K \quad (11)$$

$$\max\{0, q_i\} \leq w_i^k \leq \min\{Q_k, Q_k + q_i\} \quad i \in V, k \in K \quad (12)$$

$$x_{ij}^k = \{0,1\} \quad i, j \in V, k \in K \quad (13)$$

In questa formulazione, i vincoli (2) e (4) assicurano che ogni richiesta venga servita una volta da uno stesso veicolo, mentre i vincoli (3) e (5) garantiscono che ogni veicolo

inizi e termini la sua rotta presso il deposito. Vincoli (6) a (8) definiscono l'inizio dei tempi di servizio, il carico di veicoli e i tempi di corsa degli utenti, rispettivamente, mentre i vincoli (9) a (12) garantiscono che questi saranno ammissibili. L'obiettivo (1) è quello di minimizzare il numero dei veicoli utilizzati, più il totale dei costi di routing. I vincoli non lineari (6) e (7) possono essere linearizzati con l'introduzione delle costanti  $M_{ij}^k$  e  $W_{ij}^k$  come segue:

$$u_j^k \geq u_i^k + d_i + t_{ij} - M_{ij}^k(1 - x_{ij}^k) \quad i, j \in V, k \in K \quad (14)$$

$$w_j^k \geq w_i^k + q_j + t_{ij} - W_{ij}^k(1 - x_{ij}^k) \quad i, j \in V, k \in K \quad (15)$$

### 3.3. Definizione del DARP stocastico (Stochastic DARP) a due stadi

Per catturare gli elementi stocastici del problema, si estende il modello deterministico ad un modello stocastico a due stadi. In tale formulazione si assume che all'inizio della giornata lavorativa molte richieste siano già note, e che durante la giornata lavorativa si presentino ulteriori richieste, denominate richieste "stocastiche", la cui distribuzione di probabilità è nota già all'inizio della giornata. L'incertezza legata a tali richieste aleatorie si rende nota in un dato momento della giornata. Questa assunzione permette di usare un modello di programmazione stocastica a due stadi per modellare l'incertezza sulle richieste tardive in cui nel primo stadio la rotta "base" viene decisa tenendo conto però del costo atteso di modifica della rotta stessa in presenza di richieste aleatorie. In un certo momento  $T$ , nell'arco della giornata, tutte le informazioni precedentemente sconosciute sono rivelate. Cioè, tutte le richieste stocastiche, la loro posizione, e le finestre temporali diventano note al tempo  $T$ . A partire da questo momento le azioni di ricorso possono essere considerate, in modo che i nuovi clienti possano essere inseriti nelle rotte dei veicoli.

Si introduce di seguito un semplice esempio esemplificativo.

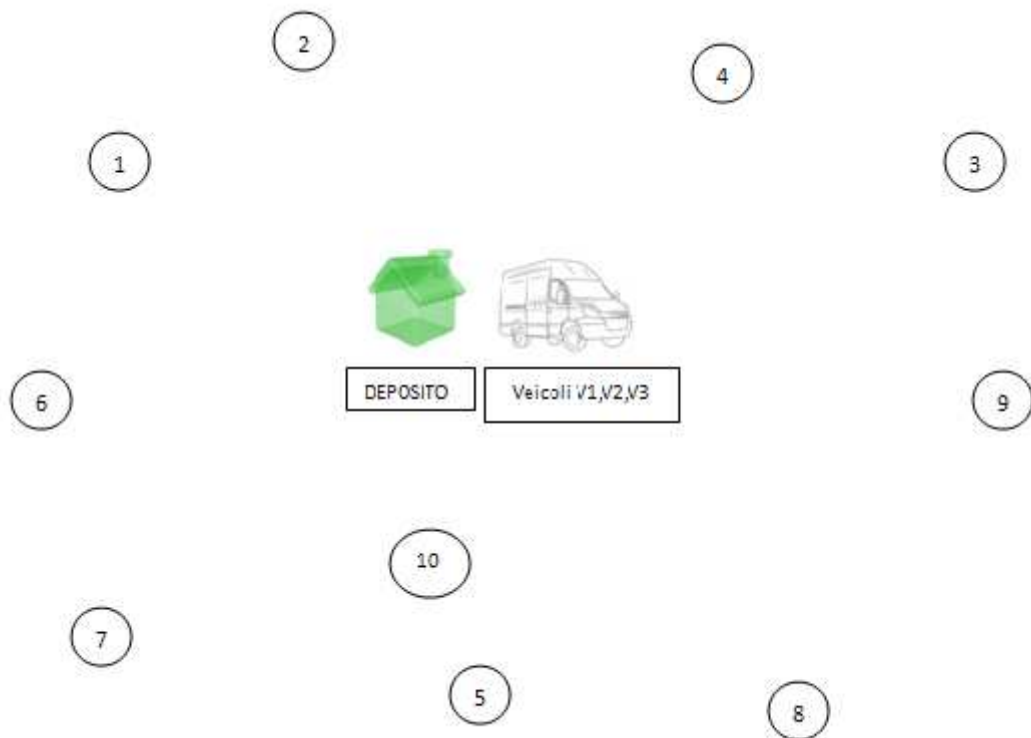


Figura 2 (a): Pianificazione rotte: Informazione sulla domanda al tempo iniziale  $T_0$

Si assume di avere 10 nodi come mostrato in Figura 2 (a) con cinque richieste: i vertici "1, 2, 3, 4, 5" sono nodi di pickup, mentre i nodi "6, 7, 8, 9, 10" sono di delivery. In particolare, la merce caricata al nodo 1 va consegnata al nodo 6, il carico del nodo 2 va al nodo 7 e così via. Inoltre si hanno a disposizione tre veicoli che si trovano nel deposito all'inizio del periodo di pianificazione.

Si suppone che durante la giornata possano arrivare nuove chiamate. In particolare, sono previsti due scenari ad ognuno dei quali è associata una specifica probabilità di accadimento. Ogni scenario è caratterizzato dalle seguenti nuove richieste (Figura 2 (b)):

- lo scenario 1 a cui è associata una probabilità di accadimento del 30% è costituito da una richiesta caratterizzata dai nodi  $a$  e  $b$ . e La merce caricata al vertice  $a$  va consegnata al vertice  $b$ ;

- lo scenario 2 a cui è associata una probabilità del 70% è costituito dai nodi  $c, d, e, f$ . Anche in questo caso, la merce caricata al vertice  $c$  va consegnata al vertice  $e$ , il vertice  $c$  di pickup ha come corrispondente vertice di delivery  $e$  ed il vertice  $d$  il vertice  $f$ .

La rotta base dovrà essere pianificata all'istante  $T_0$  in condizioni di incertezza in modo da tener conto del possibile costo (atteso) derivante dal soddisfacimento delle richieste aleatorie.

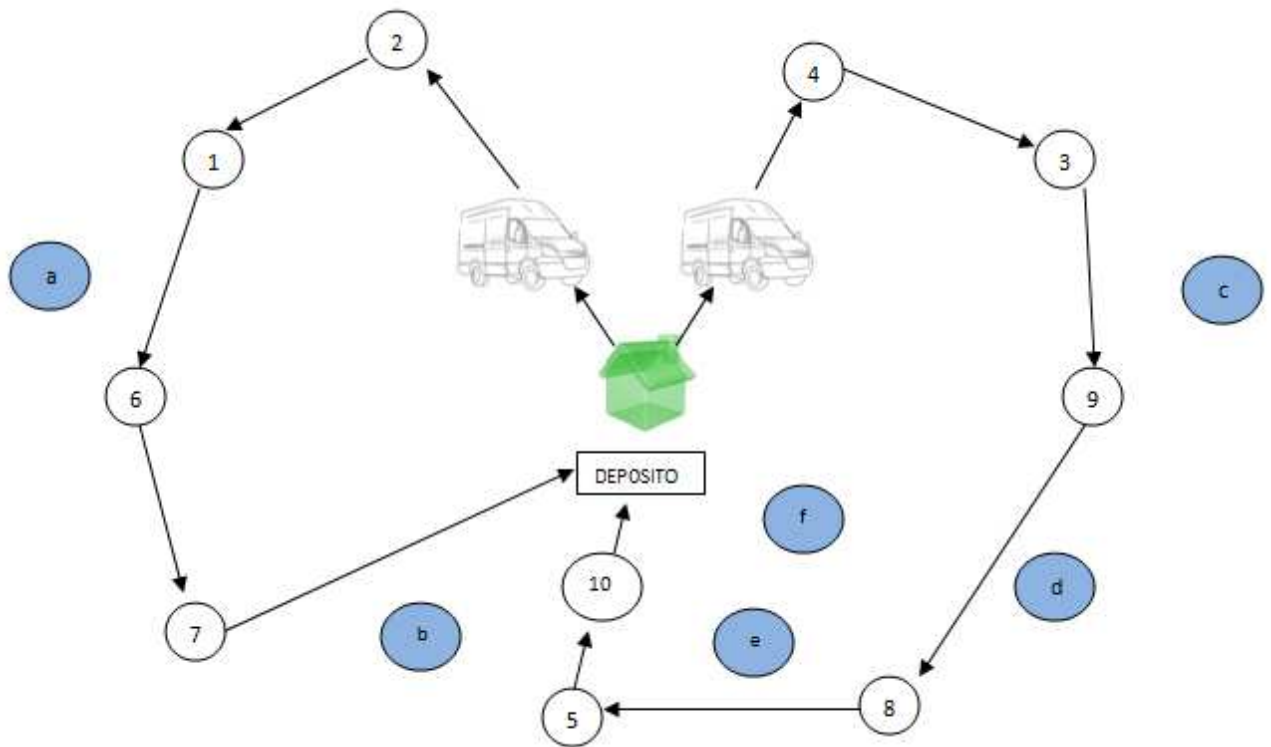


Figura 2 (b): Pianificazione rotte: Prima pianificazione delle rotte in condizioni di incertezza

Tale pianificazione Figura 2 (b) prevede l'utilizzo di due veicoli: il primo veicolo sarà utilizzato per la rotta  $Deposito \rightarrow 2 \rightarrow 1 \rightarrow 6 \rightarrow 7 \rightarrow Deposito$  e il secondo veicolo per la rotta  $Deposito \rightarrow 4 \rightarrow 3 \rightarrow 9 \rightarrow 8 \rightarrow 5 \rightarrow 10 \rightarrow Deposito$ .

Supponiamo ora di osservare il sistema nell'istante  $T$  (Figura 2(b)) che è il momento della giornata in cui le richieste (fino a tale momento note solo in probabilità) si palesano effettivamente secondo uno solo degli scenari possibili.

Le rotte pianificate al tempo  $T_0$  cambieranno per tener conto delle nuove informazioni (certe) sulla domanda dei nuovi clienti. E' evidente che il tratto di rotta già percorso non potrà essere modificato.

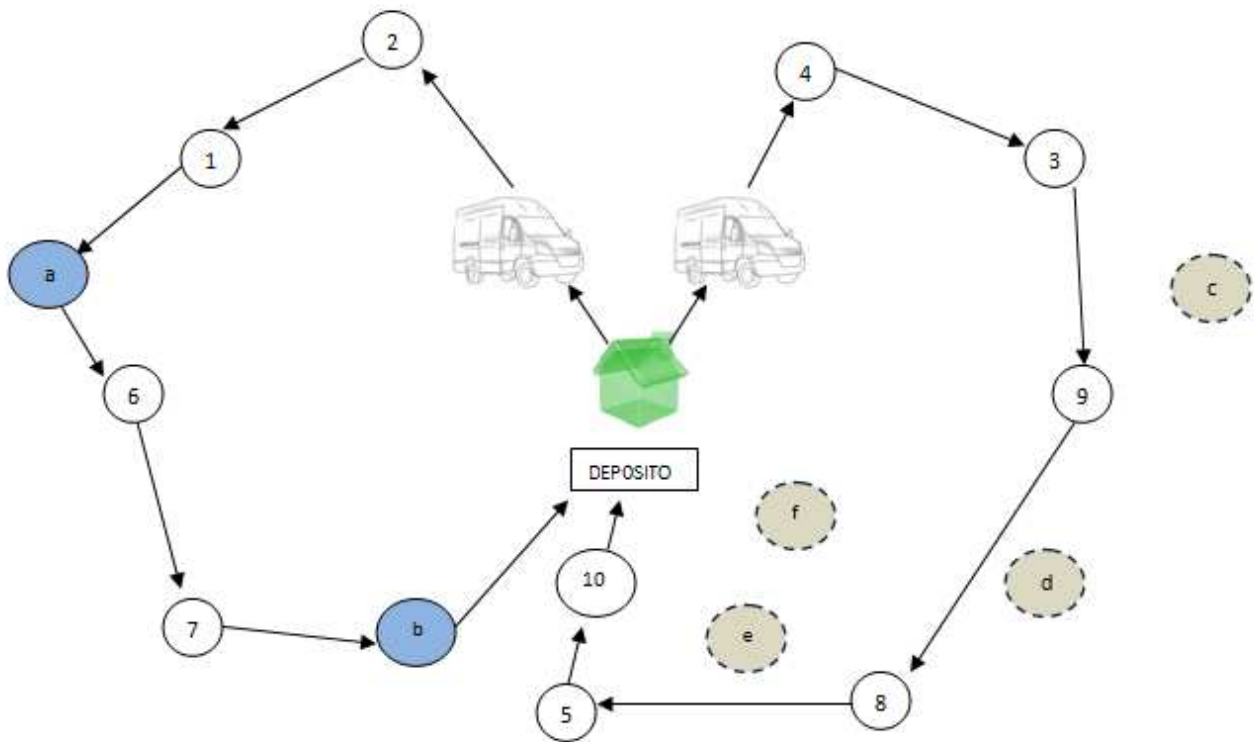


Figura 2 (c): Pianificazione rotte: Ripianificazione delle rotte al tempo  $T$  considerando lo scenario 1

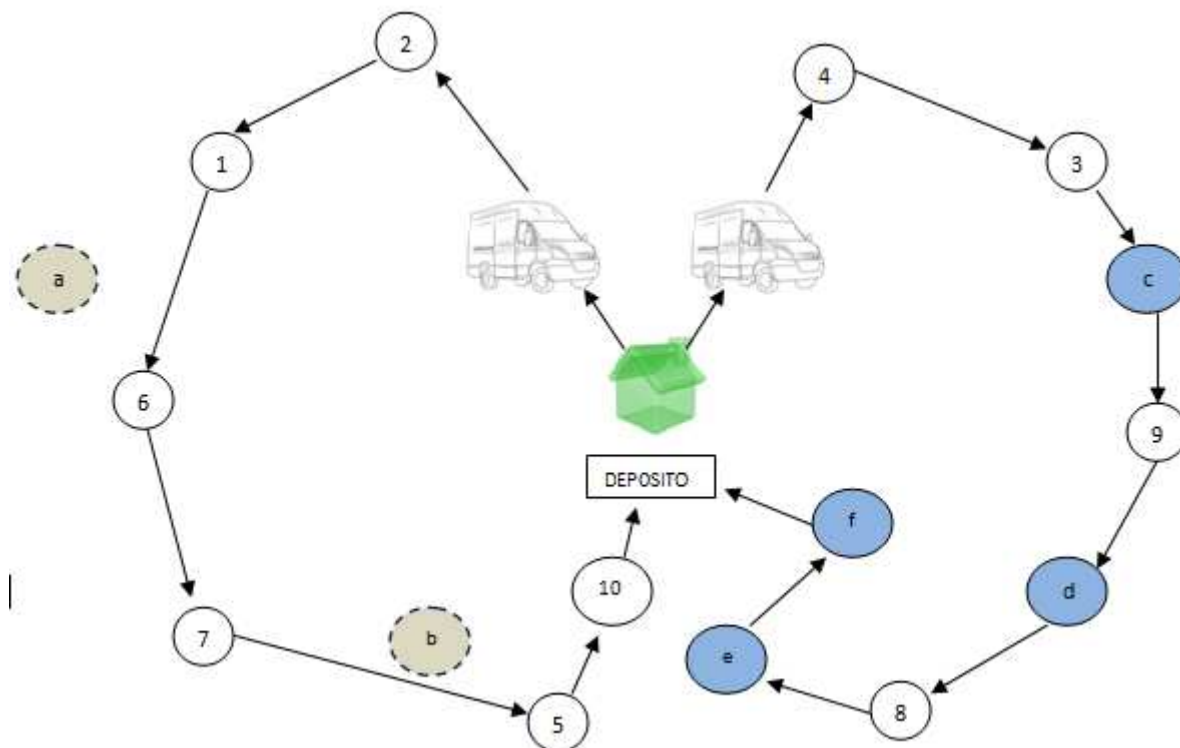


Figura 2 (d): Pianificazione rotte: Ripianificazione delle rotte al tempo  $T$  considerando lo scenario 2; notare che le soluzioni di secondo stadio non possono richiedere la modifica delle tratte delle rotte già servite al tempo  $T$

Le figure 2 (c) e 2 (d) mostrano il risultato dell'aggiustamento delle nuove rotte considerando in un caso che il verificarsi dello scenario 1 e nell'altro il verificarsi dello scenario 2.

