

Università degli Studi di Salerno



## **Facoltà di Scienze Matematiche Fisiche e Naturali**

Dipartimento di Matematica e Informatica

Dottorato di Ricerca in Scienze Matematiche  
Fisiche e Informatiche

XI Ciclo – Nuova Serie

TESI DI DOTTORATO

# **Il problema dell'allocazione dei prodotti nei magazzini: modelli matematici ed approcci euristici.**

CANDIDATO: **FRANCESCO RENDE**

COORDINATORE: **PROF. PATRIZIA LONGOBARDI**

TUTOR: **PROF. CIRO D'APICE**

# INDICE

<b>1</b>	<b>INTRODUZIONE.....</b>	<b>3</b>
<b>2</b>	<b>IL RUOLO DEI MAGAZZINI NELLA <i>SUPPLY CHAIN MANAGEMENT</i> .....</b>	<b>5</b>
2.1	PROGETTAZIONE DEL MAGAZZINO .....	7
2.1.1	<i>Struttura e dimensionamento</i> .....	7
2.1.2	<i>Definizione del layout</i> .....	8
2.1.3	<i>Selezione dell'attrezzatura</i> .....	9
2.1.4	<i>Strategie operative</i> .....	10
2.2	LE ATTIVITÀ DI GESTIONE DEL MAGAZZINO .....	10
2.2.1	<i>Ricezione e spedizione</i> .....	12
2.2.2	<i>Immagazzinamento</i> .....	12
2.2.3	<i>Prelievo degli ordini</i> .....	13
<b>3</b>	<b>IL PROBLEMA DELL'ALLOCAZIONE DELLE MERCI ALLE POSIZIONI DI STOCCAGGIO .....</b>	<b>16</b>
3.1	STORAGE LOCATION ASSIGNMENT PROBLEM .....	16
3.1.1	<i>SLAP con informazioni sugli articoli</i> .....	17
3.1.2	<i>SLAP con informazioni sui prodotti</i> .....	17
3.1.3	<i>SLAP senza informazioni</i> .....	24
3.2	PRODUCT ALLOCATION PROBLEM .....	25
3.3	CONTRIBUTI PRESENTI IN LETTERATURA PER IL PAP .....	28
<b>4</b>	<b>UN MODELLO MATEMATICO PER IL PAP IN UN MAGAZZINO MULTI-LIVELLO CON VINCOLI DI COMPATIBILITÀ FRA I PRODOTTI (MLPAP) .....</b>	<b>32</b>
4.1	DEFINIZIONE DEL MODELLO.....	32
4.2	IMPLEMENTAZIONE DEL MODELLO .....	40
4.3	GENERAZIONE DELLE ISTANZE DI TEST.....	43
4.4	DEFINIZIONE DEGLI SCENARI E SETTAGGIO DEI PARAMETRI DI CONTROLLO.....	44
4.5	IDENTIFICAZIONE DEI PARAMETRI CHIAVE DEL MODELLO.....	46
4.6	COMPORAMENTO DEL MODELLO AL VARIARE DEL NUMERO DI CLASSI DI PRODOTTI .....	47
4.7	COMPORAMENTO DEL MODELLO AL VARIARE DELLA CAPACITÀ DEGLI SLOT .....	50
4.8	COMPORAMENTO DEL MODELLO AL VARIARE DEI PARAMETRI M, W E H .....	53
<b>5</b>	<b>APPROCCI EURISTICI RISOLUTIVI PER IL MLPAP .....</b>	<b>61</b>
5.1	PROCEDURE EURISTICHE E METAEURISTICHE .....	61
5.2	LA PROCEDURA DI RICERCA LOCALE .....	64
5.3	PROGETTAZIONE DI UN PROCEDURA EURISTICA BASATA SULLA RICERCA LOCALE ITERATIVA.....	67
5.4	PROGETTAZIONE DI UN PROCEDURA EURISTICA BASATA SULLA TECNICA DEL ROLLOUT .....	70
5.5	PROGETTAZIONE DI UN PROCEDURA EURISTICA BASATA SU UN PROCESSO DI CLUSTERING.....	73
<b>6</b>	<b>ANALISI DEI RISULTATI SPERIMENTALI .....</b>	<b>77</b>
6.1	ANALISI DELLE PRESTAZIONI IN TERMINI DELLA FUNZIONE DI COSTO .....	77
6.2	ANALISI DEI TEMPI DI ELABORAZIONI .....	82
<b>7</b>	<b>CONCLUSIONI E SVILUPPI FUTURI .....</b>	<b>84</b>
<b>8</b>	<b>APPENDICE A - RISULTATI SPERIMENTALI .....</b>	<b>87</b>
8.1	VARIAZIONE DEL NUMERO DI CLASSI DI PRODOTTI.....	87
8.2	VARIAZIONE DELLA CAPACITÀ DEGLI SLOT .....	88
8.3	VARIAZIONE DEL PARAMETRO W .....	90
8.4	VARIAZIONE DEL PARAMENTRO H.....	96
8.5	VARIAZIONE DEL PARAMENTRO M .....	102
<b>9</b>	<b>BIBLIOGRAFIA.....</b>	<b>109</b>

## 1 Introduzione

I sistemi logistici, con le infrastrutture, i processi, le strategie che di questi sistemi costituiscono le nervature, hanno un'importanza fondamentale nella vita economica di un Paese, inseriti come sono all'interno dei grandi flussi di merci e servizi di un'economia sempre più globalizzata. Proprio per questa loro rilevanza nel ciclo economico ed industriale, negli ultimi venti anni si è assistito ad una continua serie di innovazioni relative alle strategie industriali, alle opzioni organizzative, ai processi aziendali ed ai sistemi gestionali.

Lo scopo di tutti i cambiamenti verificatisi nei diversi processi della logistica è sempre stato quello di contribuire in misura rilevante all'abbassamento dei livelli di costo del ciclo produttivo e distributivo, nell'ambito della più generale ricerca della profittabilità in ciascun sottoprocesso aziendale, per aumentare la competitività in un mercato sempre più complesso. La continua e costante tensione verso il raggiungimento di questi obiettivi ha portato ad un'analoga attenzione verso tutti gli elementi di costo di ciascuna fase del processo logistico, inducendo una serie di ottimizzazioni tecnico-organizzative, ben supportate da una tecnologia sempre più efficace e disponibile a basso costo. Si tratta allora di ripensare l'azienda per renderla capace di individuare e ridurre al minimo le attività a basso valore aggiunto, di analizzare ed eliminare gli sprechi, di ottimizzare i processi interni, di soddisfare le aspettative del cliente, di perseguire il "miglioramento continuo". L'approccio da seguire è pragmatico: i processi e le problematiche relative vanno analizzati e le soluzioni pensate ed applicate come sforzo comune, soluzioni che non devono essere altamente sofisticate, bensì profondamente semplici.

Il seguente lavoro di tesi ha avuto l'obiettivo di studiare ed approfondire le problematiche relative alla gestione efficiente del magazzino ed in particolare lo studio del problema di allocazione dei prodotti ai punti di stoccaggio che rappresenta una delle principali problematiche che si affronta nella progettazione o nella razionalizzazione degli spazi di un magazzino.

Nella gestione di un magazzino il confine tra potenzialità da sfruttare e problemi da risolvere è molto sottile. Le scelte intraprese in fase di progettazione sono determinanti per trasformare le criticità organizzative e/o economiche in punti di forza ed elementi portanti dell'intera struttura organizzativa, produttiva e gestionale dell'impresa. Dietro una gestione efficiente dei processi logistici del magazzino vi è una attenta fase di progettazione, pianificazione e gestione dell'insieme dei processi logistici e organizzativi dell'azienda. Durante la fase di progettazione è necessario ponderare le diverse scelte in modo tale da razionalizzare l'insieme delle risorse a disposizione dell'azienda siano esse di natura materiale o umana.

Nel seguente lavoro di tesi il problema di allocazione dei prodotti ai punti di stoccaggio è stato studiato dal punto di vista formale nel caso particolare dei magazzini multilivello e con vincoli di compatibilità tra classi di prodotti adiacenti (vincoli tipici per le aziende che operano nel settore della grande distribuzione organizzata).

Obiettivo della prima parte del lavoro svolto è stato, infatti, la formulazione di un modello matematico in grado di rappresentare l'insieme dei vincoli del problema preso in esame.