Il verificarsi dello scenario 1 (Figura 2 (c)) porta alla ripianificazione delle rotte indicate di seguito:

- per il veicolo 1:  $(v1) \rightarrow 2 \rightarrow 1 \rightarrow a \rightarrow 6 \rightarrow 7 \rightarrow b \rightarrow Deposito$ ;
- per il veicolo 2:  $(v2 \text{ si trova in } 4) \rightarrow 3 \rightarrow 9 \rightarrow 8 \rightarrow 5 \rightarrow 10 \rightarrow Deposito$ .

Il verificarsi dello scenario 2 (Figura 2 (d)) comporta che nella prima rotta, le domande ai nodi 5 e 10 che erano servite dal veicolo 1, ora sono servite dal veicolo 2 nella nuova pianificazione delle rotte che sono il risultato dell'ripianificazione al tempo  $T_n$ . Le nuove rotte sono le seguenti:

- per il veicolo 1:  $(v1) \rightarrow 2 \rightarrow 1 \rightarrow 6 \rightarrow 7 \rightarrow 5 \rightarrow 10 \rightarrow Deposito$ ;
- per il veicolo 2:  $(v2 \text{ si trova in } 4) \rightarrow 3 \rightarrow c \rightarrow 9 \rightarrow d \rightarrow 8 \rightarrow e \rightarrow f \rightarrow Deposito$ .

Nell'esempio considerato le rotte sono ammissibili in quanto soddisfano i vincoli del problema.



Formalmente, per introdurre la formulazione matematica del modello DARP stocastico, introduciamo la seguente notazione. Sia  $P^+ = \{2n + 1, \dots, M\}$  l'insieme dei vertici di pick up rilevati al tempo  $T$  e  $D^+$  i vertici di delivery associati. L'insieme di tutti i nodi diviene  $V^+ = V \cup P^+ \cup D^+$ . Uno dei possibili modi per rappresentare l'incertezza associata alla comparsa di una serie di richieste è la definizione di un set di scenari futuri; ogni scenario corrisponde ad un set di richieste che si manifesteranno dopo il tempo  $T$ ; ad ogni scenario è inoltre associata una probabilità  $p_s$  di realizzazione.

Si definiscono inoltre le finestre temporali  $\forall i \in P^+ [e_i, l_i]$ ,  $d_i$  la durata di servizio e  $q_{is}$  il carico per il nodo  $i$  in ogni scenario  $S$ , rispettivamente. La variabile  $x_{ijs}^{k+}$  è uguale a 1 se e solo se l'arco  $(i, j)$  è attraversato dal veicolo  $k \in K$  nel secondo stadio, sotto lo scenario  $S$  ma non nella soluzione di primo stadio; la variabile  $x_{ijs}^{k-}$  è uguale a 1 se e solo se l'arco  $(i, j)$  è attraversato dal veicolo  $k \in K$  nel primo stadio, ma non nella soluzione di ricorso nello scenario  $S$ . Inoltre, sia  $u_{is}^{k+}$  il momento in cui il veicolo  $k$  inizia il servizio vertice  $i \in P^+$ ,  $w_{is}^{k+}$  il carico del il veicolo  $k$  dopo aver lasciato il vertice  $i$ , e  $r_{is}^{k+}$  il tempo di corsa dell'utente  $i$  nello scenario  $S$ .

### 3.4. Formulazione stocastica del DARP

Il DARP stocastico è sottoposto a tutti i vincoli del DARP deterministico, più alcuni vincoli aggiuntivi sotto riportati. Da notare che nel DARP stocastico vengono introdotte le variabili  $x_{ijs}^{k+}$  ed  $x_{ijs}^{k-}$  che rappresentano, rispettivamente, gli archi da aggiungere o da eliminare ad ogni rotta  $k$  nello scenario  $s$ . Tali variabili sono quindi le variabili di secondo stadio; ovvero la soluzione del problema prevede un set di  $x_{ijs}^{k+}$ ,  $x_{ijs}^{k-}$  per ogni scenario, ed una volta realizzata l'incertezza solo il set di variabili corrispondenti allo scenario realizzatosi influiranno sulla soluzione da adottare.

Da notare, infine, che l'obiettivo del DARP stocastico è la minimizzazione del costo atteso, dove per costo atteso si intende il costo della soluzione di primo stadio sommato al costo delle modifiche da effettuare in ogni scenario pesato per la rispettiva probabilità.

Di seguito la formulazione matematica del DARP stocastico:

$$\text{Min } \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} c_{ij}^k x_{ij}^k + K * \sum_{k \in K} (1 - x_{00dest}^k) + \sum_{s \in S} p_s [\sum_{i,j \in V^+} c_{ij}^k (x_{ijs}^{k+} - x_{ijs}^{k-}) + K * \sum_{k \in K} x_{00dest}^k - x_{00dest}^k]$$

(16)

Soggetto a (2) – (15)

$$\sum_{k \in K, j \in V^+} (x_{ijs}^{k-} + x_{ij}^k - x_{ijs}^{k-}) = 1 \quad i \in P, s \in S \quad (17)$$

$$\sum_{k \in K, j \in V^+} (x_{ijs}^{k+} + x_{ij}^k - x_{ijs}^{k-}) = 1 \quad i \in P^+, s \in S \quad (18)$$

$$\sum_{i \in V^+} (x_{i0dest}^k + x_{i0dest}^{k+} - x_{i0dest}^{k-}) = \sum_{i \in V^+} (x_{0i}^k + x_{0is}^{k+} - x_{0is}^{k-}) = 1 \quad k \in K, s \in S \quad (19)$$

$$\sum_{j \in V^+} (x_{ij}^k + x_{ijs}^{k+} - x_{ijs}^{k-}) - \sum_{j \in V^+} (x_{(n+i)j}^k + x_{(n+i)js}^{k+} - x_{(n+i)js}^{k-}) = 0 \quad i \in V, k \in K, s \in S \quad (20)$$

$$\sum_{j \in V^+} (x_{ij}^k + x_{ijs}^{k+} - x_{ijs}^{k-}) - \sum_{j \in V^+} (x_{(M+i)j}^k + x_{(M+i)js}^{k+} - x_{(M+i)js}^{k-}) = 0 \quad i \in P^+, k \in K, s \in S \quad (21)$$

$$\sum_{j \in V^+} (x_{ji}^k + x_{jis}^{k+} - x_{jis}^{k-}) - \sum_{j \in V^+} (x_{ij}^k + x_{ijs}^{k+} - x_{ijs}^{k-}) = 0 \quad i \in V^+ \setminus \{0\} \setminus \{0_{dest}\}, k \in K, s \in S \quad (22)$$

$$u_{js}^{k+} \geq u_{is}^{k+} + d_i + t_{ij} - M_{ijs}^k (1 - x_{ij}^k - x_{ijs}^{k+} + x_{ijs}^{k-}) \quad i, j \in V^+, k \in K, s \in S \quad (23)$$

$$w_{js}^{k+} \geq (w_{is}^{k+} + q_{js}) - W_{ijs}^k (1 - x_{ij}^k - x_{ijs}^{k+} + x_{ijs}^{k-}) \quad i, j \in V^+, k \in K, s \in S \quad (24)$$

$$r_{is}^{k+} = u_{(n+i)s}^{k+} - (u_{is}^{k+} + d_i) \quad i \in P, k \in K, s \in S \quad (25)$$

$$r_{is}^{k+} = u_{(M+i)s}^{k+} - (u_{is}^{k+} + d_i) \quad i \in P^+, k \in K, s \in S \quad (26)$$

$$u_{0dest}^{k+} - u_{0s}^{k+} \leq T_k \quad k \in K, s \in S \quad (27)$$

$$e_i \leq u_{is}^{k+} \leq l_i \quad i \in V^+, k \in K, s \in S \quad (28)$$

$$t_{i,n+i} \leq r_{is}^{k+} \leq L \quad i \in P, k \in K, s \in S \quad (29)$$

$$t_{i,M+i} \leq r_{is}^{k+} \leq L \quad i \in P^+, k \in K, s \in S \quad (30)$$

$$x_{ijs}^{k-} \leq x_{ij}^k \quad i \in P, j \in P, s \in S \quad (31)$$

$$t * x_{ijs}^{k-} \leq u_{is}^{k+} + d_i \quad i \in P, s \in S \quad (32)$$

$$\max\{0, q_i\} \leq w_{is}^{k+} \leq \min\{Q_k, Q_k + q_i\} \quad i \in V, k \in K, s \in S \quad (33)$$

$$\max\{0, q_{is}\} \leq w_{is}^{k+} \leq \min\{Q_k, Q_k + q_{is}\} \quad i \in V^+ \setminus V, k \in K, s \in S \quad (34)$$

$$x_{ij}^k = \{0,1\} \quad i, j \in V, k \in K \quad (35)$$

$$x_{jis}^{k+} = \{0,1\} \quad i, j \in V, k \in K, s \in S \quad (36)$$

$$x_{jis}^{k-} = \{0,1\} \quad i, j \in V, k \in K, s \in S \quad (37)$$

I vincoli (17) (18) sono simili al vincolo (2), e assicurano che tutte le richieste note siano servite. Da notare che non è consentito saltare il servizio per le richieste in ritardo. Il vincolo (19) assicura che tutti i veicoli che lasciano il deposito vi facciano poi ritorno. I vincoli (20) e (21) assicurano invece che se un veicolo serve un determinato nodo di pick up, allora lo stesso veicolo serve anche il corrispondente nodo di delivery, mentre il vincolo (22) esprime il classico vincolo di conservazione del flusso (valido per tutti i nodi eccetto che per il deposito). I vincoli (23) e (24) costituiscono la versione stocastica, già linearizzata dei vincoli (14) e (15) del problema deterministico, mentre i vincoli (25) e (26) permettono il calcolo della durata di ogni rotta in ogni scenario. I vincoli (27), (28), (29) e (30) assicurano rispettivamente il rispetto dei vincoli temporali sulla durata massima delle rotte, sulle finestre temporali fissate per ogni nodo e sul tempo massimo di viaggio per ogni cliente. Il vincolo (31) afferma che un viaggio tra il nodo  $i$  e  $j$  nella soluzione di primo stadio non può essere saltato nella soluzione di secondo stadio se non è già presente nella soluzione di primo stadio; similmente, il vincolo (32) proibisce che, gli archi già visitati prima del tempo  $t$ , nella soluzione di primo stadio, possano essere rimossi dalla soluzione di secondo stadio. Infine, i vincoli (33) e

(34) corrispondono ai vincoli di capacità dei veicoli, mentre i vincoli (35), (36) e (37) definiscono le variabili come binarie.

## 4. Algoritmo risolutivo per lo Stochastic DARP

Per risolvere lo Stochastic DARP è stata ideata una euristica basata sull'idea di considerare le informazioni fornite dagli scenari per la costruzione della soluzione di primo stadio.

### 4.1. Schema *Euristica 1*

1. Si genera il problema  $P_0$  che considera solo le chiamate già conosciute prima del tempo  $T$ . Si determina la soluzione  $SL_0$  utilizzando l'algoritmo risolutivo Tabu Search.
2. Per ogni scenario  $s$  si genera un problema  $P_s$  costituito dai vertici del problema  $P_0$  e dai clienti aggiuntivi specifici dello scenario  $s$ . Quindi, per ogni scenario, si individuerà una soluzione  $SL_s$  applicando l'euristica tabu e assumendo  $SL_0$  come soluzione iniziale.
3. A partire dalle soluzioni  $SL_s$  e  $SL_0$  si individua una nuova soluzione di primo stadio  $SL_0^*$  generata attraverso un criterio euristico di fusione delle soluzioni. Se la soluzione di primo stadio  $SL_0^*$  non è ammissibile, si trova una sua approssimazione invocando nuovamente la tabu search.
4. I problemi  $P_s$  definiti al punto 2 vengono risolti applicando una euristica tabu modificata e assumendo  $SL_0^*$  come soluzione iniziale. L'euristica tabu modificata non permette nessuna modifica alle rotte della soluzione iniziale per gli spostamenti pianificati prima del tempo  $T$ . Le soluzioni così ottenute rappresentano le soluzioni di secondo stadio.

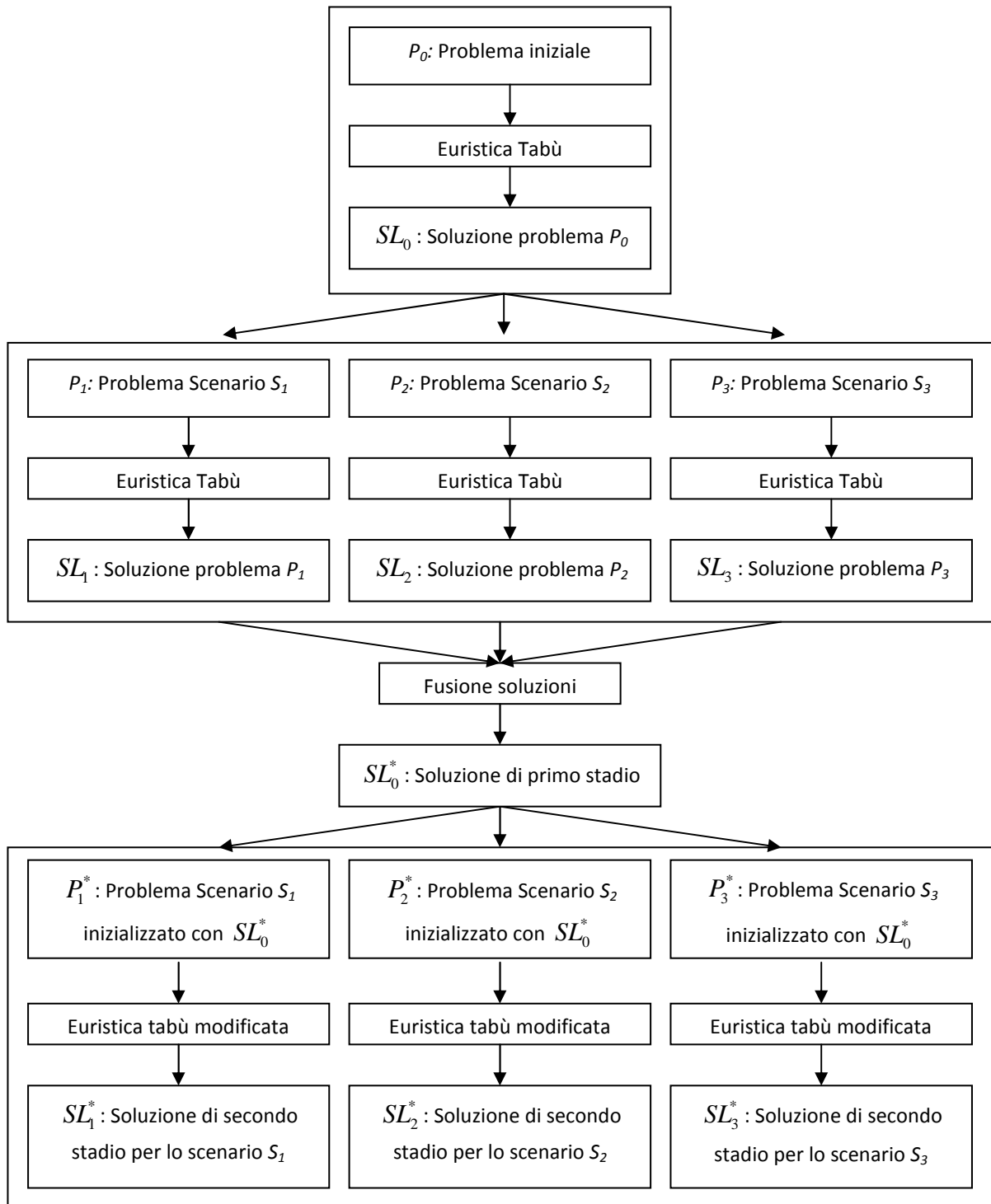


Figura 3: Schema dell'Euristica 1 per la soluzione dello Stochastic DARP

In pratica l'algoritmo trova prima una soluzione di primo stadio e in seguito le soluzioni di secondo stadio. Tramite la procedura di fusione delle soluzioni la soluzione di primo stadio viene integrata con le informazioni derivanti da ogni scenario. A questo punto, occorre notare che:

- Il punto 1 dell'algoritmo è risolto attraverso la tabu search (codice in C++).
- Il punto 2 definisce un nuovo problema per ogni scenario. I nuovi problemi  $P_s$  altro non sono che il precedente problema  $P_o$  con l'aggiunta dei nuovi vertici (specifici per ogni scenario  $s$ ). Ogni  $P_s$  viene quindi risolto con l'euristica tabu ma utilizzando  $SL_0^0$  come soluzione iniziale.
- Il punto 3 determina  $SL_0^1$  costruita in modo da essere ammissibile e "baricentrica" rispetto ai diversi scenari.
- L'euristica modificata, denominata "tabu con blocco nodi", tiene in conto del semplice fatto che al tempo  $T$  parte delle rotte sono già state percorse, e quindi non è possibile modificare le parti di rotte già percorse per servire i nuovi clienti.