Per istanze reali, il modello definito è, dal punto di vista computazionale, *NP hard* per cui, nella seconda parte del lavoro di tesi, l'attività di ricerca è stata indirizzata verso la progettazione e realizzazione di euristiche e meta euristiche, basate sulla ricerca locale, capaci di individuare soluzioni di buona qualità in tempi di calcolo ragionevoli. In particolare, si è avuto modo di progettare e realizzare una procedura euristica di ricerca locale che è stata utilizzata per implementare due diverse strategie di risoluzione del problema:

- una prima strategia basata sulla tecnica della ricerca locale iterativa, che si prefigge di superare il problema dei minimi locali attraverso l'implementazione di una specifica procedura in grado di effettuare una perturbazione della soluzione corrente;
- una seconda strategia basata sulla tecnica del *rollout*, che consente di valutare, per ogni fase del processo di *rollout*, la migliore soluzione sulla base dei diversi scenari generati.

Un terzo approccio è stato ispirato dalla filosofia organizzativa che considera la compatibilità tra i prodotti il fattore di *familiarità* (*Family Grouping*). Sostanzialmente consiste nel creare *cluster* di prodotti o classi di prodotto, fra di loro compatibili per ottenere una soluzione iniziale in grado di massimizzare la concentrazione dei prodotti compatibili all'intero del *layout*.

La fase di sperimentazione effettuata ha permesso sia di validare gli algoritmi realizzati sia di confrontare i risultati ottenuti attraverso i diversi approcci oggetto di definizione nel seguente lavoro di tesi.

## 2 Il ruolo dei magazzini nella *Supply Chain Management*

I magazzini rappresentano i nodi dei sistemi logistici in cui i materiali, le merci e i prodotti sono ricevuti, stoccati e prelevati per la loro destinazione finale. Rappresentano uno spazio fondamentale per la corretta gestione dei processi produttivi e commerciali di qualsiasi impresa. Il mantenimento di una scorta, infatti, può servire tanto a bilanciare e sincronizzare flussi produttivi, come accade per le industrie manifatturiere, quanto a garantire un determinato livello di servizio, nel caso ad esempio di imprese di commercializzazione o di erogazione di servizi.

Le principali motivazioni che inducono le aziende a dotarsi di un magazzino sono di seguito riportate.

1. Fluttuazioni della domanda: il processo con il quale si manifesta la domanda del mercato è un processo stocastico che se pur attentamente analizzato e studiato tramite delle tecniche di previsione, comporta degli scostamenti tra quanto previsto dall'azienda e quanto richiesto effettivamente dai clienti. Per tale ragione il magazzino serve sia ad evitare la possibilità di andare in rottura di *stock* sia a conservare la quantità di merce invenduta in un determinato periodo per far sì che possa essere disponibile per il periodo successivo.
2. Svalutazione monetaria: la scorta può essere vista come un investimento conveniente per tutti quegli articoli che hanno un indice di rivalutazione superiore al tasso medio di inflazione della moneta.
3. Inaffidabilità dei fornitori: la merce tenuta a magazzino protegge l'azienda da fornitori che non rispettano i termini contrattuali rispetto alle date di consegna.
4. Sconti di quantità: attraverso politiche di acquisizione di grandi quantità di merce si possono realizzare economie di scala e quindi ottenere maggiori profitti.
5. Flessibilità rispetto alle variazioni del *mix* produttivo e alle variazioni dei volumi.

A fronte di questi vantaggi che porta con sé la gestione di un magazzino bisogna tener conto anche dei costi da tenere costantemente sotto controllo. I principali costi sono raggruppabili nelle seguenti categorie:

1. costi economici; la decisione di investire in scorta è una decisione strategica. Se da un lato infatti si vuole garantire un adeguato livello di servizio, in termini di puntualità nelle consegne, dall'altro si deve tener conto dell'immobilizzo di capitale rappresentato dalla scorta e quindi si devono sostenere i costi sotto forma di interessi;
2. costi operativi; le attività svolte all'interno del magazzino generano costi quali ad esempio il pagamento degli stipendi agli addetti alle varie funzioni;

3. costi fissi; infrastrutture, energia, tasse, assicurazioni, ecc;
4. perdita di valore a causa di danneggiamento, obsolescenza, furti, deterioramento; il valore della merce si decrementa con il passare del tempo e quindi c'è bisogno di una gestione razionale che permetta di soddisfare le esigenze produttive e dei clienti.

Sostanzialmente si possono distinguere tre tipi di magazzino: il magazzino di distribuzione, il magazzino di produzione e il magazzino a contratto. Il magazzino di distribuzione è un magazzino in cui vengono stoccati, e a volte assemblati, prodotti provenienti da diversi fornitori che devono essere smistati a vari clienti; il magazzino di produzione è usato per lo stoccaggio di materie prime, semilavorati e prodotti finiti che provengono da un centro di produzione; infine il magazzino a contratto è un magazzino che esegue attività di *warehousing* per conto di diversi clienti.

Il flusso di materiali che attraversa il magazzino, tipicamente, segue uno schema consolidato: la merce, inviata dai fornitori, viene trasportata da autotreni e scaricata in un'apposita area del magazzino denominata area di ricezione (*receiving zone*); qui ha luogo un primo controllo qualitativo e quantitativo della merce, a valle del quale quest'ultima viene preparata per essere stoccata. Se le unità di carico (*pallet*, scatole, ecc.) che arrivano in magazzino differiscono da quelle in cui la merce viene immagazzinata allora si provvede al riassetto della merce. Altrimenti vengono direttamente applicate delle etichette (codice a barre o *tag* RFID – *Radio Frequency Identification* –) alle unità da stoccare e si procede con l'invio del tutto alla cosiddetta area di stoccaggio (*storage zone*). Quest'ultima può suddividersi a sua volta in due zone (non necessariamente fisicamente distinte): la zona ad accesso rapido (*pick/forward zone*) e la zona di stoccaggio remoto (*reserve zone*). Questa distinzione viene adottata per rendere più efficienti le operazioni di prelievo: la *reserve zone* è l'area nella quale vengono stoccati i prodotti in grandi quantità e funge da polmone per la *pick zone*, la quale custodisce carichi minori, dei medesimi prodotti, che consentono operazioni di prelievo più agevoli. Ogni qualvolta perviene all'azienda un ordine da parte dei clienti, viene generata una lista contenente tutte le SKU (*Stock Keeping Units*), e le relative quantità, che guida il processo di prelievo degli ordini (*order picking*) svolto dall'addetto al prelievo. Una volta prelevati tutti i prodotti della lista, questi vengono trasportati nell'area di approntamento delle spedizioni (*shipping zone*) dove vengono imballati e preparati per essere inviati a destinazione.

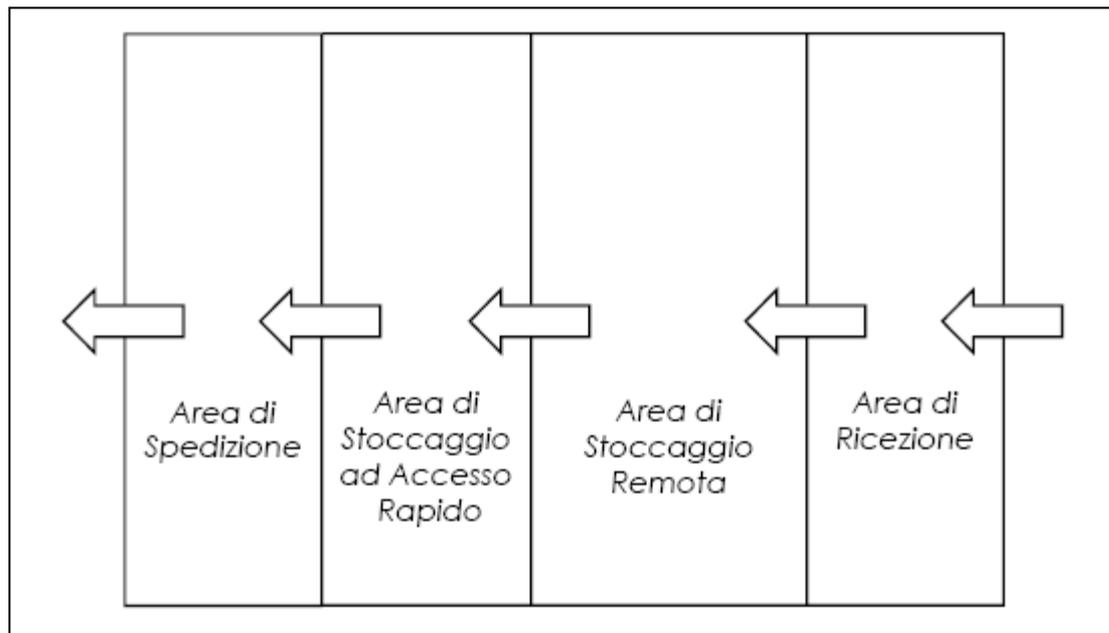


Figura 1- Rappresentazione schematica di un magazzino

Si possono, dunque, raggruppare le attività che hanno luogo in un magazzino in quattro categorie: ricezione, stoccaggio, prelievo e spedizione.

## **2.1 Progettazione del magazzino**

La progettazione del magazzino racchiude in sé tutte le scelte strategiche concernenti la gestione di un magazzino. Si tratta degli aspetti più delicati di cui il *management* si deve occupare in quanto sono decisioni difficilmente reversibili nel breve periodo se non a fronte di ingenti esborsi finanziari.

Le maggiori problematiche da trattare sono la determinazione della struttura del magazzino nel suo complesso, il dimensionamento dello stesso e delle aree in cui esso è suddiviso, la definizione del *layout*, la scelta delle attrezzature e dei mezzi con i quali svolgere le varie operazioni all'interno del magazzino e infine la formulazione delle strategie con le quali condurre l'attività operativa.

### **2.1.1 Struttura e dimensionamento**

La definizione delle dimensioni del magazzino in termini di lunghezza, larghezza e altezza dell'edificio è una delle prime attività a carico del *management*. La determinazione del volume necessario non è un processo facile a causa della molteplicità dei fattori che influenzano tale scelta: il tipo di sistemi di movimentazione dei materiali utilizzati, le modalità di stoccaggio, il numero di articoli, la domanda degli articoli, ecc.

Dopo aver individuato il corretto dimensionamento del magazzino, il passo successivo si sostanzia con l'individuazione delle sezioni funzionali in cui suddividere il magazzino: di quante aree di ricezione e spedizione disporre, se disporre di un'unica area di stoccaggio oppure fare la distinzione tra zona di stoccaggio ad accesso remoto e zona ad accesso rapido, dotarsi di zone di stoccaggio speciali per prodotti che presentano determinate caratteristiche, ecc. L'insieme delle decisioni intraprese si riflette sul flusso che i materiali seguono all'interno del magazzino e costituisce un vincolo nella scelta delle modalità operative che verranno effettuate nella fase di pianificazione operativa. In tale fase, l'obiettivo principale è dunque la definizione dei diversi reparti e la comprensione delle loro interazioni. Successivamente si passa al loro dimensionamento tenendo conto di alcune linee guida fondamentali: la superficie dell'area di ricezione deve essere abbastanza ampia da consentire l'ingresso di più forniture esterne contemporaneamente; la zona di spedizione può eventualmente essere più piccola della zona di ricezione poiché le spedizioni sono sotto il diretto controllo del magazzino e quindi è possibile una loro pianificazione che limiti le congestioni del traffico dei prodotti in uscita; la superficie dell'area di stoccaggio deve essere progettata razionalmente perché è vero che deve essere dimensionata in modo tale da contenere le merci nei periodi di punta e da garantire che i mezzi addetti alla movimentazione non si intralcino a vicenda, però è anche vero che una superficie troppo ampia rispetto alle reali esigenze comporta un aumento del tempo medio di transito e dunque un inutile aggravio dei costi di gestione.

In generale, dunque, un magazzino è suddiviso in diversi dipartimenti che vanno a competere tra loro per occupare lo spazio totale del magazzino. Il problema del dimensionamento può essere visto come un problema di allocazione dello spazio tra questi diversi dipartimenti. L'obiettivo è quello di determinare la dimensione dell'area di ricevimento (i metri quadrati necessari per svolgere le attività di scarico e controllo), la dimensione dell'area di stoccaggio (i metri quadrati necessari per stivare le quantità previste), la dimensione dell'area di prelievo (la superficie per scaffali, la superficie per corridoi, la superficie per aree ausiliarie) e infine la dimensione dell'area di spedizione (la superficie necessaria per effettuare le operazioni di imballaggio, pesatura, marchiatura, ecc).

### **2.1.2 Definizione del *layout***

Lo studio del *layout* del magazzino si articola in due fasi fondamentali:

1. la determinazione del macro *layout*: determinazione della disposizione delle diverse aree operative previste nell'impianto e nei sistemi di movimentazione;
2. la determinazione del micro *layout*: determinazione del *layout* di ogni singola area operativa.

Nella definizione del macro *layout*, oltre ad essere localizzati i diversi reparti individuati, viene anche stabilito il numero di porte di I/O del magazzino. A supporto di questa decisione può essere utilizzata una semplice formula [1] che tiene in considerazione parametri quali il tasso di domanda giornaliero dei diversi prodotti, il tempo medio richiesto per caricare/scaricare un *truck*, la capacità del *truck* e infine il tempo giornaliero dedicato alle operazioni di carico e scarico merci.

Nella definizione del micro *layout*, invece, la parte più interessante concerne la determinazione del *layout* dell'area di stoccaggio. Questa scelta dipende fortemente dalla tipologia di prodotti trattati, dalle dimensioni e dall'unità in cui le merci vengono stoccate: bancali, scatole, singole unità. Quest'ultime influenzano anche la scelta dei sistemi di immagazzinamento che si possono classificare per linee di massima in tre categorie: le pile, gli scaffali e le cassettiere. Dalla scelta di una delle tre modalità deriva anche il numero di corridoi e l'orientamento che questi devono avere all'interno della struttura.

Un altro aspetto da tenere in considerazione è la tipologia di sistemi di movimentazione che si decide di adottare: quest'ultimi influenzano direttamente la larghezza dei corridoi. Esistono tre tipi di sistemi di movimentazione:

- sistema *picker-to-product*: le operazioni di stoccaggio e prelievo vengono effettuate da una squadra di operatori che si avvalgono di carrelli e di veicoli motorizzati;
- sistema *picker-to-belt*: ad operare sono sempre squadre di persone che però completano l'operazione con l'utilizzo di un nastro trasportatore;
- sistema *order-to-picker*: l'intera operazione è eseguita da un sistema completamente automatizzato di immagazzinamento e prelievo, denominato in gergo AS/RS (*automatic storage/retrieval system*), che trasporta le merci fino all'area di preparazione delle commesse.

### **2.1.3 Selezione dell'attrezzatura**

La selezione dell'attrezzatura afferisce la scelta del livello di automazione e dei sistemi di stoccaggio e movimentazione da utilizzare.

Determinare il miglior livello di automazione è una scelta tutt'altro che ovvia: la difficoltà primaria sta nell'identificare le diverse alternative ragionevoli da considerare e successivamente nel come scegliere quella più adatta alla situazione di riferimento. Per tale ragione, nella maggior parte dei casi, la soluzione da adottare viene scelta in base al fattore esperienza di progettisti e *manager*.

#### **2.1.4 Strategie operative**

Quest'area del *warehouse design* si focalizza sulla determinazione di quelle strategie operative, in termini di stoccaggio e di prelievo, che si ripercuotono sull'intero sistema e che non possono essere variate frequentemente.

Per quanto riguarda lo stoccaggio, la principale scelta è la tipologia di sistema da adottare; esistono, sostanzialmente, tre opzioni: un sistema a posizioni fisse, un sistema a posizioni variabili oppure un sistema a classi. Tale decisione è da ponderare accuratamente in quanto condiziona la capacità di stoccaggio dell'intero magazzino.

Per quanto concerne il prelievo, invece, si tratta di decidere se adottare una strategia a prelievo singolo oppure a prelievo di lotti, se implementare un sistema nel quale è l'addetto al prelievo che si reca nella zona di stoccaggio e preleva l'*item* desiderato (*picker-to-part*) oppure un sistema nel quale gli *item* vengono inviati alle baie di prelievo, dove vengono collezionati e utilizzati per il riassetto dell'ordine cliente dall'operatore (*part-to-picker*), se utilizzare o meno una *pick zone*, ecc.