Di seguito il punto 3 della prima euristica viene illustrato dettagliatamente. Si noti che tale punto rappresenta la parte veramente innovativa dell'algoritmo proposto.

#### 4.2. Descrizione della Procedura per la fusione delle soluzioni

La procedura di fusione delle soluzioni accetta in input le varie soluzioni ottenute considerando uno scenario alla volta e le fonde insieme per generare la soluzione di primo stadio  $S_{L0}^1$ . La procedura di fusione consiste in due differenti steps:

1. *Assegnamento delle rotte*: per ogni scenario, ogni rotta  $(R_s^k)^0$  (i.e. la k-esima rotta relativa allo scenario  $S$  all'iterazione 0) deve essere assegnata ad una ed una sola delle rotte  $(R_0^k)^0$  della soluzione iniziale  $SL_0^0$ . In altre parole, ad ogni rotta  $(R_0^k)^0$  viene assegnata una ed una sola rotta di ogni differente scenario  $S$ . L'assegnamento viene effettuato sulla base della

similitudine della rotta  $(R_s^k)^0$  con la rotta  $(R_0^k)^0$ , dove per similitudine si intende il numero di nodi in comune. Tale assegnamento è necessario in quanto l'esecuzione della procedura tabu sui problemi  $P_s$  può portare a delle soluzioni  $SL_s^0$  che differiscono di molto dalla soluzione  $SL_0^0$  iniziale.

2. *Costruzione delle rotte  $(R_0^k)^1$* : al fine di costruire la soluzione  $SL_0^1$ , ogni vertice  $i$  del problema  $P_0$  deve essere assegnato ad una ed una sola rotta  $(R_0^k)^1$ . Si noti che la notazione  $(R_0^k)^1$  indica la rotta  $k$ -esima relativa al problema iniziale  $P_0$  all'iterazione 1. Per effettuare tale assegnamento si può calcolare la probabilità di assegnare il vertice  $i$  alla rotta  $k$  sulla base dell'assegnamento delle rotte effettuato nel precedente step. In particolare, la probabilità  $q_{ik}$  del vertice  $i$  di essere assegnato alla rotta  $k$  sarà uguale alla somma delle probabilità  $p_s$  degli scenari per cui la rotta  $(R_s^k)^0$  (contenente il vertice  $i$ ) è stata assegnata alla rotta  $(R_0^k)^0$ . Anche l'ordine di visita di ogni vertice all'interno della propria rotta dovrà essere determinato attraverso un calcolo probabilistico. In particolare, la posizione  $(Pos_0^{ki})^1$  di ogni nodo  $(N_0^{ki})^1$  della soluzione  $SL_0^1$  sarà pari alla somma di convoluzione<sup>1</sup> tra le posizioni dello stesso nodo nelle precedenti soluzioni  $SL_s^0$  e le probabilità dei rispettivi scenari.

---

<sup>1</sup> La somma di convoluzione tra i vettori  $A = \{A_0, A_1, \dots, A_n\}$  e  $B = \{B_0, B_1, \dots, B_n\}$  è uguale a:  $\sum_{i=0}^n A_i \cdot B(n-i)$



### 4.3. Esempio operativo

Si supponga che il problema iniziale  $P_0$  contenga i seguenti vertici:

**Tabella 1: Vertici del problema originale**

	0	1	2	3	4	5	6	7	8
Vertici originali $P_0$		P	P	P	P	D	D	D	D

Si noti che i vertici "1, 2, 3, 4" sono nodi di pickup, mentre i nodi "5, 6, 7, 8" sono di delivery. In particolare, la merce caricata al nodo 1 va consegnata al nodo 5, il carico del nodo 2 va al nodo 6 e così via.

La soluzione  $S_{L_0}^0$  (trovata tramite l'euristica tabu, primo step dell'euristica) è la seguente:

**Tabella 2: Soluzione problema originale**

$S_{L_0}^0$		
$(R_0^1)^0$	$(R_0^2)^0$	$(R_0^3)^0$
0	0	0
1	3	0
4	7	
2	0	
5		
8		
6		
0		

Ovvero  $S_{L_0}^0$  è formata da tre diverse rotte; la prima rotta parte dal deposito, passa dal primo, quarto, secondo, quinto, ottavo e sesto vertice prima di tornare al deposito, Similmente per la seconda rotta che visita terzo e settimo nodo. La terza rotta invece non si muove dal deposito.

Si suppone che durante la giornata possano arrivare nuove chiamate. In particolare, sono previsti due scenari:

**Tabella 3: Scenari considerati**

Scenario	Vertici aggiuntivi				Probabilità
Primo scenario	a	b			0.70
	P	D			
Secondo scenario	c	d	e	f	0.30
	P	P	D	D	

Anche in questi casi, la merce caricata al vertice “a” va consegnata al vertice “b”, il vertice “c” di pickup ha come corrispondente vertice di delivery “e” ed il vertice “d” il vertice “f”.

Risolvendo i due problemi  $P_1$  e  $P_2$  relativi ai due scenari, si ottengono le seguenti soluzioni:

**Tabella 4: Soluzioni dei problemi relativi ai due scenari**

$SL_1^0$		
$(R_1^1)^0$	$(R_1^2)^0$	$(R_1^3)^0$
0	0	0
1	0	3
4		a
2		b
5		7
8		0
6		
0		

$SL_2^0$		
$(R_2^1)^0$	$(R_2^2)^0$	$(R_2^3)^0$
0	0	0
3	1	0
2	4	
c	d	
e	5	
6	8	
7	f	
0	0	

A questo punto, si entra nel terzo step dell’euristica proposta: la costruzione della soluzione di primo stadio (fusione delle soluzioni). In primo passo da effettuare è il 3.1, ovvero l’assegnamento delle rotte.

Si può notare come la rotta  $(R_1^1)^0$  del problema relativo al primo scenario sia “simile” alla rotta  $(R_2^2)^0$  del problema relativo al secondo scenario, e che entrambe siano “simili” alla rotta  $(R_0^1)^0$  del problema

iniziale. È quindi logico creare la rotta  $(R_0^1)^1$  di fusione a partire dalle rotte  $(R_1^1)^0$  e  $(R_2^2)^0$ . Abbiamo però bisogno di un criterio oggettivo per giudicare quanto due rotte siano simili.

Un criterio ragionevole per giudicare la similarità tra due rotte consiste nel contare i vertici in comune. Andiamo quindi a misurare la similarità delle rotte delle soluzioni  $SL_1^0$  e  $SL_2^0$  con le rotte della soluzione  $SL_0^0$

**Tabella 5: Assegnazione rotte**

	$(R_0^1)^0$	$(R_0^2)^0$	$(R_0^3)^0$
$(R_1^1)^0$	8	2	2
$(R_1^2)^0$	2	2	2
$(R_1^3)^0$	2	4	2
	$(R_0^1)^0$	$(R_0^2)^0$	$(R_0^3)^0$
$(R_2^1)^0$	4	4	2
$(R_2^2)^0$	6	2	2
$(R_2^3)^0$	2	2	2

Sembra quindi logico assegnare la rotta  $(R_1^1)^0$  alla rotta  $(R_0^1)^1$  in quanto la similarità tra le due rotte è 8 (molto maggiore rispetto alla similarità con le altre due rotte di  $SL_0^0$ ).

Ugualmente semplice risulta assegnare la rotta  $(R_1^3)^0$  a  $(R_0^2)^0$  ed anche la rotta  $(R_2^2)^0$  a  $(R_0^1)^0$ . Per le altre rotte la situazione si presenta più difficile, in quanto ci sono dei casi in cui una rotta è ugualmente simile a due rotte contemporaneamente (ad esempio,  $(R_2^1)^0$  potrebbe essere assegnata sia a  $(R_0^1)^0$  che a  $(R_0^2)^0$ ). In caso di "parità" si può sempre scegliere un criterio di assegnamento alternativo, come ad esempio assegnare solo una rotta di ogni scenario ad una rotta del problema iniziale.

Seguendo tale principio, andiamo ad assegnare le rotte prodotte dagli scenari alle rotte della soluzione iniziale:

**Tabella 6: Assegnazione rotte**

$SL_0^0$	$SL_1^0$	$SL_2^0$	$\Rightarrow$	$SL_0^1$
$(R_0^1)^0$	$(R_1^1)^0$	$(R_2^2)^0$		$(R_0^1)^1$
$(R_0^2)^0$	$(R_1^3)^0$	$(R_2^1)^0$		$(R_0^2)^1$
$(R_0^3)^0$	$(R_1^2)^0$	$(R_2^3)^0$		$(R_0^3)^1$

Seguendo gli assegnamenti sopra riportati, per costruire la nuova rotta  $(R_0^1)^1$  verranno utilizzate le rotte  $(R_1^1)^0$  e  $(R_2^2)^0$ ; si procede allo stesso modo per le altre rotte. Ovvero le rotte  $(R_1^3)^0$  e  $(R_2^1)^0$  sono associate alla rotta  $(R_0^2)^0$ , ed ovviamente le rotte  $(R_1^2)^0$  e  $(R_2^3)^0$  alla rotta  $(R_0^3)^0$ .

A questo punto, si può passare al punto 3.2, ovvero alla costruzione della soluzione  $SL_0^1$ . Una volta effettuati gli assegnamenti, possiamo procedere alla costruzione della nuova soluzione. Quali vertici dovranno far parte della rotta  $(R_0^1)^1$ ? E quali vertici faranno invece parte di  $(R_0^2)^1$  e di  $(R_0^3)^1$ ?

Semplicemente, attribuiamo ogni vertice alla rotta "più probabile". Ovvero, utilizziamo gli assegnamenti effettuati in precedenza, andiamo a calcolare quale è la probabilità di appartenenza di ogni vertice ad ogni nuova rotta. Per effettuare tale ricostruzione, si costruisce una tabella con i vertici del problema iniziale per righe e le rotte  $(R_0^k)^1$  come colonne:

**Tabella 7: Matrice assegnamento vertici alle nuove rotte**

	$(R_0^1)^1$	$(R_0^2)^1$	$(R_0^3)^1$
1	1	0	0
2	0.7	0.3	0
3	0	1	0
4	1	0	0
5	1	0	0
6	0.7	0.3	0
7	0	1	0
8	1	0	0

Nell'esempio, si ha che il vertice "1" ha probabilità 1 di essere assegnato alla rotta  $(R_0^1)^1$  perché entrambe le rotte  $(R_1^1)^0$  ed  $(R_2^2)^0$  nel quale era presente risultano assegnate alla rotta  $(R_0^1)^1$ . Il vertice due era invece presente nelle rotte  $(R_1^1)^0$  e nelle rotte  $(R_2^2)^0$ , assegnate rispettivamente a  $(R_0^1)^1$  ed a  $(R_0^2)^1$ ; quindi il vertice 2 avrà probabilità 0.7 di essere assegnato a  $(R_0^1)^1$  (in quanto 0.7 è la probabilità dello scenario 1, ovvero lo scenario a cui si riferisce  $(R_1^1)^0$ ) e probabilità 0.3 di essere assegnato a  $(R_0^2)^1$

A questo punto, le nuove rotte  $(R_0^k)^1$  possono essere ottenute assegnando ogni vertice alla rotta  $(R_0^k)^1$  per la quale si riscontra la massima probabilità. Seguendo tale principio, andiamo ad assegnare le rotte prodotte dagli scenari alle rotte della soluzione iniziale:

**Tabella 8: Nuova soluzione  $SL_0^1$**

$SL_0^1$	vertici
$(R_0^1)^1$	1 2 4 5 6 8
$(R_0^2)^1$	3 7
$(R_0^3)^1$	No vertici

Rimane da risolvere il problema dell'ordine con cui i vertici dovranno essere visitati (ovvero determinare il valore  $(Pos_0^{ki})^1$ ). Per risolvere tale problema si ricorre ad una matrice di posizionamento.

**Tabella 9: Matrice di posizionamento**

	$(R_0^1)^1$	$(R_0^2)^1$	$(R_0^3)^1$
1	1	NA	NA
2	3	NA	NA
3	NA	1	NA
4	2	NA	NA
5	4	NA	NA
6	6	NA	NA
7	NA	4.6	NA
8	5	NA	NA

In questo caso, si ha che il vertice "1" occupa la prima posizione sia nella rotta  $(R_1^1)^0$  che nella rotta  $(R_2^2)^0$ , quindi la posizione del vertice uno è  $(0.7*1+0.3*1)/(0.7+0.3)$ . Per calcolare la posizione del vertice "2", si prende in considerazione solo la rotta  $(R_1^1)^0$ , poiché l'altra rotta in cui il vertice "2" è presente, ovvero  $(R_2^2)^0$ , è stata assegnata a  $(R_0^2)^1$ ; quindi per il vertice "2" si ha  $(3*0.7)/0.7 = 3$ . Il vertice sette è presente in posizione 4 nella rotta  $(R_1^3)^0$  ed in posizione 6 nella rotta  $(R_2^1)^0$ ; quindi la sua posizione è  $(0.7*4+0.3*6)/(0.7+0.3) = 4.6$ . Ordinando i nodi a seconda dei valori nella matrice di posizionamento, si ottiene la seguente soluzione di primo stadio:

**Tabella 10: Soluzione di primo stadio**

$SL_0^1$		
$(R_0^1)^1$	$(R_0^2)^1$	$(R_0^3)^1$
0	0	0
1	3	0
4	7	
2	0	
5		
6		
8		
0		

Casualmente, la soluzione è identica alla soluzione iniziale. La soluzione di fusione può essere utilizzata come soluzione di "Primo stadio" in quanto derivata dalla soluzione della parte deterministica del problema considerando poi le modifiche richieste dalle soluzioni relative ai singoli scenari.

Per rendere ancora più chiaro il concetto di matrice di posizionamento, supponiamo di avere le seguenti matrici fittizie (relative ad un problema più complesso qui non specificato) di assegnamento e posizionamento:

**Tabella 11: Matrice di assegnamento fittizia**

Vertici originali $P_o$	$(R_o^1)^1$	$(R_o^2)^1$
1	0.8	0.2
2	0.7	0.3
3	0.2	0.8
4	0.8	0.2
5	0.7	0.3
6	0.2	0.8

**Tabella 12: Matrice di posizionamento fittizia**

Vertici originali $P_o$	$(R_o^1)^1$	$(R_o^2)^1$
1	3.5	
2	1.2	
3		1.3
4	4.7	
5	5.9	
6		5.3

In questo caso la soluzione  $SL_o^1$  diverrebbe la seguente:

**Tabella 13: Soluzione  $SL_o^1$  fittizia**

$(R_o^1)^1$	$(R_o^2)^1$
0	0
2	3
1	6
4	0
5	
0	

Questo perché nella matrice di posizionamento il vertice "2" risulta avere una "posizione pesata" all'interno delle rotte che hanno permesso la creazione di  $(R_o^1)^1$  di 1.2, contro la posizione pesata del vertice "1" che vale 3.5.

#### 4.4. Motivazione generale sulla scelta euristica e sul blocco clienti

L'intera euristica si basa su due passaggi fondamentali:

1. La generazione della soluzione di primo stadio attraverso la fusione delle soluzioni relative ai diversi scenari (passo 3 della procedura). Tale fusione è motivata dalla volontà di costruire una soluzione di primo stadio che sia "baricentrica" rispetto ai cambiamenti da dover effettuare in ogni scenario al fine di servire i nuovi clienti. In pratica, a partire dalla soluzione  $SL_0^0$  trovata considerando i soli clienti deterministici, si tenta di costruire una soluzione di primo stadio  $SL_0^1$  nella quale l'ordine di visita dei clienti deterministici sia tale da minimizzare il costo atteso da sostenere per servire i nuovi clienti.

2. Assumendo di conoscere la soluzione ottimale di primo stadio, rimane il problema di determinare le soluzioni di secondo stadio. In pratica, le soluzioni di secondo stadio devono specificare come agire non appena diviene noto quale scenario si sia realizzato. Siccome si suppone che le richieste aggiuntive da soddisfare si conosceranno solo in seguito ad un generico momento  $T$ , è ovvio che tutte le soluzioni di secondo stadio prevedranno le stesse identiche rotte della soluzione di primo stadio fino al suddetto momento  $T$ . Questo perché fino al momento  $T$  non esiste alcun motivo per cui le rotte da seguire debbano scostarsi dalla soluzione di primo stadio. Dopo  $T$ , ogni soluzione di primo stadio deve prevedere come soddisfare i rimanenti clienti deterministici più i nuovi clienti aggiuntisi dopo  $T$ . Quindi, nel momento in cui si calcolano le soluzioni di secondo stadio, bisogna partire dalla soluzione di primo stadio, considerando come immutabili (bloccati) i clienti serviti fino al tempo  $T$ .



## 5. Descrizione esperimenti, risultati e valutazioni

### 5.1. Introduzione

Il presente capitolo descrive il protocollo sperimentale adottato per valutare le prestazioni dell'euristica proposta, presentata nel precedente capitolo, e i risultati delle sperimentazioni effettuate.

Saranno dapprima presentati i risultati dell'euristica rispetto alla soluzione esatta del modello (si veda paragrafo 3.5) ottenuta attraverso l'utilizzo del solver Cplex. Questo primo blocco di esperimenti è stato condotto su test di piccole dimensioni per verificare la bontà della soluzione euristica ottenuta.

Nella seconda parte del capitolo saranno invece presentati i risultati ottenuti considerando problemi test di dimensioni più elevate tratti dalla letteratura ed opportunamente modificati per tener conto dell'incertezza. In questa seconda fase, i risultati dell'euristica sono stati confrontati con un'euristica base di confronto. Tale scelta è stata dettata dall'impossibilità sia di risolvere in tempi ragionevoli con Cplex il modello stocastico sia di trovare un lower bound di buona qualità con cui poter ragionevolmente effettuare dei confronti (il rilassamento lineare del modello è infatti molto debole e fornisce lower bounds di pessima qualità).

### 5.2. Validazione della qualità della soluzione euristica ottenuta su piccole istanze

Al fine di confrontare le performance dell'Euristica 1 con un algoritmo esatto, è stato implementato il modello del DARP stocastico all'interno del programma di ottimizzazione AIMMS.

AIMMS fornisce una potente interfaccia grafica ed un linguaggio particolarmente espressivo per la formulazione di una vasta gamma di problemi di ottimizzazione; inoltre, diversi risolutori (e.g. CPLEX, AOA, MINOS, BARON) sono inclusi all'interno della suite. In particolare, il risolutore CPLEX è stato utilizzato per risolvere il modello del DARP stocastico studiato, trattandosi della scelta di default per modelli di *Mixed Integer Programming* (MIP).

Un problema di non poco conto è consistito nella determinazione della dimensione delle istanze da sottoporre al modello in AIMMS. Il DARP stocastico è un problema di tipo combinatorio e la sua risoluzione, attraverso un metodo, richiede dei tempi di calcolo che aumentano vertiginosamente con l'aggiunta di poche richieste, o anche di un solo nodo.

Per tale motivo prima di procedere con una sperimentazione estensiva del modello, è stata effettuata una sperimentazione preliminare prendendo in considerazione una singola istanza di piccole dimensioni ed aumentandone progressivamente il numero di nodi.

In particolare è stata scelta casualmente l'istanza numero 7 utilizzata da Laporte, ed a partire da tale istanza è stata creata una micro istanza costituita da 9 richieste e 3 scenari. Si è quindi proceduto con il testare il modello su tale micro istanza; successivamente è stata aumentata la dimensione della micro istanza di test di una richiesta alla volta, fino ad arrivare al numero di 12 richieste. In più, si è anche tentato di risolvere istanze con 15 e 20 nodi. I risultati in termini di tempo di esecuzione e costo della soluzione AIMMS sono riportati nella seguente tabella:

**Tabella 14: Risultati Cplex su piccole istanze di prova**

Nome TEST	Num richieste	Num scenari	Costo		Tempo	
			Cplex	Euristica 1	Cplex	Euristica 1
9r_3s	9	3	62.539	70.558	~3 h	1.6 min
10r_3s	10	3	55.644	77.924	~6 h	2 min
11r_3s	11	3	92.395 <sup>2</sup>	83.534	~12 h	2.6 min
12r_3s	12	3	92.789 <sup>3</sup>	85	~18 h	2.9 min
15r_3s	15	3	Nessun risultato da AIMMS dopo 24 ore			
20r_3s	20	3	Nessun risultato da AIMMS			

Come si può facilmente notare, il tempo computazionale aumenta rapidamente all'aumentare delle richieste da soddisfare. Nel caso di 20 richieste AIMMS non produce nessuna soluzione, neanche raggiungendo il limite massimo di iterazioni posto come default. Al contrario, il tempo richiesto dall'Euristica 1 è visibilmente:

<sup>2</sup> Soluzione trovata dopo aver raggiunto il limite massimo di iterazioni

<sup>3</sup> Soluzione trovata dopo aver raggiunto il limite massimo di iterazioni

- (a) largamente inferiore;
- (b) crescente solo linearmente con il numero delle richieste contenute nelle istanze.

Per quanto riguarda i costi delle soluzioni, quando AIMMS identifica la soluzione ottima, l'euristica solitamente produce soluzioni di livello inferiore; tuttavia, quando AIMMS raggiunge il limite delle iterazioni senza aver identificato la soluzione ottimale (come nel caso delle istanze con 11 e 12 richieste), allora la soluzione fornita dall'euristica risulta migliore.

Per testare la scalabilità dell'euristica su una batteria più significativa di istanze, sono state create delle micro istanze (avendo sempre come base il test bed utilizzato da Laporte) con le seguenti caratteristiche:

1. otto richieste e due scenari;
2. dieci richieste e tre scenari;
3. dodici richieste e cinque scenari.

Tali micro istanze sono state create scegliendo casualmente il numero di richieste indicato dall'insieme delle richieste di ogni istanza e lasciando invariato ogni altro parametro.

Per le istanze con 8 richieste non è stato settato un tempo massimo di risoluzione, mentre per le istanze con 10 e 12 richieste è stato settato un tempo massimo pari a tre ore.

La seguente tabella riporta i risultati ottenuti sulle istanze di test con 8 richieste e 2 scenari.

**Tabella 15: Risultati Euristicia 1 e Cplex sulle istanze con 8 richieste e 2 scenari**

8 Richieste - 2 Scenari	Costo		Tempo		
	<i>Istanza</i>	<i>Cplex</i>	<i>Euristicia 1</i>	<i>Cplex (min)</i>	<i>Euristicia 1 (min)</i>
	pr01	65.067	67.065	11.23	<1
	pr02	56.046	56.046	237.41	<1
	pr03	82.789	82.79	237.77	<1
	pr04	53.862	64.272	16.01	<1
	pr05	49.234	57.107	4.92	<1
	pr06	60.834	69.209	11.53	<1
	pr07	68.072	68.072	12.24	<1
	pr08	64.84	73.422	9.092	<1
	pr09	67.646	67.647	28.38	<1
	pr10	72.065	72.854	76.96	<1
	<i>Media</i>	64.045	67.848	64.5542	<1

Si può prontamente notare che AIMMS ha trovato la soluzione ottimale in un tempo ragionevole. Anche l'Euristicia 1 ha fornito una soluzione, sia pure non ottima in tempi notevolmente ridotti rispetto all'algoritmo esatto. Le seguente tabella riporta la differenze percentuale tra il costo delle soluzioni fornite dall'Euristicia 1 e il costo delle soluzioni fornite da Cplex.

Le percentuali sono calcolate come:

$$(\text{Costo Euristicia 1} - \text{Costo Cplex}) / (\text{Costo Cplex})$$

**Tabella 16: Differenze percentuali tra i costi delle soluzioni dell'Euristicia 1 e i costi delle soluzioni Cplex**

8 Richieste - 2 Scenari	Delta Costo
<i>Istanza</i>	<i>Euristicia1 - Cplex</i>
pr01	2.98%
pr02	0.00%
pr03	0.00%
pr04	16.20%
pr05	13.79%
pr06	12.10%
pr07	0.00%
pr08	11.69%
pr09	0.00%
pr10	1.08%
<i>Media</i>	5.78%

Come si può notare i gap percentuali sono abbastanza contenuti.

La seguente tabella riporta invece i risultati sulle istanze con 10 richieste:

**Tabella 17: Risultati Euristicia 1 e Cplex sulle istanze con 10 richieste e 3 scenari**

10 Richieste - 3 Scenari	Costo		Tempo	
	Istanza	Cplex	Euristicia 1	Cplex (min)
pr01	96.089	96.616	180	<1
pr02	55.886	63.506	180	<1
pr03	77.435	90.328	180	<1
pr04	61.416	81.301	180	<1
pr05	56.313	69.366	180	<1
pr06	66.877	74.382	180	<1
pr07	59.225	79.871	180	<1
pr08	79.387	82.46	180	<1
pr09	64.24	73.091	180	<1
pr10	75.605	86.539	180	<1
<i>Media</i>	69.247	79.746	180	<1

Sulle istanze da dieci richieste Cplex raggiunge sempre il limite delle tre ore, segno che la soluzione ottima non è stata individuata. Bisogna però notare che la soluzione ottenuta con Cplex è comunque migliore di quella fornita dall'euristica; dal punto di vista dei tempi di risoluzione comunque l'euristica si dimostra sempre incredibilmente veloce.

**Tabella 18: Differenze percentuali tra i costi delle soluzioni dell'Euristicia 1 e i costi delle soluzioni Cplex**

10 Richieste - 3 Scenari	Delta Costo
Istanza	Euristicia1 - Cplex
pr01	0.55%
pr02	12.00%
pr03	14.27%
pr04	24.46%
pr05	18.82%
pr06	10.09%
pr07	25.85%
pr08	3.73%
pr09	12.11%
pr10	12.63%
<i>Media</i>	13.45%

La differenza in termini percentuali, tra i risultati dell'Euristicia 1 e i risultati di Cplex, oscilla tra circa lo 0.55% ed il 25.85%. Questo significa che utilizzare l'euristica su piccole istanze comporta, in media, l'ottenimento di una soluzione di buona ( a volte non ottima) qualità, in tempi sicuramente ragionevoli.

Infine, l'ultima tabella riporta i risultati sulle istanze da dodici richieste:

**Tabella 19: Risultati Euristicia 1 e Cplex sulle istanze con 12 richieste e 5 scenari**

12 Richieste - 5 Scenari	Costo		Tempo	
	<i>Istanza</i>	<i>Cplex</i>	<i>Euristicia</i>	<i>Cplex (min)</i>
pr01	na	96.513	180	<1
pr02	578.041	61.371	180	<1
pr03	na	87.346	180	<1
pr04	na	89.983	180	<1
pr05	na	65.654	180	<1
pr06	na	101.833	180	<1
pr07	na	81.211	180	<1
pr08	603.289	90.502	180	<1
pr09	333.021	94.061	180	<1
pr10	na	96.545	180	<1
<i>Media</i>	504.78	86.502	180	<1

In questo caso è ovvio che nel limite delle tre ore Cplex non è in grado di fornire una soluzione, mentre l' Euristicia 1 risolve le istanze sempre in brevissimo tempo.

Le differenze in termini percentuali tra l'euristica e Cplex per istanze con dodici richieste sono riportate di seguito:

**Tabella 20: Differenze percentuali tra i costi delle soluzioni dell'Euristicia 1e i costi delle soluzioni Cplex**

12 Richieste - 5 Scenari	Delta Costo
<i>Istanza</i>	<i>Euristicia1 - Cplex</i>
pr01	
pr02	-841.88%
pr03	
pr04	
pr05	
pr06	
pr07	
pr08	-566.60%
pr09	-254.05%
pr10	
<i>Media</i>	-554.18%

In questo caso la situazione risulta drammaticamente diversa. Il metodo esatto utilizzato da Cplex non riesce a trovare una soluzione ammissibile (avendo fissato un tempo massimo di run pari a tre ore) per la maggior parte delle istanze. Allo stesso tempo, nei pochi casi in cui Cplex fornisce una soluzione, le

soluzioni fornite dall'euristica sono fino ad otto volte migliori in termini di costi. Inoltre, anche nel caso di dodici richieste l'euristica richiede meno di un minuto per fornire una soluzione.

In conclusione, si nota come per piccolissime istanze (numero di richieste  $\leq 10$ ), l'utilizzo di un metodo esatto può essere vantaggioso, rispetto all'utilizzo di una euristica, per la soluzione del DARP stocastico. Tuttavia, non appena il numero di richieste aumenta, anche di poco, l'utilizzo di un metodo esatto risulta, nella pratica, infattibile, a causa dell'elevato carico computazionale richiesto per la risoluzione esatta del modello.

### **5.3. Validazione dell'euristica su istanze tratte dalla letteratura: confronto con un'altra euristica "base"**

Al fine di confrontare la bontà dell'Euristica 1 proposta su problemi test tratti dalla letteratura e considerata l'impossibilità di effettuare il confronto con Cplex, si è considerato come base di confronto un altro semplice approccio euristico.

Tale approccio consiste nel risolvere il problema originale considerando solo i clienti già noti all'inizio del tempo di servizio e trascurando completamente la componente stocastica del problema.

La soluzione così trovata costituisce direttamente la soluzione di primo stadio  $SL_0^0$ . Le soluzioni di secondo stadio sono trovate come precedentemente descritto. La differenza tra questo approccio semplificato e l'euristica proposta consiste quindi nella non "baricentricità" della soluzione  $SL_0^0$ ; ovvero  $SL_0^0$  nel secondo approccio è costruita senza tener conto degli scenari, quindi ci si aspetta che modificare  $SL_0^0$  dovrebbe risultare maggiormente costoso nel momento in cui lo scenario si realizzi (ovvero il costo atteso dovrebbe essere maggiore). Lo schema di questo approccio (Figura 4) è riportato di seguito:

1. Si genera il problema  $P_0$  che considera solo le chiamate già conosciute prima del tempo  $T$ . Si determina la soluzione  $SL_0$  utilizzando l'algoritmo risolutivo Tabu Search. La soluzione  $SL_0$  viene assunta come soluzione di primo stadio  $SL_0^*$
2. Per ogni scenario  $s$  si genera un problema  $P_s$  costituito dai vertici del problema  $P_0$  e dai clienti aggiuntivi specifici dello scenario  $s$ . I problemi  $P_s$  vengono risolti applicando una euristica tabu modificata e assumendo  $SL_0^*$  come soluzione iniziale. L'euristica tabu modificata non permette nessuna modifica alle rotte della soluzione iniziale per gli spostamenti pianificati prima del tempo  $T$ . Le soluzioni così ottenute rappresentano le soluzioni di secondo stadio.