### **2.2 Le attività di gestione del magazzino**

La filosofia da adottare nella gestione di un magazzino, al fine di essere competitivi sul mercato, vede il magazzino non più come una semplice area dove movimentare le merci ma come un centro altrettanto importante quanto le altre funzioni aziendali, un centro che deve produrre un servizio e che non deve essere considerato un costo ma deve generare profitto.

La revisione della politica nei confronti e all'interno dei magazzini è stata generata da diversi fattori fra i quali il più importante è stato, agli inizi degli anni '70 del secolo scorso, l'elevato costo del denaro. Avere magazzini con elevata quantità di materiali stoccati significava principalmente, e significa tuttora, immobilizzi di capitale elevati che venivano sottratti alla liquidità aziendale. Inoltre, il veloce progresso tecnologico aumentava i rischi di obsolescenza di molti prodotti, repentinamente superati da altri tecnologicamente più avanzati. A questi due fattori si deve aggiungere la continua richiesta da parte dei clienti di un miglioramento dei servizi e la necessità aziendale di ridurre i costi.

Adottare questa nuova filosofia di gestione significa pianificare e programmare minuziosamente, in relazione alle diverse caratteristiche ed esigenze di ogni azienda, ognuna delle quattro classi di attività precedentemente elencate, con l'obiettivo di contribuire alla realizzazione delle cosiddette 7 condizioni giuste della logistica:

1. Il prodotto giusto;
2. Nella quantità giusta;
3. Nella condizione giusta;
4. Nel luogo giusto;
5. Al momento giusto;
6. Al cliente giusto;
7. Al costo giusto.

Le problematiche da affrontare in materia di gestione di magazzino si differenziano in problemi di *design/progettazione* e problemi operativi. Di seguito viene proposta una rappresentazione tabellare dei diversi problemi per come essa è conosciuta nella letteratura scientifica.

Tipologia	Problema		Decisioni
Warehouse Design	Overall structure		Flusso dei materiali
			Identificazione delle aree
			Localizzazione delle aree
	Sizing and dimensioning		Dimensione del magazzino
			Dimensione delle aree
	Department layout		Numero, lunghezza e larghezza dei corridoi
			Orientamento dei corridoi
			Numero e localizzazione delle porte di I/O
	Equipment selection		Livello di automazione
			Attrezzatura per lo stoccaggio
Attrezzatura per la movimentazione			
Operation strategy		Strategia per lo stoccaggio	
		Strategia per l'order picking	
Warehouse Operation	Receiving and shipping		Truck-dock assignment
			Order-Truck assignment
			Scheduling delle spedizioni
	Storage	SKU-department assignment	Assegnamento degli item alle differenti aree di magazzino
			Allocazione dello spazio
		Zoning	Assegnamento delle SKU alle zone
	Order picking	Storage location assignment	Assegnamento degli addetti ai prelievi alle zone
		Batching	Assegnamento delle posizioni di stoccaggio
			Assegnamento delle classi di stoccaggio
	Routing		Dimensione delle partite
		Assegnamento degli ordini alle partite	
		Sequenza di prelievo	
Sorting		Selezione del Dwell point (per i sistemi AS/RS)	
		Order-lane assignment	

Tabella 1 - *Design and operation problems* [17]

### 2.2.1 Ricezione e spedizione

La ricezione e la spedizione sono gli estremi del *continuum* rappresentante le operazioni che avvengono all'interno di un magazzino. I problemi e le decisioni da prendere in tale ambito in realtà non sono molte, così come non sono molti i contributi presenti in letteratura su questa tematica.

Il primo aspetto di cui preoccuparsi è quello di assegnare correttamente i mezzi di trasporto, sia in entrata che in uscita, alle varie porte di I/O delle quali si è deciso di dotare il magazzino. Tale decisione è molto importante in quanto determina i flussi di materiali che scorrono all'interno del magazzino. Fatto ciò, bisogna schedare le varie forniture e le varie spedizioni così come avviene nel classico problema dello *scheduling*, in cui le macchine sono rappresentate dalle porte di I/O e i *jobs* che devono essere processati sono rappresentati proprio dai mezzi di trasporto dai/sui quali deve essere scaricata/caricata la merce. Naturalmente, il *manager* che si occupa di questi aspetti, nel cercare una soluzione, deve far leva su quella base di conoscenza in cui sono presenti informazioni come il tempo di arrivo previsto delle forniture, la tipologia di merce ordinata, il tasso di domanda del mercato, il *layout* del magazzino e le risorse dedicate alla movimentazione della merce, così come deve tener conto dei vincoli che inevitabilmente complicano il problema: risorse e attrezzature limitate, livello di servizio al cliente e dunque tempo di fornitura e tempo per le operazioni di carico e scarico merce, finestre temporali in cui il personale del magazzino è impegnato nello scarico e carico merci, numero di porte e loro disposizione lungo il perimetro del magazzino, politica di gestione delle forniture e delle spedizioni, ecc.

In letteratura esistono veramente pochi modelli formali per l'ottimizzazione delle operazioni di ricezione e spedizione. Tra questi, la maggior parte, se non quasi la totalità, tratta il tema dell'assegnamento dei mezzi alle porte I/O di un magazzino di *cross-docking*, ossia un magazzino in cui la merce che viene scaricata da un autotreno viene successivamente caricata in un altro autotreno per essere spedita al cliente, bypassando in tal modo la fase di stoccaggio.

### 2.2.2 Immagazzinamento

La fase dello stoccaggio insieme a quella del prelievo costituiscono il cuore dell'attività che viene svolta all'interno di un magazzino e, pertanto, sono anche le fasi che detengono la quota parte maggiore dei costi totali sostenuti per la gestione di un magazzino. Una loro intelligente progettazione e pianificazione può ridurre notevolmente i costi operativi e costituire un vantaggio competitivo per l'impresa.

Le problematiche che rivestono maggiore rilievo riguardano la gestione delle scorte e l'allocazione delle posizioni di stoccaggio. Con la gestione delle scorte in pratica si intende determinare la

quantità ottimale di scorta da tenere in magazzino per ogni SKU e la frequenza con la quale provvedere all'approvvigionamento delle stesse. L'allocazione delle posizioni di stoccaggio invece ha come obiettivo lo smistamento delle diverse SKU nelle varie aree di magazzino e nelle varie zone. In sintesi, la gestione delle scorte risponde alle domande del cosa stoccare e in che quantità mentre l'allocazione delle posizioni risponde al dove stoccare quanto determinato dalla gestione delle scorte.

A testimonianza della loro importanza si evidenziano le attività di ricerca scientifica (così come riportate nelle sezioni successive) tese alla riduzione dei livelli di inventario pur mantenendo un adeguato livello di servizio al cliente e di ridurre i costi legati alla movimentazione dei materiali all'interno del magazzino.

### **2.2.3 Prelievo degli ordini**

L'*order picking* è il processo di recupero dei prodotti dall'area di stoccaggio in risposta ad una specifica richiesta del cliente.

Esso è stato identificato come l'operazione più costosa e più *labour-intensive* (nei magazzini con sistemi manuali) e *capital-intensive* (nei magazzini con sistemi automatizzati) nella maggior parte dei magazzini [12], [6], [45]; infatti, rappresenta più del 55% dei costi totali di un magazzino e pertanto una sua qualche inefficienza comporta un insoddisfacente livello di servizio offerto ai clienti ed elevati costi operativi per il magazzino e, conseguentemente, per l'intera *supply chain*. Per tali ragioni, le figure professionali che operano nel settore, siano essi *manager* o ricercatori, considerano l'*order picking* come l'area d'intervento primaria per ottenere miglioramenti in termini di produttività.

L'obiettivo principale del processo di prelievo degli ordini è la massimizzazione del livello di servizio soggetta a vincoli quali il lavoro, il capitale e le macchine [12]. Il livello di servizio è composto da una serie di fattori quali la media e la variazione del tempo di consegna, l'integrità e la precisione dell'ordine. Minore è il tempo di prelievo degli articoli contenuti nell'ordine, minore sarà il tempo per rendere disponibile la spedizione al cliente; non solo, un contenuto tempo di prelievo garantisce anche un'elevata flessibilità nella gestione delle modifiche in fase avanzata negli ordini. Ridurre al minimo il tempo di recupero dell'ordine (il tempo di prelievo delle SKU che compongono l'ordine) è, quindi, la necessità di qualsiasi sistema di *order picking*.