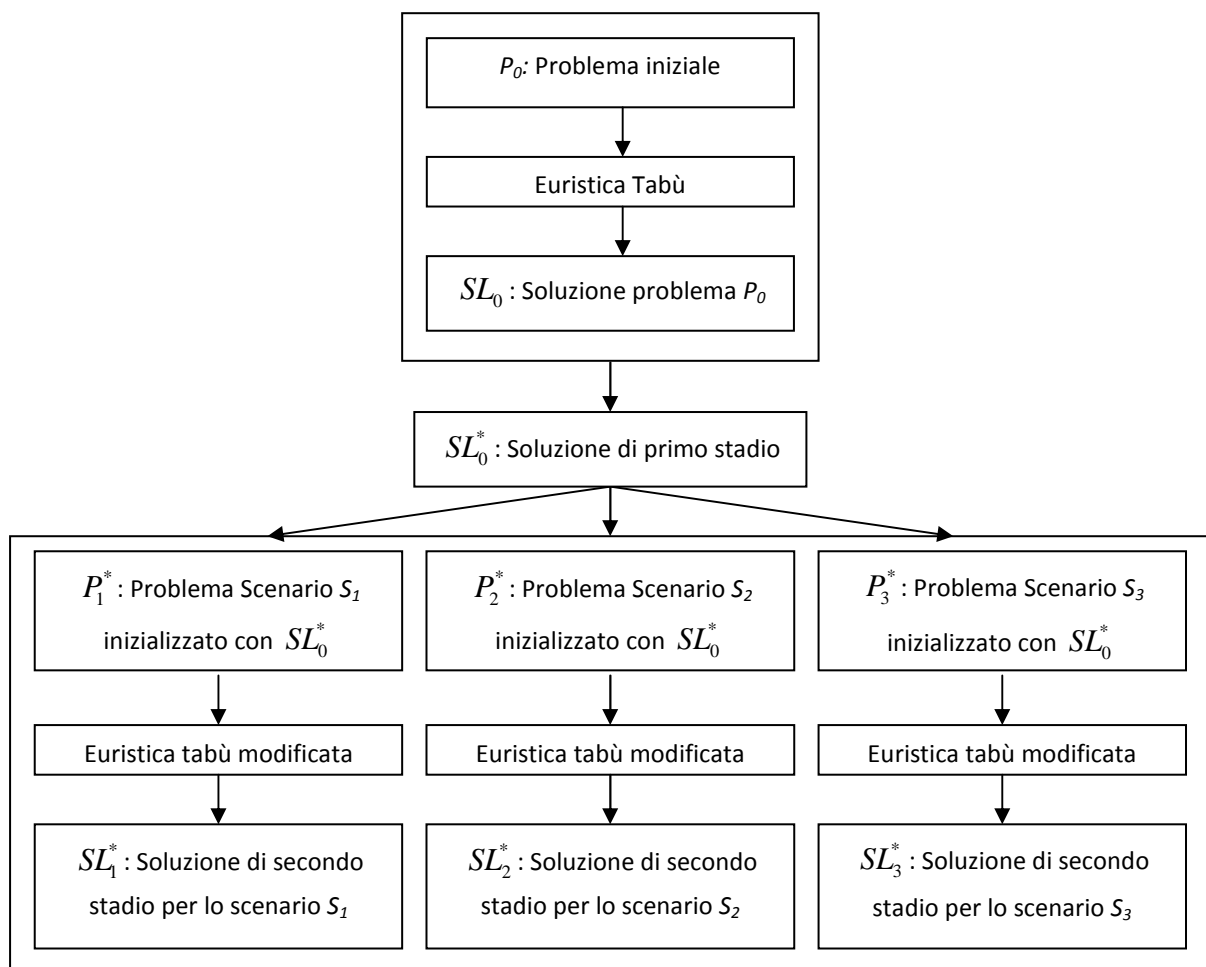


Figure 4: Schema dell'Euristica 2 per la soluzione dello Stochastic DARP



Una volta definita la base di confronto, è stato necessario adattare le istanze di Cordeau et Laporte, Cordeau e Laporte (2003), pubblicamente disponibili all'indirizzo web: <http://neumann.hec.ca/chairedistributique/data/darp/tabu/> pensate per il DARP statico, alle esigenze del DARP stocastico. La strategia adottata per trasformare una istanza DARP "statica" in "stocastica" è sintetizzata di seguito:

1. si assume che le richieste stocastiche rese note solo durante l'arco della giornata siano  $M$ ;
2. si ipotizza l'esistenza di  $S$  possibili scenari;
3. ad ogni scenario  $s$  si assegnano (a) una probabilità e (b) un numero casuale di richieste tra 0 ed  $M$ .  
La stessa richiesta può essere assegnata a più di uno scenario. Si noti che in tal modo lo stesso scenario non può contenere due volte la stessa richiesta aleatoria, mentre la stessa richiesta aleatoria può essere assegnata a diversi scenari.
4. Le probabilità degli scenari sono normalizzate per avere sommatoria pari ad uno.

In appendice è riportato un esempio esplicativo su come sono state modificate le istanze deterministiche di Laporte in modo da tener conto dell'incertezza nelle domande..

Le sperimentazioni hanno sono state condotte sullo spazio delle configurazioni definito dai seguenti parametri:

1. Il numero massimo di richieste aleatorie  $M$  posto pari al 20%, 50% e 80% del totale delle richieste.
2. Il numero di scenari da considerare: per tutte le istanze si sono considerati 30, 60 e 90 scenari, rispettivamente.

Inoltre, per ogni istanza è stato ipotizzato un istante di realizzazione degli scenari (ovvero il momento in cui si viene a conoscenza di quali siano le domande da servire) pari a 240 (ovvero la metà del tempo massimo per ogni rotta).

Complessivamente sono state generate 90 istanze considerando che ogni valore di aleatorietà (20%, 50% e 80%) è stato sperimentato rispettivamente per ogni numero di scenari (30, 60 e 90) e considerando il numero di istanze (10) prese come riferimento. Le 90 istanze sono state eseguite su tre run generando quindi 270 istanze test.

#### 5.4. Risultati Euristiche 1

Di seguito sono sintetizzati i risultati ottenuti dalle sperimentazioni sulle istanze generate per l'euristica ideata.

Nelle tabelle che seguono sono riportati i risultati numerici relativi all'euristica 1. Ogni riga della tabella si riferisce ad una particolare combinazione di istanza utilizzata, numero di scenari considerati e numero di richieste aleatorie. Inoltre, ogni riga riporta le prestazioni medie ottenute su tre diversi run in termini di costo atteso e di tempo di esecuzione.

Per poter leggere in maniera chiara i risultati sotto riportati è opportuno dare delle indicazioni sul significato delle varie voci delle tabelle. Ogni tabella riporta il test effettuato per ogni istanza con le relative richieste per ognuna di esse; per ogni sotto-istanza generata è indicato il tempo medio di esecuzione espresso in minuti ed il costo medio atteso.

Le voci  $R_i^{S(M)}$  riportate sulle righe della prima colonna di ogni tabella sono di seguito illustrate:

- $i$  rappresenta il numero dell'istanza considerata nel test;
- $S$  rappresenta il numero di scenari considerati nell' $i$  – *sima* istanza. Per ogni istanza sono stati considerati 30, 60 e 90 scenari.
- $M$  rappresenta il numero di richieste aleatorie esaminate per l'istanza  $i$ . Il numero di richieste aleatorie  $M$  sono scelte a caso tra le richieste deterministiche considerando una aleatorietà del 20%, 50% e 80%.

Ad esempio, prendendo come spunto la tabella 17,  $R_1^{30(4)}$  rappresenta il test sull'istanza 1 con 30 scenari e 4 richieste aleatorie queste ultime calcolate come il 20% delle 24 richieste totali.

**Tabella 21: Risultati della Euristicia 1 per la prima istanza con 24 richieste**

TEST_24 RICHIESTE (48 nodi)	Tempo medio di esecuzione (minuti)	Costo atteso (medio)
$R_1^{30(4)}$	74	213.31
$R_1^{30(12)}$	43	153.62
$R_1^{30(19)}$	33	113.41
$R_1^{60(4)}$	266	211.38
$R_1^{60(12)}$	158	155.12
$R_1^{60(19)}$	117	119.73
$R_1^{90(4)}$	228	221.55
$R_1^{90(12)}$	151	190.93
$R_1^{90(19)}$	116	134.89

**Tabella 22: Risultati della Euristicia 1 per la seconda istanza con 48 richieste**

TEST_48 RICHIESTE (96 nodi)	Tempo medio di esecuzione (minuti)	Costo atteso (medio)
$R_2^{30(9)}$	276	339.5
$R_2^{30(24)}$	179	480.79
$R_2^{30(38)}$	117	434.47
$R_2^{60(9)}$	392	378.73
$R_2^{60(124)}$	740	377.15
$R_2^{60(38)}$	550	321.07
$R_2^{90(9)}$	567	415.7
$R_2^{90(24)}$	679	511.82
$R_2^{90(38)}$	712	310.06

**Tabella 23: Risultati della Euristicia 1 per la terza istanza con 72 richieste**

TEST_72 RIHIESTE (144 nodi)	Tempo medio di esecuzione (minuti)	Costo atteso (medio)
$R_3^{30(14)}$	365	920.02
$R_3^{30(36)}$	657	621.68
$R_3^{30(57)}$	597	663.93
$R_3^{60(14)}$	762	855.54
$R_3^{60(36)}$	414	718.67
$R_3^{60(57)}$	592	522.77
$R_3^{90(14)}$	870	841.67
$R_3^{90(36)}$	589	727.98
$R_3^{90(57)}$	934	467.99

**Tabella 24: Risultati della Euristicia 1 per la quarta istanza con 96 richieste**

TEST_ 96 RICHIESTE (192 nodi)	Tempo medio di esecuzione (minuti)	Costo atteso (medio)
$R_4^{30(19)}$	443	1096.15
$R_4^{30(48)}$	286	708.53
$R_4^{30(76)}$	237	579.6
$R_4^{60(19)}$	1170	928.03
$R_4^{60(48)}$	766	806.37
$R_4^{60(76)}$	456	583.2
$R_4^{90(19)}$	1643	922.67
$R_4^{90(48)}$	957	869.53
$R_4^{90(76)}$	785	575.89

**Tabella 25: Risultati della Euristicia 1 per la quinta istanza con 120 richieste**

TEST_ 120 RICHIESTE (240 nodi)	Tempo medio di esecuzione (minuti)	Costo atteso (medio)
$R_5^{30(24)}$	848	1264.47
$R_5^{30(60)}$	364	936.81
$R_5^{30(96)}$	371	984.64
$R_5^{60(24)}$	1320	1146.12
$R_5^{60(60)}$	965	1088.39
$R_5^{60(96)}$	605	640.51
$R_5^{90(24)}$	2030	1208.92
$R_5^{90(60)}$	1746	1156.91
$R_5^{90(96)}$	1172	887.28

**Tabella 26: Risultati della Euristicia 1 per la sesta istanza con 144 richieste**

TEST_ 144 RICHIESTE (288 nodi)	Tempo medio di esecuzione (minuti)	Costo atteso (medio)
$R_6^{30(28)}$	969	1632.1
$R_6^{30(72)}$	776	1385.69
$R_6^{30(115)}$	499	945.07
$R_6^{60(28)}$	2069	1291.9
$R_6^{60(72)}$	1459	1110.15
$R_6^{60(115)}$	965	1020.43
$R_6^{90(28)}$	3599	1400.46
$R_6^{90(72)}$	231	1777.45
$R_6^{90(115)}$	1938	993.05

**Tabella 27: Risultati della Euristicica 1 per la settima istanza con 36 richieste**

TEST_36 RICHIESTE (72 nodi)	Tempo medio di esecuzione (minuti)	Costo atteso (medio)
$R_7^{30(7)}$	10	541.9
$R_7^{30(18)}$	9	406.81
$R_6^{30(28)}$	3	251.8
$R_7^{60(7)}$	140	454.62
$R_7^{60(18)}$	110	281.31
$R_7^{60(28)}$	78	235.47
$R_7^{90(7)}$	255	701.5
$R_7^{90(18)}$	187	345.3
$R_7^{90(28)}$	148	215.32

**Tabella 28: Risultati della Euristicica 1 per la ottava istanza con 72 richieste**

TEST_72 RICHIESTE (144 nodi)	Tempo medio di esecuzione (minuti)	Costo atteso (medio)
$R_8^{30(14)}$	379	897.55
$R_8^{30(36)}$	241	703.63
$R_8^{30(57)}$	189	550.91
$R_8^{60(14)}$	846	876.89
$R_8^{60(36)}$	643	697.74
$R_8^{60(57)}$	409	523.58
$R_8^{90(14)}$	1393	924.24
$R_8^{90(36)}$	832	683.3
$R_8^{90(57)}$	525	442.86

**Tabella 29: Risultati della Euristicica 1 per la nona istanza con 108 richieste**

TEST_108 RICHIESTE (216 nodi)	Tempo medio di esecuzione (minuti)	Costo atteso (medio)
$R_9^{30(21)}$	113	1197.82
$R_9^{30(54)}$	614	1024.29
$R_9^{30(86)}$	520	804.49
$R_9^{60(21)}$	1505	1103.14
$R_9^{60(54)}$	1131	929.39
$R_9^{60(86)}$	115	734.47
$R_9^{90(21)}$	260	1238.16
$R_9^{90(54)}$	238	1063.75
$R_9^{90(86)}$	138	828.63

**Tabella 30: Risultati della Euristicia 1 per la decima istanza con 144 richieste**

TEST_ 144 RICHIESTE (288 nodi)	Tempo medio di esecuzione (minuti)	Costo atteso (medio)
$R_{10}^{30(28)}$	230	1527.21
$R_{10}^{30(72)}$	128	1317.91
$R_{10}^{30(115)}$	108	1152.57
$R_{10}^{60(28)}$	410	1460.67
$R_{10}^{60(72)}$	356	1354.4
$R_{10}^{60(15)}$	224	1090.62
$R_{10}^{90(28)}$	606	1418.8
$R_{10}^{90(72)}$	377	1194.89
$R_{10}^{90(115)}$	358	1044.38

La Figura 5: Costo atteso medio in funzione del numero di scenari e della percentuale di aleatorietà5 e la Figura 6: Tempo medio di risoluzione in funzione del numero di scenari e della percentuale di aleatorietà6 sintetizzano rispettivamente i costi attesi medi e i tempi medi di esecuzione rispetto al numero di scenari ed al numero di richieste aleatorie.

La Figura 5 riporta il costo medio (media calcolata su tutte e 10 le istanze testate) in funzione del numero di scenari. Inoltre, il costo è ulteriormente differenziato a seconda della percentuale di richieste aleatorie utilizzate (ognuna rappresentata da una diversa linea).

La Figura 6 riporta le invece il tempo medio in funzione delle stesse variabili. Queste figure mettono chiaramente in luce diverse caratteristiche della prima euristica:

1. Il costo medio decresce all'aumentare delle richieste aleatorie. Anche se contro intuitivo, questo fenomeno è legato al fatto che un numero maggiore di richieste aleatorie si risolve in istanze con, in media, un numero minore di nodi (in quanto i nodi deterministici diminuiscono con l'aumentare della aleatorietà del problema) e quindi un costo di primo stadio inferiore.
2. Il numero di scenari considerato non influenza il costo medio di soluzione in maniera consistente.
3. Il tempo medio di esecuzione cresce sia con il numero di scenari sia con l'aumentare della percentuale di aleatorietà, come prevedibile.

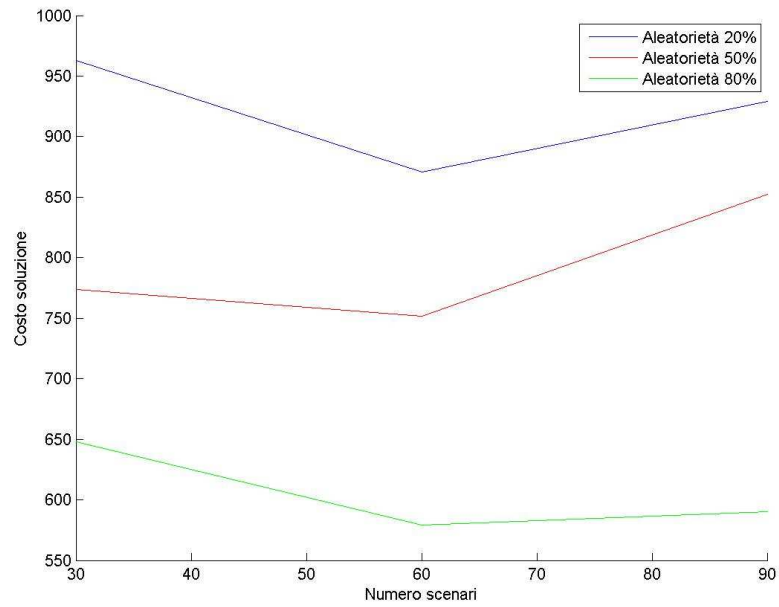


Figura 5: Costo atteso medio in funzione del numero di scenari e della percentuale di aleatorietà (Euristica1)

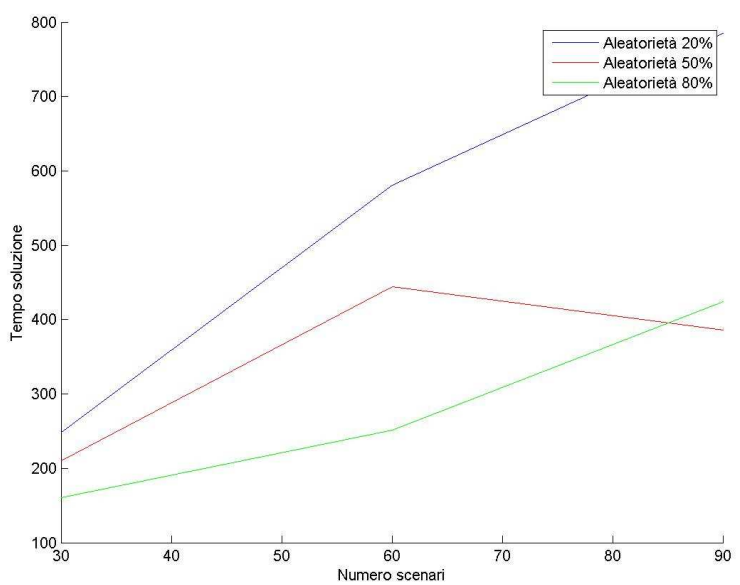


Figura 6: Tempo medio di risoluzione in funzione del numero di scenari e della percentuale di aleatorietà (Euristica 1)

## 5.5 Risultati dell' Euristicica di confronto

Le tabelle che seguono riportano i risultati numerici relativi alla Euristicica 2 sopra descritta. Il significato dei valori è analogo a quello della tabella presentata nel paragrafo 5.4.

**Tabella 31: Risultati della Euristicica 2 per la prima istanza con 24 richieste**

TEST_24 RICHIESTE (48 nodi)	Tempo medio di esecuzione (minuti)	Costo atteso (medio)
$R_1^{30(4)}$	28	197.66
$R_1^{30(12)}$	19	154.67
$R_1^{30(19)}$	13	119.9
$R_1^{60(4)}$	94	203.75
$R_1^{60(12)}$	59	156.81
$R_1^{60(19)}$	51	125.71
$R_1^{90(4)}$	79	196.41
$R_1^{90(12)}$	52	178.51
$R_1^{90(19)}$	46	147.35

**Tabella 32: Risultati della Euristicica 2 per la seconda istanza con 48 richieste**

TEST_48 RICHIESTE (96 nodi)	Tempo medio di esecuzione (minuti)	Costo atteso (medio)
$R_2^{30(9)}$	76	371.44
$R_2^{30(24)}$	61	519.54
$R_2^{30(38)}$	42	552.59
$R_2^{60(9)}$	121	442.59
$R_2^{60(124)}$	256	437.87
$R_2^{60(38)}$	204	367.89
$R_2^{90(9)}$	183	485.13
$R_2^{90(24)}$	211	549.51
$R_2^{90(38)}$	273	312.93



**Tabella 33: Risultati della Euristicia 2 per la terza istanza con 72 richieste**

TEST_72 RICHIESTE (144 nodi)	Tempo medio di esecuzione (minuti)	Costo atteso (medio)
$R_3^{30(14)}$	126	1040.13
$R_3^{30(36)}$	236	606.5
$R_3^{30(57)}$	243	691.6
$R_3^{60(14)}$	288	848.39
$R_3^{60(36)}$	144	716.95
$R_3^{60(57)}$	238	530.13
$R_3^{90(14)}$	290	902.05
$R_3^{90(36)}$	220	722.47
$R_3^{90(57)}$	350	513.34

**Tabella 34: Risultati della Euristicia 2 per la quarta istanza con 96 richieste**

TEST_96 RICHIESTE (192 nodi)	Tempo medio di esecuzione (minuti)	Costo atteso (medio)
$R_4^{30(19)}$	164	978.89
$R_4^{30(48)}$	105	1037.45
$R_4^{30(76)}$	97	666.58
$R_4^{60(19)}$	412	874.36
$R_4^{60(48)}$	271	802.21
$R_4^{60(76)}$	188	610.52
$R_4^{90(19)}$	611	886.59
$R_4^{90(48)}$	338	924.68
$R_4^{90(76)}$	324	611.85

**Tabella 35: Risultati della Euristicia 2 per la quinta istanza con 120 richieste**

TEST_120 RICHIESTE (240 nodi)	Tempo medio di esecuzione (minuti)	Costo atteso (medio)
$R_5^{30(24)}$	289	1280.69
$R_5^{30(60)}$	123	990.51
$R_5^{30(96)}$	143	1061.92
$R_5^{60(24)}$	418	1116.53
$R_5^{60(60)}$	358	1111.48
$R_5^{60(96)}$	236	780.8
$R_5^{90(24)}$	598	1219.25
$R_5^{90(60)}$	650	1141.28
$R_5^{90(96)}$	444	943.7

**Tabella 36: Risultati della Euristicica 2 per la sesta istanza con 144 richieste**

TEST_144 RICHIESTE (288 nodi)	Tempo medio di esecuzione (minuti)	Costo atteso (medio)
$R_6^{30(28)}$	295	1555.28
$R_6^{30(72)}$	251	1309.83
$R_6^{30(115)}$	187	977.53
$R_6^{60(28)}$	748	1184.06
$R_6^{60(72)}$	453	1085.36
$R_6^{60(115)}$	341	1013.21
$R_6^{90(28)}$	948	1312.97
$R_6^{90(72)}$	63	1627.55
$R_6^{90(115)}$	685	1016.05

**Tabella 37: Risultati della Euristicica 2 per la settima istanza con 36 richieste**

TEST_36 RICHIESTE (72 nodi)	Tempo medio di esecuzione (minuti)	Costo atteso (medio)
$R_7^{30(7)}$	3	612.43
$R_7^{30(18)}$	3	550.24
$R_6^{30(28)}$	1	308.07
$R_7^{60(7)}$	49	600.65
$R_7^{60(18)}$	39	315.6
$R_7^{60(28)}$	31	202.79
$R_7^{90(7)}$	88	702.76
$R_7^{90(18)}$	68	405.76
$R_7^{90(28)}$	62	218.19

**Tabella 38: Risultati della Euristicica 2 per la ottava istanza con 72 richieste**

TEST_72 RICHIESTE (144 nodi)	Tempo medio di esecuzione (minuti)	Costo atteso (medio)
$R_8^{30(14)}$	120	895.21
$R_8^{30(36)}$	102	782.53
$R_8^{30(57)}$	70	584.33
$R_8^{60(14)}$	279	881.53
$R_8^{60(36)}$	207	714.21
$R_8^{60(57)}$	161	547.44
$R_8^{90(14)}$	474	912.27
$R_8^{90(36)}$	287	683.17
$R_8^{90(57)}$	207	462.44

**Tabella 39: Risultati della Euristicica 2 per la nona istanza con 108 richieste**

TEST_ 108 RICHIESTE (216 nodi)	Tempo medio di esecuzione (minuti)	Costo atteso (medio)
$R_9^{30(21)}$	41	1172.88
$R_9^{30(54)}$	252	1073.89
$R_9^{30(86)}$	230	944.75
$R_9^{60(21)}$	503	1154.67
$R_9^{60(54)}$	378	907.87
$R_9^{60(86)}$	47	774.96
$R_9^{90(21)}$	89	1276.1
$R_9^{90(54)}$	90	1057.32
$R_9^{90(86)}$	55	861.55

**Tabella 40: Risultati della Euristicica 2 per la decima istanza con 144 richieste**

TEST_ 144 RICHIESTE (288 nodi)	Tempo medio di esecuzione (minuti)	Costo atteso (medio)
$R_{10}^{30(28)}$	83	1474.92
$R_{10}^{30(72)}$	43	1387.57
$R_{10}^{30(115)}$	42	1214.41
$R_{10}^{60(28)}$	163	1397.33
$R_{10}^{60(72)}$	134	1332.08
$R_{10}^{60(115)}$	96	1143.92
$R_{10}^{90(28)}$	237	1448.77
$R_{10}^{90(72)}$	147	1268.89
$R_{10}^{90(115)}$	139	1131.39

I risultati sono sintetizzati nelle Figura 7: Costo atteso medio in funzione del numero di scenari e della percentuale di aleatorietà (Euristicica2)7 e Figura 8: Tempo medio di risoluzione in funzione del numero di scenari e della percentuale di aleatorietà (Euristicica 2)8, che riportano rispettivamente le stesse informazioni delle Figure 5 e 6.

Anche per la seconda euristica valgono le osservazioni fatte per la prima: il costo medio aumenta con il diminuire della percentuale di aleatorietà considerata, mentre rimane relativamente stabile rispetto al numero di scenari. Il tempo computazione aumenta anche in questo caso sia con la percentuale di aleatorietà che con il numero di scenari considerati.

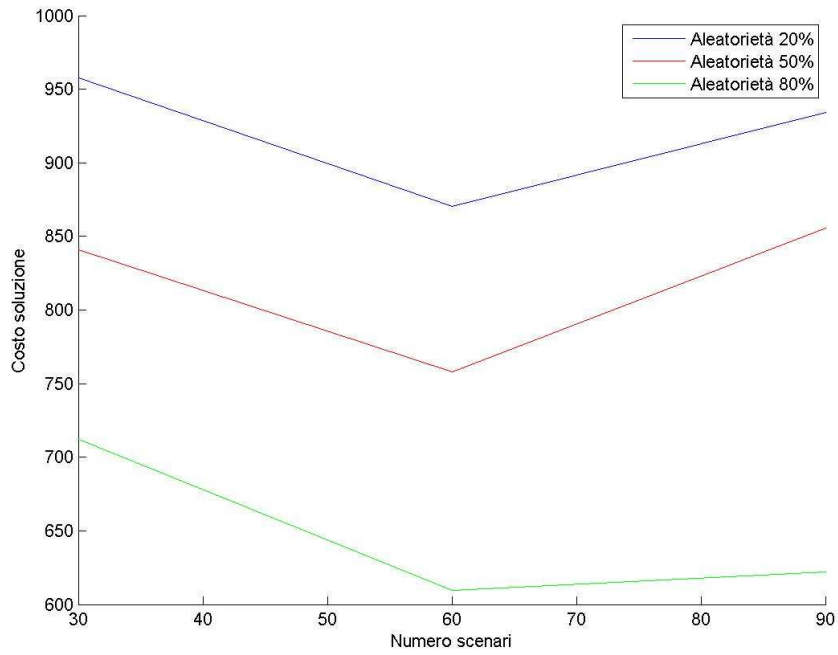


Figura 7: Costo atteso medio in funzione del numero di scenari e della percentuale di aleatorietà (Euristica2)

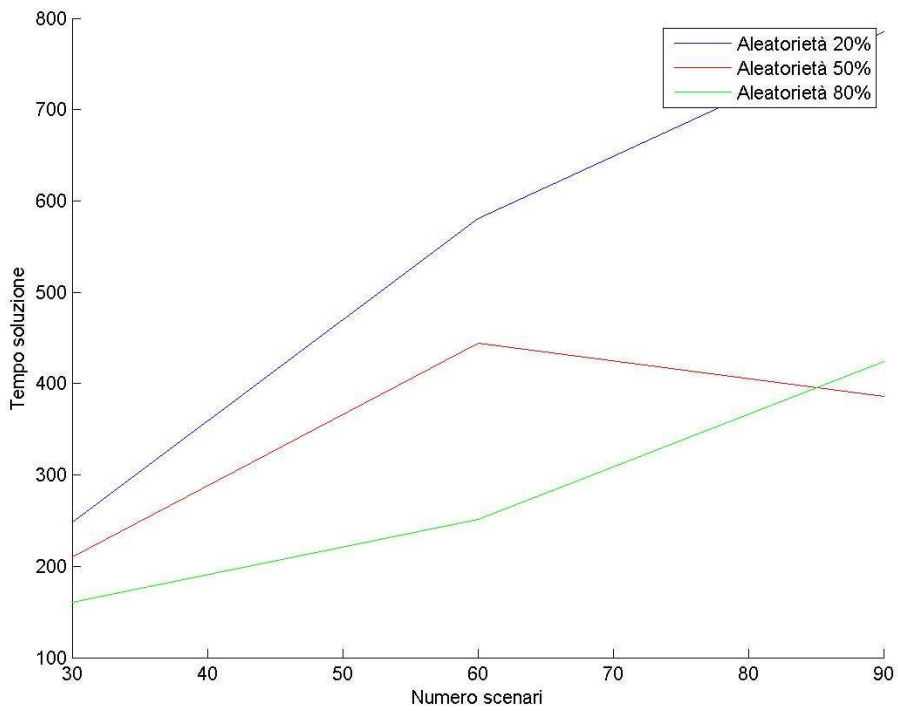


Figura 8: Tempo medio di risoluzione in funzione del numero di scenari e della percentuale di aleatorietà (Euristica 2)

## 5.6. Confronto delle due euristiche (Euristica 1, Euristica 2)

Nei seguenti paragrafi è riportato il confronto dei risultati delle sperimentazioni effettuate con le due diverse euristiche, sia in termini di tempo di esecuzione che in termini di costo delle soluzioni trovate.

Per confrontare le due euristiche, è stata calcolata la loro differenza percentuale in termini di tempi e di costi. Valori negativi di differenza percentuale indicano una migliore performance della Euristica 1; valori positivi una migliore performance della Euristica 2.

In particolare, i valori percentuali sono sempre calcolati come:

$$\frac{(\text{Performance Euristica 1} - \text{Performance Euristica 2})}{(\text{Performance Euristica 2})}$$

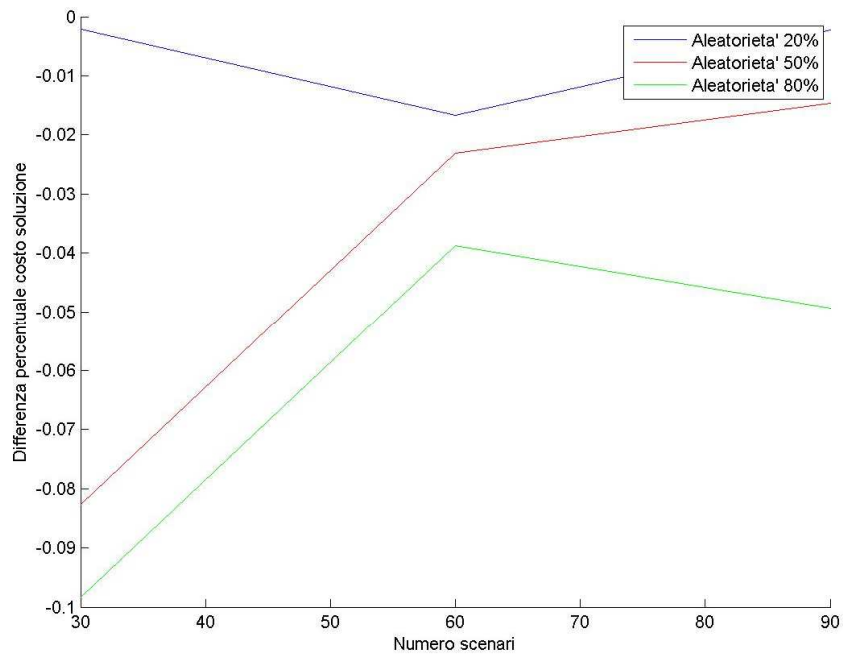


Figura 9: Differenza percentuale in termini di costo medio atteso tra l'Euristica 1 e l'Euristica 2.

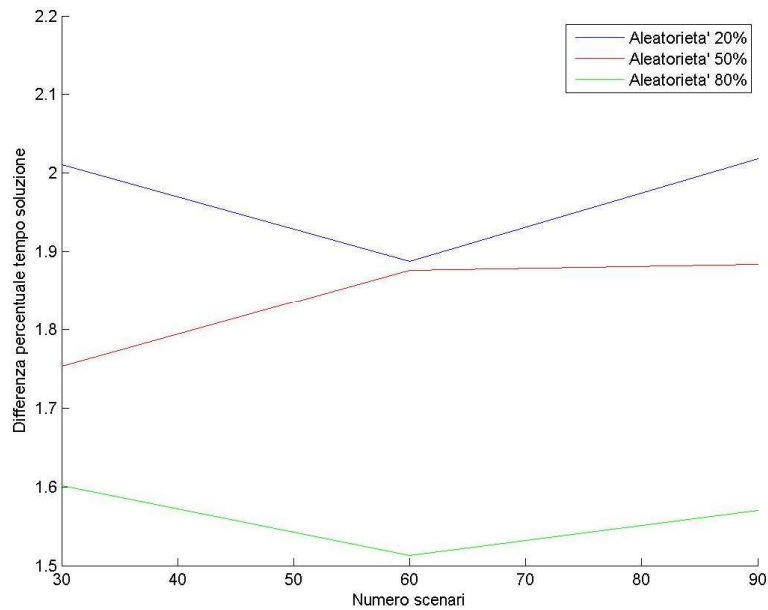


Figura 10: Differenza percentuale in termini di tempo medio di risoluzione tra l'Euristica 1 e l'Euristica 2

I due grafici sopra riportati mettono chiaramente in evidenza i seguenti importanti punti:

1. L'Euristica 1 produce risultati mediamente migliori della Euristica 2. La differenza in termini percentuali raggiunge il suo massimo (10%) per istanze con basso numero di scenari e alto numero di richieste aleatorie.
2. La differenza tra le due euristiche diviene maggiormente marcata via via che la percentuale di aleatorietà aumenta; ciò è facilmente spiegabile poiché maggiore è il numero di richieste aleatorie, maggiore è la possibilità che la soluzione "baricentrica" fornita dalla prima euristica sia effettiva nell'abbattere il costo atteso della soluzione;
3. La riduzione in termini di costo atteso fornita dalla Euristica 1 non sembra seguire un andamento ben delineato in funzione del numero di scenari; infatti nel caso di media ed alta aleatorietà, un basso numero di scenari corrisponde alla massima differenza in termini di costi percentuali; tale comportamento non è però confermato dalle istanze con il 20% di richieste aleatorie.
4. L'Euristica 2 è sempre più veloce dell'Euristica 1, con differenze fino a circa il 200% .