Esistono diversi modi di condurre il processo di prelievo; tuttavia si può affermare che in quasi tutti i metodi di prelievo vengono seguiti tre *step* basilari: la lottizzazione (*batching*), l'instradamento e il sequenziamento (*routing and sequencing*), l'ordinamento (*sorting*).

Il *batching* consiste nel partizionare un dato insieme di ordini in modo tale da ottenere diverse liste di prelievo che vengono assegnate ai vari addetti al prelievo. L'obiettivo è quello di prelevare e spedire tutti gli oggetti all'interno delle finestre temporali stabilite bilanciando il lavoro svolto dai diversi *picker* e minimizzando il tempo dedicato alle operazioni di prelievo.

La decisione di *routing* si sostanzia nella determinazione del miglior percorso da seguire nel prelevamento dei vari oggetti. L'obiettivo naturalmente è quello di minimizzare i tempi di prelievo. Il problema è riconducibile al problema del commesso viaggiatore (*Traveling Salesman Problem*) in cui però assume particolare importanza il *layout* interno del magazzino, più precisamente il modo in cui sono strutturati i suoi corridoi. I contributi presenti in letteratura hanno individuato quattro classi principali di problemi: *sequencing and routing for conventional multi-parallel-aisle systems*, *sequencing and routing for man-on-board AS/RS*, *sequencing and routing for unit-load AS/RS* e infine *sequencing and routing for carousel systems*.

Il *sorting* è un'operazione richiesta nel momento in cui vengono soddisfatti diversi ordini contemporaneamente. Esistono due modalità di esecuzione che si differenziano per il momento in cui avviene tale operazione: si può ordinare i vari articoli mentre si procede al loro prelievo oppure terminare l'operazione di prelievo e successivamente procedere al riassetto dell'ordine.

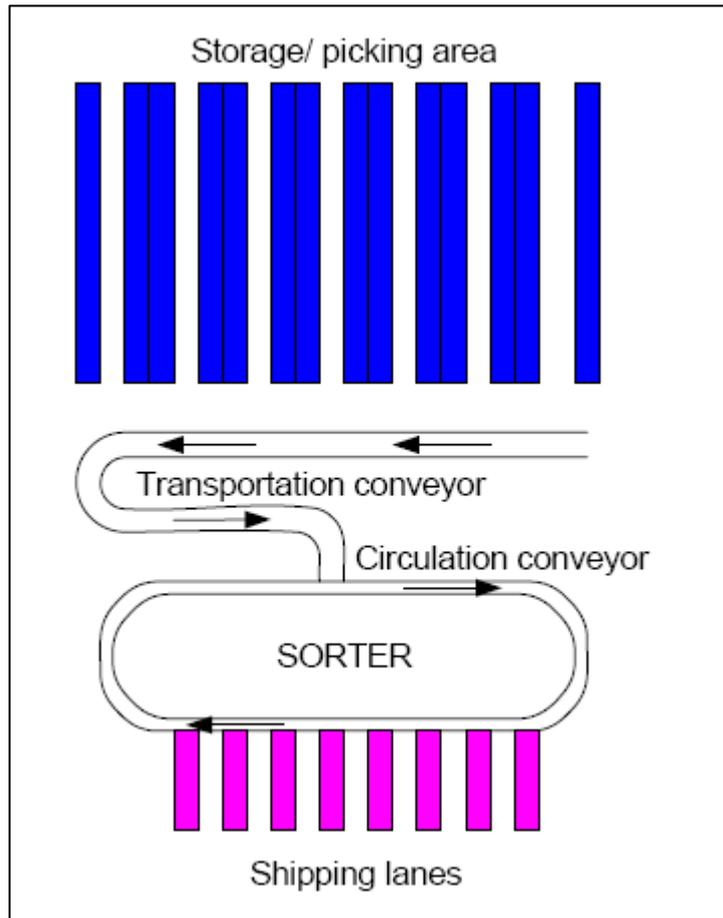


Figura 2 - Un tipico sistema di A/S (*Accumulation/Sorting*)

## 3 Il problema dell'allocazione delle merci alle posizioni di stoccaggio

Il seguente capitolo si focalizza su un problema operativo che riguarda la fase dello stoccaggio: l'allocazione dei prodotti alle posizioni di stoccaggio. In letteratura tale problema è noto come *Storage Location Assignment Problem* (SLAP).

Nel corso del capitolo viene dapprima fornita una descrizione del suddetto problema, con una panoramica su quanto presente nella letteratura scientifica di settore, e in seguito viene proposta la formulazione matematica dello stesso nel caso particolare del PAP (*Product Allocation Problem*).

### 3.1 *Storage Location Assignment Problem*

Lo SLAP consiste nell'allocare i prodotti in arrivo al magazzino nelle diverse posizioni di stoccaggio presenti all'interno dell'apposita area, con l'obiettivo di minimizzare i costi di movimentazione e di massimizzare l'utilizzazione dello spazio a disposizione. Le soluzioni adottate si basano sul principio che gli articoli richiesti frequentemente devono essere collocati negli spazi più vicini alla zona di approntamento delle spedizioni affinché il tempo totale richiesto dalle attività di movimentazione sia minimo [9].

In maniera più rigorosa il problema può essere definito nel seguente modo.

Date le informazioni:

- sull'area di stoccaggio: *layout*;
- sulle posizioni di stoccaggio: numero, accessibilità, dimensioni, locazione;
- sulle SKU che devono essere immagazzinate: dimensioni, domanda, quantità, tempo di fornitura;

determinare l'allocazione ottimale dei prodotti nelle varie posizioni tenendo conto di:

- capacità di stoccaggio;
- capacità di prelievo degli addetti e efficienza di tale processo;
- tempo di risposta;
- compatibilità tra gli articoli e le posizioni di stoccaggio e compatibilità tra i prodotti;
- politica di prelievo adottata (*First In First Out - FIFO*, *Last In First Out - LIFO*).

A seconda della quantità di informazioni note, si possono distinguere tre diverse categorie di problemi che danno luogo ad altrettante tipologie di SLAP:

1. Informazioni sugli articoli;
2. Informazioni sui prodotti;
3. Nessuna informazione.

### 3.1.1 SLAP con informazioni sugli articoli

In questo tipo di problema, noto come SLAP/II, si assumono completamente note le informazioni sulle SKU che vengono trattate, in merito a tempi di arrivo e di spedizione.

La struttura del problema ricorda molto quella di un problema di assegnamento in cui però due articoli possono anche occupare la stessa posizione purché non lo facciano contemporaneamente. Lo stesso problema è stato anche definito *Vector Assignment Problem* (VAP) dal momento che l'occupazione della posizione da parte dell'articolo non è espressa attraverso una singola variabile binaria ma attraverso un vettore di variabili binarie che colleziona gli stati della posizione nei diversi periodi di tempo [11].

Questo tipo di problema è oggetto di continue ricerche in ambito accademico perché fornisce un *lower bound* in termini di costo e un *upper bound* in termini di *performance*. Tuttavia l'onere computazionale richiesto per la sua risoluzione è tutt'altro che trascurabile quando si tratta di istanze di grandi dimensioni tanto da ricorrere a metodi risolutivi di tipo euristici. Un esempio di euristica è la *Duration of Stay* (DOS) di Goetschelckx e Ratliff [13]. La DOS, calcolata per ogni unità  $i$  di una SKU con lotto di acquisto pari a  $Q$ , rappresenta la durata del soggiorno dell'*item* in magazzino ed è supposta pari a  $i/\lambda$ ,  $i = 1, \dots, Q$ , dove  $\lambda$  è il tasso di domanda della SKU. Gli oggetti che hanno la DOS minore vengono assegnati alle posizioni di stoccaggio più vicine alla porta di *output*.