## 6. Conclusioni

Il DARP è un importante e difficile problema di routing che si incontra in diversi contesti e che sta guadagnando una sempre e più crescente importanza soprattutto nel modellare servizi che coinvolgono il trasporto di persone, ed in particolare di disabili ed anziani, e che pertanto richiedono alti livelli di efficienza ed efficacia in termini di puntualità, riduzione dei tempi di inattività e velocità del tempo di corsa.

Dopo più di venti anni di ricerca si è pervenuti ad ottime euristiche per il caso statico che permettono di risolvere problemi con grandi istanze in tempi ragionevoli, mentre ancora poca attenzione è stata rivolta alla versione stocastica del problema che permette di modellare con più realismo il funzionamento del sistema di trasporto DARP.

Per quanto concerne il DARP stocastico, il numero di lavori presenti in letteratura è considerevolmente limitato come evidenziato nel capitolo 1 sullo stato dell'arte.

Nel problema considerato in questo lavoro di tesi si suppone che una parte delle richieste da servire siano note in anticipo, mentre le richieste rimanenti si rivelino in un dato momento della giornata.

Le richieste aggiuntive, dette richieste aleatorie, vengono modellate attraverso scenari che hanno una data probabilità di realizzazione.

L'utilità del modello considerato può essere compresa più a fondo se si riporta il problema a casi di reale interesse: un esempio può essere dato da una azienda di trasporti che offre il proprio servizio ad un serie di clienti abituali e che deve far fronte all'incertezza derivante dal fatto che non sempre è a conoscenza dei clienti da servire durante l'arco della giornata lavorativa. Tale azienda, sulla base della sua esperienza passata, attraverso uno storico sui dati dei clienti che hanno richiesto il servizio, potrebbe essere in grado di prevedere e di pianificare a priori, anche se parzialmente, quali richieste si aggiungeranno ulteriormente.

Rispetto alle ricerche fatte finora, questo lavoro presenta diversi avanzamenti ed aspetti innovativi. In particolare, è stato proposto un modello di programmazione stocastica a due stadi con ricorso

per il DARP in condizioni di incertezza. Con tale modello si costruisce una soluzione di primo stadio che soddisfi le richieste note in anticipo e che nel contempo ha le caratteristiche di robustezza, in quanto tiene conto, attraverso le variabili di secondo stadio, di come le rotte esistenti dovrebbero essere modificate per accogliere eventuali richieste aleatorie. Nello stesso spirito, la soluzione di primo stadio potrebbe affrontare cancellazioni e altri eventi imprevisti, come guasti e ritardi dovuti al traffico dei veicoli.

Per tale modello è stata sviluppata e testata una nuova euristica risolutiva, denominata Euristica 1, basata sulla idea originale della “fusione degli scenari” per l’ottenimento di una soluzione di primo stadio che risulti essere “baricentrica” rispetto a tutti gli scenari considerati. Tale euristica è stata confrontata con il solver CPLEX su piccole istanze e con una seconda euristica, denominata Euristica 2, basata sulla semplice idea di adottare come soluzione di primo stadio la soluzione del DARP deterministico su istanze tratte dalla letteratura ed opportunamente modificate per tener conto dell’incertezza nelle domande.. Anche questa euristica “naive”, è stata implementata e testata per la prima volta in questa tesi.

Le sperimentazioni hanno permesso di raccogliere preziose informazioni che si spera possano essere utili per altri ricercatori che in futuro volessero sviluppare altre euristiche per il DARP stocastico.

In conclusione, gli esperimenti computazionali hanno indicato che l’Euristica 1 risulta produrre soluzioni fino al 10% migliori in termini di costo atteso della soluzione rispetto all’Euristica 2, al costo di un maggior impegno computazionale. Quando convenga adottare una delle due euristiche, dipende strettamente dal contesto. Qualora il 10% del costo atteso consista in un risparmio notevole, allora l’Euristica 1 sarà sicuramente la scelta principe da adottare. Qualora invece servisse una soluzione in tempi rapidi, anche se più costosa, l’Euristica 2 andrebbe sicuramente privilegiata.

L’adozione di un metodo esatto invece, risulta essere conveniente solo quando le istanze sono estremamente piccole (numero richieste  $\leq 10$ ), non sembra quindi essere una scelta praticabile in alcun contesto reale.



## Appendice

La seguente tabella riporta la prima istanza utilizzata in Cordeau e Laporte (2003): tutti i nodi sono considerati deterministici, ovvero si conoscono con sicurezza tutte le richieste sin dall'inizio della giornata lavorativa.

**Tabella 46: Prima Istanza fornita da Laporte (Cordeau e Laporte, 2003)**

Nodo	Coordinata X	Coordinata Y	Tempo di servizio	Quantita' da trasportare	Inizio Finestra Temporale	Fine Finestra Temporale
0	-1.044	2	0	0	0	1440
1	-2.973	6.414	10	1	0	1440
2	-3.066	0.546	10	1	0	1440
3	5.164	0.547	10	1	0	1440
4	-1.317	6.934	10	1	0	1440
5	-6.741	6.832	10	1	0	1440
6	4.891	0.627	10	1	0	1440
7	0.524	2.226	10	1	0	1440
8	-6.5	7.723	10	1	0	1440
9	-0.417	-0.157	10	1	0	1440
10	2.303	1.164	10	1	0	1440
11	2.548	0.629	10	1	0	1440
12	-4.261	-2.639	10	1	0	1440
13	-7.667	9.934	10	1	325	358
14	-2.067	5.789	10	1	111	152
15	-5.204	0.657	10	1	395	421
16	-4.138	5.082	10	1	386	401
17	-9.194	2.759	10	1	86	114
18	-6.512	3.021	10	1	409	426
19	1.86	9.672	10	1	454	470
20	-4.094	8.321	10	1	175	202
21	-3.776	-3.333	10	1	416	453
22	2.377	2.908	10	1	147	177
23	-4.303	2.045	10	1	471	499
24	-3.53	-2.49	10	1	321	346
25	-5.476	1.437	10	-1	258	287
26	-4.933	3.337	10	-1	329	361
27	5.74	2.382	10	-1	209	252
28	-2.275	5.541	10	-1	416	460
29	-5.662	7.334	10	-1	305	349
30	-3.856	-0.37	10	-1	432	458
31	-1.678	1.954	10	-1	202	236

32	-1.156	1.161	10	-1	225	252
33	-4.655	9.797	10	-1	102	123
34	1.623	0.932	10	-1	260	276
35	0.129	0.735	10	-1	178	215
36	-2.64	2.953	10	-1	381	397
37	0.435	1.469	10	-1	0	1440
38	-5.066	-2.313	10	-1	0	1440
39	-2.283	-0.981	10	-1	0	1440
40	-7.11	-1.862	10	-1	0	1440
41	-0.785	3.207	10	-1	0	1440
42	1.188	-2.493	10	-1	0	1440
43	-1.893	-2.373	10	-1	0	1440
44	-1.192	1.175	10	-1	0	1440
45	2.984	1.163	10	-1	0	1440
46	1.227	-5.581	10	-1	0	1440
47	-3.793	-2.161	10	-1	0	1440
48	4.288	-0.297	10	-1	0	1440

Nel caso del DARP stocastico, alcune delle richieste deterministiche devono essere considerate come stocastiche e organizzate in scenari; nel caso in cui si decida di considerare il 20% delle richieste come stocastiche, la lista dei nodi deterministici diventa la seguente:

**Tabella 42: Nodi della prima istanza considerati come “deterministici”**

Nodo	Coordinata X	Coordinata Y	Tempo di servizio	Quantita' da trasportare	Inizio Finestra Temporale	Fine Finestra Temporale
0	-1.044	2	0	0	0	1440
1	-2.973	6.414	10	1	0	1440
2	-3.066	0.546	10	1	0	1440
3	5.164	0.547	10	1	0	1440
4	-1.317	6.934	10	1	0	1440
5	-6.741	6.832	10	1	0	1440
6	4.891	0.627	10	1	0	1440
7	0.524	2.226	10	1	0	1440
8	-6.5	7.723	10	1	0	1440
9	-0.417	-0.157	10	1	0	1440
10	2.303	1.164	10	1	0	1440
11	-7.667	9.934	10	1	325	358
12	-5.204	0.657	10	1	395	421
13	-4.138	5.082	10	1	386	401
14	-9.194	2.759	10	1	86	114
15	-6.512	3.021	10	1	409	426
16	1.86	9.672	10	1	454	470

17	-4.094	8.321	10	1	175	202
18	-3.776	-3.333	10	1	416	453
19	2.377	2.908	10	1	147	177
20	-3.53	-2.49	10	1	321	346
21	-5.476	1.437	10	-1	258	287
22	-4.933	3.337	10	-1	329	361
23	5.74	2.382	10	-1	209	252
24	-2.275	5.541	10	-1	416	460
25	-5.662	7.334	10	-1	305	349
26	-3.856	-0.37	10	-1	432	458
27	-1.678	1.954	10	-1	202	236
28	-1.156	1.161	10	-1	225	252
29	-4.655	9.797	10	-1	102	123
30	1.623	0.932	10	-1	260	276
31	0.435	1.469	10	-1	0	1440
32	-2.283	-0.981	10	-1	0	1440
33	-7.11	-1.862	10	-1	0	1440
34	-0.785	3.207	10	-1	0	1440
35	1.188	-2.493	10	-1	0	1440
36	-1.893	-2.373	10	-1	0	1440
37	-1.192	1.175	10	-1	0	1440
38	2.984	1.163	10	-1	0	1440
39	1.227	-5.581	10	-1	0	1440
40	4.288	-0.297	10	-1	0	1440

Gli 8 nodi considerati come stocastici sono invece raggruppati, in maniera casuale, in un dato numero di scenari, ad esempio 30 (Tabella 43). Per ogni scenario, la prima riga presenta nell'ordine il numero dello scenario, la sua probabilità e il numero di richieste aleatorie incluse nello scenario. Per righe sono invece indicati i dati di ogni richiesta associata allo scenario: X e Y rappresentano le coordinate, Q la quantità da trasportare, T\_in l' inizio della finestra temporale e T\_fin la fine della finestra temporale.

**Tabella 43: Scenari generati con le rispettive richieste associate**

<b>Scenario 1: probabilità 0,03 numero richieste 2</b>					
X	Y	Ts	Q	T_in	T_fin
-4.261	-2.639	10	1	241	1440
-2.64	2.953	10	-1	621	637
-2.067	5.789	10	1	241	281
-5.066	-2.313	10	-1	129	1440
<b>Scenario 2: probabilità 0,07 numero richieste 3</b>					
X	Y	Ts	Q	T_in	T_fin
-4.303	2.045	10	1	471	499
-3.793	-2.161	10	-1	0	1440
-4.261	-2.639	10	1	241	1440
-2.64	2.953	10	-1	621	637
-2.067	5.789	10	1	241	281
-5.066	-2.313	10	-1	129	1440
<b>Scenario 3: probabilità 0,02 numero richieste 2</b>					
X	Y	Ts	Q	T_in	T_fin
-4.303	2.045	10	1	471	499
-3.793	-2.161	10	-1	0	1440
-4.261	-2.639	10	1	241	1440
-2.64	2.953	10	-1	621	637
<b>Scenario 4: probabilità 0,04 numero richieste 3</b>					
X	Y	Ts	Q	T_in	T_fin
2.548	0.629	10	1	241	1440
0.129	0.735	10	-1	418	455
-2.067	5.789	10	1	241	281
-5.066	-2.313	10	-1	129	1440
-4.303	2.045	10	1	471	499
-3.793	-2.161	10	-1	0	1440
<b>Scenario 5: probabilità 0,01 numero richieste 1</b>					
X	Y	Ts	Q	T_in	T_fin
2.548	0.629	10	1	241	1440
0.129	0.735	10	-1	418	455
<b>Scenario 6: probabilità 0,06 numero richieste 3</b>					
X	Y	Ts	Q	T_in	T_fin
2.548	0.629	10	1	241	1440
0.129	0.735	10	-1	418	455
-2.067	5.789	10	1	241	281
-5.066	-2.313	10	-1	129	1440
-4.261	-2.639	10	1	241	1440
-2.64	2.953	10	-1	621	637
<b>Scenario 7: probabilità 0,01 numero richieste 1</b>					

X	Y	Ts	Q	T_in	T_fin
-2.067	5.789	10	1	241	281
-5.066	-2.313	10	-1	129	1440
<b>Scenario 8: probabilità 0,02 numero richieste 1</b>					
X	Y	Ts	Q	T_in	T_fin
-2.067	5.789	10	1	241	281
-5.066	-2.313	10	-1	129	1440
<b>Scenario 9: probabilità 0,01 numero richieste 3</b>					
X	Y	Ts	Q	T_in	T_fin
-2.067	5.789	10	1	241	281
-5.066	-2.313	10	-1	129	1440
-4.303	2.045	10	1	471	499
-3.793	-2.161	10	-1	0	1440
2.548	0.629	10	1	241	1440
0.129	0.735	10	-1	418	455
<b>Scenario 10: probabilità 0,02 numero richieste 3</b>					
X	Y	Ts	Q	T_in	T_fin
-4.261	-2.639	10	1	241	1440
-2.64	2.953	10	-1	621	637
-4.303	2.045	10	1	471	499
-3.793	-2.161	10	-1	0	1440
2.548	0.629	10	1	241	1440
0.129	0.735	10	-1	418	455
<b>Scenario 11: probabilità 0,04 numero richieste 3</b>					
X	Y	Ts	Q	T_in	T_fin
2.548	0.629	10	1	241	1440
0.129	0.735	10	-1	418	455
-4.303	2.045	10	1	471	499
-3.793	-2.161	10	-1	0	1440
-2.067	5.789	10	1	241	281
-5.066	-2.313	10	-1	129	1440
<b>Scenario 12: probabilità 0,07 numero richieste 3</b>					
X	Y	Ts	Q	T_in	T_fin
2.548	0.629	10	1	241	1440
0.129	0.735	10	-1	418	455
-4.303	2.045	10	1	471	499
-3.793	-2.161	10	-1	0	1440
-2.067	5.789	10	1	241	281
-5.066	-2.313	10	-1	129	1440
<b>Scenario 13: probabilità 0,03 numero richieste 4</b>					
X	Y	Ts	Q	T_in	T_fin
-4.303	2.045	10	1	471	499
-3.793	-2.161	10	-1	0	1440

2.548	0.629	10	1	241	1440
0.129	0.735	10	-1	418	455
-4.261	-2.639	10	1	241	1440
-2.64	2.953	10	-1	621	637
-2.067	5.789	10	1	241	281
-5.066	-2.313	10	-1	129	1440
<b>Scenario 14: probabilità 0,05 numero richieste 4</b>					
X	Y	Ts	Q	T_in	T_fin
-4.303	2.045	10	1	471	499
-3.793	-2.161	10	-1	0	1440
-2.067	5.789	10	1	241	281
-5.066	-2.313	10	-1	129	1440
-4.261	-2.639	10	1	241	1440
-2.64	2.953	10	-1	621	637
2.548	0.629	10	1	241	1440
0.129	0.735	10	-1	418	455
<b>Scenario 15: probabilità 0,02 numero richieste 2</b>					
X	Y	Ts	Q	T_in	T_fin
-4.303	2.045	10	1	471	499
-3.793	-2.161	10	-1	0	1440
-4.261	-2.639	10	1	241	1440
-2.64	2.953	10	-1	621	637
<b>Scenario 16: probabilità 0,04 numero richieste 0</b>					
X	Y	Ts	Q	T_in	T_fin
<b>Scenario 17: probabilità 0,02 numero richieste 1</b>					
X	Y	Ts	Q	T_in	T_fin
2.548	0.629	10	1	241	1440
0.129	0.735	10	-1	418	455
<b>Scenario 18: probabilità 0,03 numero richieste 1</b>					
X	Y	Ts	Q	T_in	T_fin
2.548	0.629	10	1	241	1440
0.129	0.735	10	-1	418	455
<b>Scenario 19: probabilità 0,01 numero richieste 3</b>					
X	Y	Ts	Q	T_in	T_fin
-4.261	-2.639	10	1	241	1440
-2.64	2.953	10	-1	621	637
-4.303	2.045	10	1	471	499
-3.793	-2.161	10	-1	0	1440
2.548	0.629	10	1	241	1440
0.129	0.735	10	-1	418	455
<b>Scenario 20: probabilità 0,02 numero richieste 3</b>					
X	Y	Ts	Q	T_in	T_fin