### 3.1.2 SLAP con informazioni sui prodotti

Spesso avere a disposizione informazioni complete sui tempi di arrivo e di spedizione per ogni singolo articolo non è così semplice e si è costretti a lavorare con dati più generali che riguardano il prodotto. Esistono diversi approcci allo SLAP con informazioni sui prodotti (SLAP/PI):

- Politica di assegnamento casuale (*Random storage*);
- Politica di assegnamento della posizione disponibile più vicina (*Closest Open Location Storage - COLS*);
- Politica di assegnamento basata su un sistema a posizioni fisse (*Dedicated storage*);

- Politica di assegnamento basata su un sistema a classi (*Class-based storage*).

La politica di assegnamento casuale delle posizioni di stoccaggio si basa sull'ipotesi di fondo che ogni posizione di stoccaggio vuota ha la stessa probabilità di essere scelta per contenere un generico prodotto così come ogni unità di un determinato prodotto ha la stessa probabilità di essere prelevata quando tale prodotto è stoccato in più postazioni. Questa strategia di assegnamento mira all'ottimizzazione dello spazio di stoccaggio consentendo di utilizzare completamente l'area di stoccaggio.

Dato che la collocazione delle merci viene decisa dinamicamente in base all'occupazione del magazzino e alle previsioni sui futuri arrivi e prelievi, le posizioni occupate dagli articoli di un dato tipo sono variabili nel tempo. La capacità del magazzino può quindi essere stimata come

$$m_{pv} = \max_t \sum_{j=1}^n I_j(t) \quad (2.1)$$

dove  $m_{pv}$  è il numero di posizioni di stoccaggio,  $n$  sono i tipi di articoli che devono essere stoccati,  $t$  sono i periodi oggetto delle previsioni e  $I_j(t)$  è il livello di inventario del  $j$ -esimo tipo di articolo al tempo  $t$  espresso in unità convenzionali di carico.

Le criticità principali associate a tale politica si manifestano nelle operazioni di prelievo. Soprattutto nel caso di magazzini con elevate capacità di stoccaggio e con un elevato numero di SKU da trattare, la politica *random* non permette agli addetti al prelievo di “memorizzare” dove ogni prodotto è localizzato. Pertanto, l'adozione di una tale politica porta con sé l'onere di un efficace ed efficiente sistema *computer-based* di gestione del magazzino che di volta in volta guida gli addetti al prelievo nelle operazioni di loro competenza.

La politica COLS è quella più semplice. Consiste nel sistemare il prodotto nella prima posizione disponibile. Se gli stessi addetti al prelievo potessero scegliere la posizione in cui porre il prodotto, probabilmente essi sceglierebbero la prima posizione a disposizione dando luogo quindi proprio ad un sistema di tipo COLS. Questo porta ad avere magazzini con scaffali pieni nella zona del magazzino più vicina alla porta di *input*, e un'area posteriore gradualmente più vuota [40]. Nonostante una inefficiente gestione degli spazi, Schwarz [44] e altri autori hanno mostrato che la strategia COLS da gli stessi risultati di una strategia *random* abbinata ad una politica FCFS (*First Come, First Served*) per depositi e recuperi.

In un sistema a posizioni fisse ciascun tipo di prodotto è assegnato a un insieme prestabilito di punti di stoccaggio. A differenza della politica casuale, di cui ne costituisce esattamente l'antitesi, la *dedicated storage* permette agli addetti alle operazioni di *picking* di memorizzare le posizioni degli articoli e quindi sfruttare il fattore esperienza per ottimizzare le operazioni di prelievo. Quindi è

sicuramente più semplice da gestire ma in termini di sfruttamento dello spazio è altrettanto meno efficiente della sua opposta, in quanto comporta un sottoutilizzo della capacità a disposizione poiché lo spazio totale richiesto è pari alla somma dei massimi spazi necessari nel tempo per i diversi prodotti. Analiticamente si può tradurre questa espressione come:

$$m_{pf} = \sum_{j=1}^n \max_t I_j(t) \quad (2.2)$$

dove  $m_{pf}$  è il numero di posizioni di stoccaggio,  $n$  sono i tipi di articoli che devono essere stoccati,  $t$  sono i periodi oggetto delle previsioni e  $I_j(t)$  è il livello di inventario del  $j$ -esimo tipo di articolo al tempo  $t$ .

L'approccio *Class-based* è sicuramente il più presente e discusso in letteratura, nonché quello a cui si attribuiscono le migliori *performance*. La logica che sta dietro questa strategia consiste nel categorizzare i prodotti, ossia suddividerli in classi basandosi su un criterio predeterminato, stabilire un *ranking* tra queste e assegnare a ciascuna classe un insieme di aree nelle quali le merci sono immagazzinate in qualsivoglia posizione. Le aree più vantaggiose vengono assegnate alle classi più critiche e via via si procede fino a quando tutte le classi non sono state allocate. Tale politica può essere equivalente sia alla casuale che alla fissa a seconda del numero di classi che vengono composte: se il numero delle classi è pari al numero di prodotti allora il sistema a classi si riduce al sistema a posizioni fisse, mentre se vi è una sola classe il sistema a classi si riduce al sistema a posizioni variabili.

Il fattore di maggior criticità nell'applicazione di questa politica di allocazione è la scelta del criterio da seguire per effettuare la classificazione dei prodotti. La scelta non è univoca ed è proprio su questo punto che si differenziano i vari contributi scientifici. I principali criteri di discriminazione sono basati sulla valutazione dei seguenti parametri:

- *Popularity* [25], [10]: indica il numero di richieste pervenute per una determinata referenza e, in definitiva, può essere interpretato come il numero di volte che l'addetto al prelievo deve visitare le posizioni dove una determinata referenza viene stoccata. Come indicano molte analisi del settore questa è la scelta che riscuote il maggiore successo, anche all'interno di realtà aziendali molto differenti tra loro;
- *Turnover* [19], [14], [44], [23]: in questo caso si utilizza come criterio discriminante per la classificazione dei prodotti la quantità totale di una specifica referenza spedita durante un

dato periodo di tempo. I prodotti con il tasso di *turnover* più elevato, detti anche *fast moving products*, sono collocati nelle locazioni di stoccaggio più vicine al punto di ingresso/uscita del magazzino. Al contrario i prodotti con un tasso di *turnover* minore, detti anche *slow moving products*, vengono collocati negli *slot* più lontani da tale punto, e quindi considerati planimetricamente più svantaggiati. Un problema generalmente riconosciuto nell'adozione di questo parametro come indicatore per la formazione delle classi di prodotto è rappresentato dalle possibili fluttuazioni nella domanda dei diversi prodotti, dovute ad esempio a fattori stagionali. A tali variazioni dovrebbe corrispondere un continuo cambiamento dell'allocazione dei diversi prodotti all'interno del magazzino, causando un numero elevato di movimentazioni interne non strettamente necessarie all'evasione degli ordini ricevuti, ed un conseguente incremento dei costi. E' inoltre importante notare come questa strategia richieda un notevole sforzo dal punto di vista del reperimento di informazioni per riuscire a massimizzare i vantaggi derivanti dalla sua applicazione;

- *Volume* [15]: questa scelta consiste nel considerare come parametro discriminante, nella formazione delle diverse classi, il prodotto tra la domanda specifica attesa in un determinato periodo di tempo e il volume occupato dalla referenza considerata. Così come sottolineato per la strategia precedente per l'implementazione di questo approccio è richiesto un notevole sforzo dal punto di vista del reperimento di informazioni;
- *Pick density* [8]: consiste nell'effettuare una graduatoria delle referenze considerando il rapporto tra la popolarità della referenza e il volume occupato dalla stessa. Questo parametro in definitiva evidenzia quali sono le referenze con il maggior tasso di prelievo in rapporto al volume loro dedicato nella scaffalatura;
- *COI* [21], [22], [30], [18], [26], [31], [3]: il coefficiente *COI* (*Cube per Order Index*) è ottenuto come rapporto tra il volume occupato dalla referenza in questione e il suo valore di popolarità. I prodotti con il minor valore di *COI*, caratterizzati quindi da una giacenza bassa rispetto alla richiesta, saranno posizionati il più vicino possibile al punto di I/O. Al contrario, ovviamente, i prodotti caratterizzati da valori dell'indice *COI* superiori saranno collocati in posizioni più svantaggiate. Il successo riscosso da questo indice nella letteratura scientifica di settore si giustifica soprattutto attraverso le numerose dimostrazioni riguardo la sua capacità di minimizzare i costi di prelievo; tuttavia anche in questo caso lo sforzo dal punto di vista del reperimento delle informazioni necessarie alla strutturazione della strategia risulta notevole.