-2.067	5.789	10	1	241	281
-5.066	-2.313	10	-1	129	1440
2.548	0.629	10	1	241	1440
0.129	0.735	10	-1	418	455
-4.261	-2.639	10	1	241	1440
-2.64	2.953	10	-1	621	637
<b>Scenario 21: probabilità 0,01 numero richieste 1</b>					
<b>X</b>	<b>Y</b>	<b>Ts</b>	<b>Q</b>	<b>T_in</b>	<b>T_fin</b>
-4.261	-2.639	10	1	241	1440
-2.64	2.953	10	-1	621	637
<b>Scenario 22: probabilità 0,05 numero richieste 2</b>					
<b>X</b>	<b>Y</b>	<b>Ts</b>	<b>Q</b>	<b>T_in</b>	<b>T_fin</b>
2.548	0.629	10	1	241	1440
0.129	0.735	10	-1	418	455
-4.261	-2.639	10	1	241	1440
-2.64	2.953	10	-1	621	637
<b>Scenario 23: probabilità 0,05 numero richieste 2</b>					
<b>X</b>	<b>Y</b>	<b>Ts</b>	<b>Q</b>	<b>T_in</b>	<b>T_fin</b>
-2.067	5.789	10	1	241	281
-5.066	-2.313	10	-1	129	1440
-4.303	2.045	10	1	471	499
-3.793	-2.161	10	-1	0	1440
<b>Scenario 24: probabilità 0,02 numero richieste 2</b>					
<b>X</b>	<b>Y</b>	<b>Ts</b>	<b>Q</b>	<b>T_in</b>	<b>T_fin</b>
-4.261	-2.639	10	1	241	1440
-2.64	2.953	10	-1	621	637
-4.30	3 2.045	10	1	471	499
-3.793	-2.161	10	-1	0	1440
<b>Scenario 25: probabilità 0,03 numero richieste 1</b>					
<b>X</b>	<b>Y</b>	<b>Ts</b>	<b>Q</b>	<b>T_in</b>	<b>T_fin</b>
-4.303	2.045	10	1	471	499
-3.793	-2.161	10	-1	0	1440
<b>Scenario 26: probabilità 0,07 numero richieste 3</b>					
<b>X</b>	<b>Y</b>	<b>Ts</b>	<b>Q</b>	<b>T_in</b>	<b>T_fin</b>
-4.303	2.045	10	1	471	499
-3.793	-2.161	10	-1	0	1440
-2.067	5.789	10	1	241	281
-5.066	-2.313	10	-1	129	1440
2.548	0.629	10	1	241	1440
0.129	0.735	10	-1	418	455
<b>Scenario 27: probabilità 0,02 numero richieste 1</b>					
<b>X</b>	<b>Y</b>	<b>Ts</b>	<b>Q</b>	<b>T_in</b>	<b>T_fin</b>
2.548	0.629	10	1	241	1440

0.129	0.735	10	-1	418	455
<b>Scenario 28: probabilità 0,08 numero richieste 1</b>					
<b>X</b>	<b>Y</b>	<b>Ts</b>	<b>Q</b>	<b>T_in</b>	<b>T_fin</b>
2.548	0.629	10	1	241	1440
0.129	0.735	10	-1	418	455
<b>Scenario 29: probabilità 0,01 numero richieste 2</b>					
<b>X</b>	<b>Y</b>	<b>Ts</b>	<b>Q</b>	<b>T_in</b>	<b>T_fin</b>
-4.261	-2.639	10	1	241	1440
-2.64	2.953	10	-1	621	637
-2.067	5.789	10	1	241	281
-5.066	-2.313	10	-1	129	1440
<b>Scenario 30: probabilità 0,03 numero richieste 2</b>					
<b>X</b>	<b>Y</b>	<b>Ts</b>	<b>Q</b>	<b>T_in</b>	<b>T_fin</b>
2.548	0.629	10	1	241	1440
0.129	0.735	10	-1	418	455
-2.067	5.789	10	1	241	281
-5.066	-2.313	10	-1	129	1440



## Bibliografia

- [1] [Cordeau, 2006] A Branch and Cut Algorithm for the Dial a Ride Problem. *Operations Research* 54(3), pp 573\_586 (2006)
- [2] [Alsalloum and Rand, 2006] Alsalloum OI and Rand GK. Extensions to emergency vehicle location models. *Computers & Operations Research*, 2006, 33(9): 2725.2743.
- [3] [Attanasio et al., 2004] Attanasio A, Cordeau J-F, Ghiani G, Laporte G. Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem. *Parallel Computing*, 2004, 30(3): 377.387.
- [4] [Araz et al., 2007] Araz C, Selim H, Ozkarahan I. A fuzzy multi-objective covering-based vehicle location model for emergency services. *Computers & Operations Research*, 2007, 34(3): 705.726.
- [5] [Badeau et al., 1997] Badeau P, Gendreau M, Guertin F, Potvin J-Y, Taillard E. A parallel tabu search heuristic for the vehicle routing problem with time windows. *Parallel Computing in Transport Research*, 1997, 5(2): 109.122.
- [6] [Beaudry et al., 2008] Beaudry A, Laporte G, Melo T, Nickel S. Dynamic transportation of patients in hospitals. *OR Spectrum*, 2008, 32(1): 77.107.
- [7] [Berbeglia et al., 2010] Berbeglia G, Cordeau J-F, Laporte G. Dynamic Pickup and Delivery Problems. *European Journal of Operational Research*, 2010, 202(1): 8.15.
- [8] [Borndorfer et al., 1997] Borndorfer R, Grotschel M, Klostermeier F, Kuttner C. Telebus Berlin: Vehicle scheduling in a dial-a-ride system. Technical report SC 97.23, Konrad-Zuse-Zentrum fur Informationstechnik, Berlin, 1997.
- [9] [Cordeau and Laporte, 2003] Cordeau J-F, Laporte G. A tabu search heuristic for the static multivehicle dial-a-ride problem. *Transportation Research Part B: Methodological*, 2003, 37(6): 579.594.
- [10] [Cordeau and Laporte, 2007] Cordeau J-F, Laporte G. The dial-a-ride problem: models and algorithms. *Annals of Operations Research*, 2007, 153(1): 29.46.
- [11] [Coslovich et al., 2005] Coslovich L, Pesenti R, Ukovich W. A two-phase insertion technique of unexpected customers for a dynamic dial-a-ride problem. *European Journal of Operational Research*, 2005, 175(3): 1605.1615.
- [12] [Desaulniers et al., 2002] Desaulniers G, Desrosiers J, Erdmann A, Solomon MM, Soumis F. The VRP with pickup and delivery. In: Toth P, Vigo D (Eds.) *The vehicle routing problem*. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, 2002, pp. 225.242.
- [13] [Desrosiers et al., 1991] Desrosiers J, Dumas Y, Soumis F, Taillefer S, Villeneuve D. An algorithm for mini-clustering in handicapped transport. *Cahiers du GERARD Report G-91-02*, Ecole des Hautes Etudes Commerciales, Montreal, Canada, 1991.
- [14] [Dumas et al., 1989] Dumas Y, Desrosiers J, Soumis F. Large scale multi-vehicle dial-a-ride problems. *Cahiers du GERARD Report G-89-30*, Ecole des Hautes Etudes Commerciales, Montreal, Canada, 1989.

- [15] [Gendreau et al., 1999] Gendreau M, Guertin F, Potvin J-Y, Taillard E. Parallel tabu search for Real-Time Vehicle Routing and Dispatching. *Transportation Science*, 1999, 33(4): 381.390.
- [16] [Gendreau et al., 2006] Gendreau M, Guertin F, Potvin J-Y, Seguin R. Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries. *Transportation Research Part C: Emerging Technologies*, 2006, 14(3): 157.174.
- [17] [Ghiani et al., 2003] Ghiani G, Guerriero F, Laporte G, Musmanno R. Real-time vehicle routing: Solution concepts, algorithms and parallel computing strategies. *European Journal of Operational Research*, 2003, 151(1): 1.11.
- [18] [Glover, 1986] Glover F. Future Paths for Integer Programming and Links to Artificial Intelligence. *Computers & Operations Research*, 1986, 13(5): 533.549.
- [19] [Hansen, 1986] Hansen P. The steepest ascent mildest descent heuristic for combinatorial programming. *Congress on Numerical Methods in Combinatorial Optimization*, Capri, 1986.
- [20] [Hanne et al., 2009] Hanne T, Melo T, Nickel S. Bringing Robustness to Patient Flow Management Through Optimized Patient Transports in Hospitals. *Interfaces*, 2009, 39(3): 241.255.
- [21] [Horn et al., 2002] Horn MET. Fleet scheduling and dispatching for demand-responsive passenger services. *Transportation Research Part C: Emerging Technologies*, 2002, 10(1): 35.63.
- [22] [Ichoua et al., 2000] Ichoua S, Gendreau M, Potvin J-Y. Diversion issues in real-time vehicle dispatching. *Transportation Science*, 2000; 34(4): 426.438.
- [23] [Ichoua et al., 2007] Ichoua S, Gendreau M, Potvin J-Y. Planned route optimization for real-time vehicle routing. In: Zaimpekis V, Tarantilis CD, Giaglis GM and Minis I (Eds), *Dynamic fleet management: concepts, systems, algorithms and case studies*. Springer Science + Business Media, New York, Etats-Unis, 2007. p. 1-18.
- [24] [Ioachim et al., 1995] Ioachim I, Desrosiers J, Dumas Y, Solomon MM, Villeneuve D. A request clustering algorithm for door to door handicapped transportation. *Transportation Science*, 1995,
- [25] [Jaw et al., 1986] Jaw JJ, Odoni AR, Psaraftis HN, Wilson NHM. A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. *Transportation Research Part B: Methodological*, 1986, 20(3): 243.257.
- [26] [Kergosien et al., 2008] Kergosien Y, Lente C, Billaut J-C. A dynamic transportation problem of patients between care units. *34rd International Conference on Operational Research Applied to Health Services (ORAHs f2008)*, Toronto, Canada, 2008.
- [27] [Laporte, 1997] Laporte G. Modeling and solving several classes of arc routing problems as traveling salesman problems. *Computers & Operations Research*, 1997, 24(11): 1057.1061.
- [28] [Larsen, 2001] Larsen A. The dynamic vehicle routing problem. Ph.D. Thesis, Institute of Mathematical Modeling, Technical University of Denmark, 2001.

- [29] [Madsen et al., 1995] Madsen OBG, Ravn HF, Rygaard JM. A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. *Annals of Operations Research*, 1995, 60(1): 193.208.
- [30] [Melachrinoudis et al., 2007] Melachrinoudis E, Ilhan AB, Min H. A dial-a-ride problem for client transportation in a health-care organization. *Computers & Operations Research*, 2007, 34(3): 742.759.
- [31] [Psaraftis, 1980] Psaraftis HN. A dynamic programming approach to the single-vehicle, many-to-many immediate request dial-a-ride problem. *Transportation Science*, 1980, 14(2): 130.154.
- [32] [Rajagopalan et al., 2008] Rajagopalan HK, Saydam C, Xiao J. A multiperiod set covering location model for dynamic redeployment of ambulances. *Computers & Operations Research*, 2008, 35(3): 814.826.
- [33] [Shang and Carolyn, 1996] Shang, JS, Carolyn KC. Multicriteria pickup and delivery problem with transfer opportunity. *Computers and Industrial Engineering*, 1996, 30(4): 631.645.
- [34] [Stein, 1978] Stein D. An asymptotic, probabilistic analysis of a routing problem. *Mathematics of Operations Research*, 1978, 3(2): 89.101.
- [35] [Taillard et al., 1997] Taillard E, Badeau P, Gendreau M, Guertin F, Potvin J-Y. A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation science*, 1997, 31(2): 170.186.
- [36] [Teodorovic and Radivojevic, 1998] Teodorovic D, Radivojevic G. A fuzzy approach to dynamic Dial-A-Ride problem. *Fuzzy Sets and Systems*, 1998, 116(1): 22.33.
- [37] [Xiang et al., 2006] Xiang Z, Chu C, Chen H. A fast heuristic for solving a large-scale static dial-a-ride problem under complex constraints. *European Journal of Operational Research*, 2006, 174(2): 1117.1139.
- [38] [Xiang et al., 2008] Xiang Z, Chu C, Chen H. The study of a dynamic dial a ride problem under time dependant and stochastic environments. *European Journal of Operational Research*, 2008, 185(2): 53.551.
- [39] [Y. Kergosien, Ch. Lenté, D. Piton and J-C. Billaut] A tabu search heuristic for a dynamic transportation problem of patients between care units, May 2010
- [40] [Bodin, L. D., & Sexton, T. (1986)] The multi-vehicle subscriber dial-a-ride problem. *TIMS Studies in Management Science*, 2, 73–86.
- [40] [Brotcorne, L., Laporte, G., & Semet, F. (2003).] Ambulance location and relocation models. *European Journal of Operational Research*, 147, 451–468.
- [41] [Colorni, A., & Righini, G. (2001).] Modeling and optimizing dynamic dial-a-ride problems. *International Transactions in Operational Research*, 8, 155–166.
- [42] [Cordeau, J.-F., & Laporte, G. (2003b) ] The dial-a-ride problem (DARP): variants, modeling issues and algorithms. *4OR: A Quarterly Journal of Operations Research*, 1, 89–101.
- [43] [Cordeau, J.-F., Laporte, G., & Mercier, A. (2001)] A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society*, 52, 928–936.

- [44] [Cordeau, J.-F., Laporte, G., Potvin, J.-Y., & Savelsbergh, M. W. P. (2007a)] Transportation on demand. In C. Barnhart & G. Laporte (Eds.), *Transportation*. Amsterdam: Elsevier.
- [45] [Cordeau, J.-F., Laporte, G., Savelsbergh, M. W. P., & Vigo, D. (2007b)] Vehicle routing. In C. Barnhart & G. Laporte (Eds.), *Transportation*. Amsterdam: Elsevier.
- [46] [Desrosiers, J., Dumas, Y., & Soumis, F. (1986)] A dynamic programming solution of the large-scale single vehicle dial-a-ride problem with time windows. *American Journal of Mathematical and Management Sciences*, 6, 301–325.
- [47] [Diana, M., & Dessouky, M. M. (2004)] A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows. *Transportation Research Part B*, 38, 539–557.
- [48] [Fu, L. (2002)] Scheduling dial-a-ride paratransit under time-varying, stochastic congestion. *Transportation Research B*, 36, 485–506.
- [49] [Gendreau, M., Laporte, G., & Semet, F. (2001)] A dynamic model and parallel tabu search algorithm for real-time ambulance relocation. *Parallel Computing*, 27, 1641–1653.
- [50] [Hunsaker, B., & Savelsbergh, M. W. P. (2002)] Efficient feasibility testing for dial-a-ride problems. *Operations Research Letters*, 30, 169–173.
- [51] [Jørgensen, R. M., Larsen, J., & Bergvinsdottir, K. B. (2007, in press)] Solving the dial-a-ride problem using genetic algorithms. *Journal of the Operational Research Society*.
- [52] [Toth, P., & Vigo, D. (1997)] Heuristic algorithms for the handicapped persons transportation problem. *Transportation Science*, 31, 60–71.
- [53] [Pallottino, P., & Scutellà, M. G. (1998)] Shortest path algorithms in transportation models: classical and innovative aspects. In P. Marcotte & S. Nguyen (Eds.), *Equilibrium and advanced transportation modeling* (pp. 245–281). Boston: Kluwer.
- [54] [Psaraftis, H. N. (1988)] Dynamic vehicle routing problems. In B. L. Golden & A. A. Assad (Eds.), *Vehicle routing: method and studies* (pp. 223–248). Amsterdam: North-Holland.
- [55] [Psaraftis, H. N. (1995)] Dynamic vehicle routing: status and prospects. *Annals of Operations Research*, 61, 143–164.
- [56] [Sexton, T., & Bodin, L. D. (1985a)] Optimizing single vehicle many-to-many operations with desired delivery times: I. Scheduling. *Transportation Science*, 19, 378–410.
- [57] [Wong, K. I., & Bell, M. G. H. (2006)] Solution of the dial-a-ride problem with multi-dimensional capacity constraints. *International Transactions in Operational Research*, 13, 195–208.
- [58] [Ropke, S., Cordeau, J.-F., & Laporte, G. (2007)] Models and branch-and-cut algorithms for pickup and delivery problems with time windows. *Networks*, 49, 258–272.
- [59] [Wolfler Calvo, R., & Colorni, A. (2007)] An effective and fast heuristic for the dial-a-ride problem. *4OR: A Quarterly Journal of Operations Research*, 5, 61–73.

[60] [Ghiani G, Guerriero F, Laporte G, Musmanno] R. Real-time vehicle routing: Solution concepts, algorithms and parallel computing strategies. *European Journal of Operational Research*, 2003, 151(1): 1.11.

[61] [P. Beraldi, P. Festa, G. Ghiani, E. Guerriero, F. Guerriero, P. Legato, E. Manni, R.M. Mazza, R. Musmanno, F. Pascali, O. Pisacane, M.G. Scutellà, C. Triki, F. Vocaturo] *Metodi di supporto alle decisioni in condizioni di incertezza*, forthcoming.