Oltre a questi parametri esiste una filosofia organizzativa che invece prende in esame la familiarità tra i prodotti e che si sostanzia nella creazione di *cluster* composti da articoli che presentano una forte correlazione o similarità tra loro. Anche qui si possono distinguere diversi modi di misurare la similarità tra prodotti; Ballou [1], ad esempio, individua tre criteri: complementarietà, compatibilità e popolarità. La complementarietà consiste nel posizionare vicino articoli che vengono ordinati solitamente insieme. La compatibilità è il criterio per il quale vengono posizionati vicino prodotti che presentano gli stessi requisiti di stoccaggio; ad esempio, non è possibile stoccare in posizioni limitrofe articoli appartenenti alla catena del fresco e articoli non appartenenti a questa categoria in quanto le esigenze di stoccaggio sono totalmente differenti. La popolarità sottolinea che i prodotti hanno tassi di *turnover* differenti all'interno di un sistema di stoccaggio e, pertanto, quelli che richiedono un gran numero di viaggi devono essere allocati in posizioni vicino alla porta di I/O.

De Koster [5] individua, invece, due differenti approcci al problema: il metodo basato sulla complementarietà e il metodo basato sul contatto. Il primo metodo consiste nel creare *cluster* di prodotti basati sul grado di complementarietà della domanda, analizzando, ad esempio, la matrice delle correlazioni tra i prodotti/clienti che indica rispettivamente con che frequenza due prodotti compaiono insieme in un ordine oppure con che frequenza due clienti distinti ordinano gli stessi prodotti. Successivamente i prodotti appartenenti allo stesso *cluster* vengono posizionati il più vicino possibile. Con il secondo metodo il raggruppamento avviene utilizzando come parametro discriminante la frequenza con cui due generici prodotti vengono "in contatto". La frequenza di contatto tra un prodotto  $i$  e un prodotto  $j$  è definita come il numero di volte che l'addetto preleva il prodotto  $j$  immediatamente dopo aver prelevato il prodotto  $i$  o viceversa.

Un altro modo, classico, di definire le classi è rappresentato dall'analisi di Pareto, nota anche come legge 80/20 o analisi ABC, basata sul criterio del *turnover*.

Il passo successivo alla definizione delle classi è la loro allocazione nelle zone di stoccaggio. Solitamente vengono identificati cinque criteri fondamentali attraverso cui è possibile suddividere il magazzino:

- **Suddivisione diagonale (*Diagonal*)** [10], [45]: questo tipo di suddivisione colloca i prodotti appartenenti ad una stessa classe su linee, ciascuna formata da punti tutti equidistanti dalla porta di I/O, poste a distanze crescenti da tale punto. Questo principio genera una suddivisione del magazzino tramite linee diagonali che separano una classe dall'altra. Naturalmente la suddivisione cambia a seconda che la porta di I/O sia al centro, nel corridoio frontale, o in una dei due angoli del magazzino.

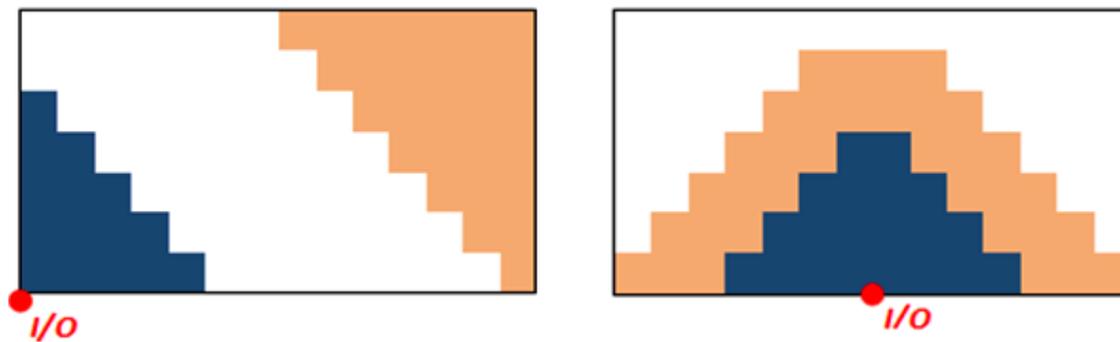


Figura 3 - Suddivisione diagonale con punto di I/O collocato in posizione angolare e in posizione centrale

- Suddivisione in corridoi (*Within-Aisle*) [25]: questa tecnica è forse la più semplice da implementare poiché la suddivisione nel magazzino viene effettuata utilizzando elementi già presenti nel magazzino stesso: i corridoi. In particolare, una volta individuata la collocazione del punto di I/O basterà allocare la classe A nei corridoi immediatamente nelle vicinanze di tale punto e le classi inferiori in quelli più distanti. In questo modo se il punto di I/O si trova al centro del fronte del magazzino i corridoi assegnati alla classe privilegiata saranno quelli centrali mentre, se si trova in posizione angolare i corridoi scelti saranno o quelli all'estrema destra o quelli all'estrema sinistra. In letteratura, numerosi studiosi affermano che sia questa la scelta che genera il maggiore risparmio: mediamente tra il 3% e il 6% meglio rispetto all'alternativa diagonale e tra il 9% e il 20% circa rispetto all'alternativa trasversale [37], [38], [36], [35], [39].

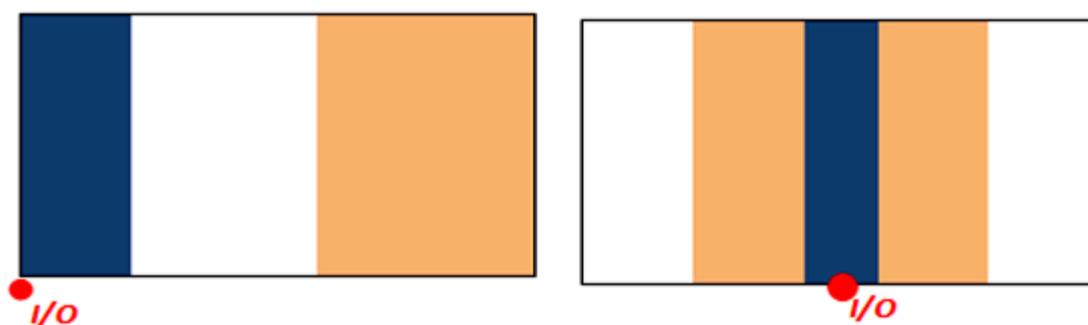


Figura 4 - Suddivisione *Within-Aisle* con punto di I/O in posizione angolare e in posizione centrale

- Suddivisione trasversale ai corridoi (*Across-Aisle*) [36]: il magazzino viene stratificato tramite linee immaginarie perpendicolari al senso di percorrenza dei corridoi. In questo modo i prodotti afferenti alle classi privilegiate verranno collocati nelle prime posizioni di

ogni scaffalatura a partire da quelle più vicino al corridoio di testa e ci si allontanerà a mano a mano che i prodotti apparterranno a classi inferiori. Implementando questa logica si avranno all'interno di ogni scaffalatura prodotti appartenenti a tutte le classi.

Questa strategia non è sensibile alla posizione della porta di I/O e genera una suddivisione equivalente sia che questa sia in centro, sia che si trovi ai lati del magazzino. Le-Duc e De Koster [28], utilizzando una *return routing policy*, ottengono una soluzione prossima alla soluzione ottima per una tale suddivisione e, lo stesso Le-Duc [29], estende questo risultato ad altre politiche di instradamento.

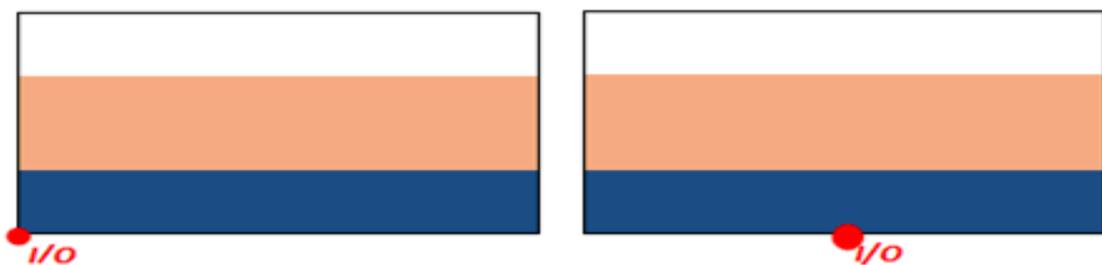


Figura 5 - Suddivisione *Across-Aisle* con punto di I/O in posizione angolare e in posizione centrale

- Suddivisione perimetrale (*Perimeter*) [36]: questa strategia è forse quella che trova il minor numero di conferme in letteratura e si basa sulla collocazione dei prodotti appartenenti alle classi più critiche lungo il perimetro del magazzino. A mano a mano che le classi diventano meno critiche, i prodotti verranno allocati sempre più all'interno del magazzino stesso;

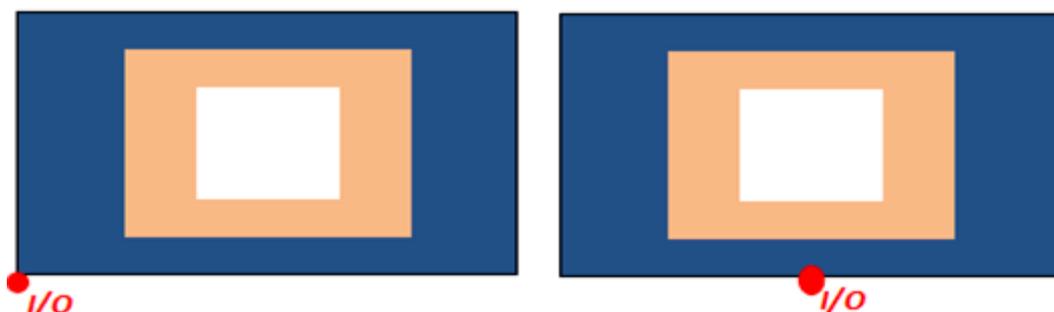
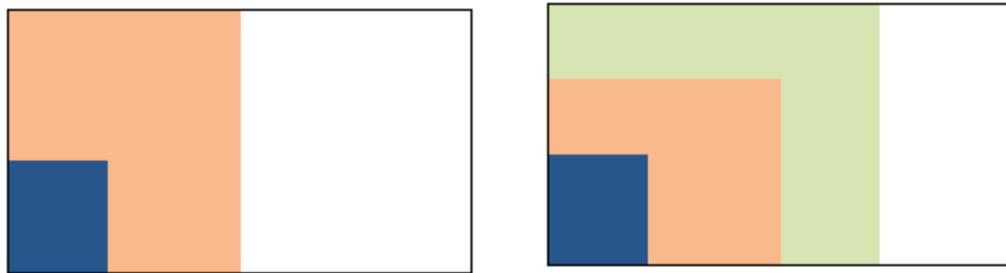


Figura 6 - Suddivisione *Perimeter* con punto di I/O in posizione angolare e in posizione centrale

- Suddivisione a L (*L-Shape*) [14], [46]: l'ultima strategia qui esposta rientra tra le cosiddette *rectangular-based*. Dato un numero di classi in cui suddividere le diverse referenze, l'approccio sarà simile a quello diagonale ma la distanza tra il punto di I/O e i diversi punti di stoccaggio sarà calcolata utilizzando appunto le distanze rettangolari, altrimenti dette

distanze di Chebyshev, in cui il percorso da seguire per muoversi da un punto ad un altro del magazzino sarà formato unicamente da movimenti trasversali e/o longitudinali. In questo modo la suddivisione in classi che ne scaturisce non sarà più per linee oblique ma per aree rettangolari che saranno ovviamente tante quante le classi in cui si vogliono suddividere le referenze. La classe più critica sarà quella che si troverà in immediato contatto con il punto di I/O mentre le classi “minori” si troveranno via via all'esterno di tale zona. Malgrado la suddivisione a L e la suddivisione diagonale siano relativamente simili concettualmente, diverse ricerche hanno evidenziato un vantaggio in termini di tempo di viaggio per le operazioni di prelievo dell'ordine del 5%.



**Figura 7 - Suddivisione L-Shape a tre e quattro classi**

### 3.1.3 SLAP senza informazioni

In questo caso, il pianificatore non ha a disposizione nessuna tipologia di informazioni ed il problema è anche conosciuto con l'acronimo di SLAP/NI. Le politiche di allocazione utilizzate in questo caso sono le seguenti:

1. *Closest Open Location (COL)*;
2. *Farthest Open Location (FOL)*;
3. *Random (RAN)*;
4. *Longest Open Location (LOL)*.

Le prime due strategie utilizzano, come criterio di scelta della posizione da utilizzare, la distanza della cella rispetto alla porta di ingresso delle merci; la *random* agisce in maniera casuale, mentre la LOL sceglie la posizione che risulta essere da più tempo inutilizzata.

Tutte le ricerche riguardanti lo SLAP assumono che il lotto di acquisto degli articoli trattati sia un'informazione nota a priori. Tuttavia, Wilson [51] ha dimostrato che il problema del dimensionamento del lotto (*Lot Sizing*) e lo SLAP possono anche essere considerati simultaneamente con l'obiettivo di ottenere un costo totale ottimale che include sia i costi di inventario che i costi di movimentazione.

### 3.2 Product Allocation Problem

Il problema che riveste maggiore interesse in letteratura è il problema di allocazione con informazioni sui prodotti, anche conosciuto con l'acronimo PAP.

Il PAP è un problema di carattere tattico che nella sua variante più generale possibile può essere espresso prendendo a riferimento un sistema a posizioni fisse in cui  $m_{pf}$  indica il numero complessivo di punti di stoccaggio. Si indichi inoltre con [9]:

- $n$ : il numero di tipi di prodotto (o classi) presenti in magazzino;
- $m_j$ : il numero di punti di stoccaggio destinati al prodotto  $j$ ,  $j = 1, \dots, n$ ; naturalmente bisogna ricordare che in un sistema a posizioni fisse vale la relazione  $\sum_{j=1}^n m_j \leq m_{pf}$ ;
- $R$ : il numero di punti di ingresso/uscita del magazzino;
- $p_{jr}$ : il numero medio giornaliero di operazioni di movimentazione in ingresso/uscita da  $r$  riguardanti il prodotto  $j$ ,  $j = 1, \dots, n$ ,  $r = 1, \dots, R$ ;
- $t_{rk}$ : il tempo di spostamento dal punto di ingresso/uscita  $r$  al punto di stoccaggio  $k$ ,  $r = 1, \dots, R$ ,  $k = 1, \dots, m_{pf}$ ;
- $c_{jk}$ : il costo di assegnamento della posizione di stoccaggio  $k$  al prodotto  $j$ ,  $j = 1, \dots, n$ ,  $k = 1, \dots, m_{pf}$ ;

Nell'ipotesi che tutti i punti di stoccaggio utilizzati abbiano un identico tasso di utilizzazione, è possibile stimare tale costo di assegnamento come:

$$c_{jk} = \sum_{r=1}^R \frac{p_{jr}}{m_j} t_{rk} \quad (2.28)$$

dove  $\frac{p_{jr}}{m_j}$  rappresenta il numero medio giornaliero di operazioni di movimentazione del prodotto  $j$  tra il punto di I/O  $r$  e la generica posizione di stoccaggio assegnata a tale prodotto e, conseguentemente,  $\frac{p_{jr}}{m_j} t_{rk}$  costituisce il tempo medio di spostamento imputabile al punto di stoccaggio  $k$  qualora quest'ultimo venga assegnato a prodotti di tipo  $j$ .

Quelli esplicitati finora costituiscono i parametri in *input* del modello matematico che rappresenta il problema considerato. Per quanto riguarda invece le variabili decisionali, indichiamo con:

- $x_{jk}$ : la variabile decisionale, di tipo binario, pari a 1 se il punto di stoccaggio  $k$  è assegnato al prodotto di tipo  $j$ , 0 altrimenti,  $j = 1, \dots, n$ ,  $k = 1, \dots, m_{pf}$ .

Il problema può essere schematizzato nel modo seguente:

$$\min_x \sum_{j=1}^n \sum_{k=1}^{m_{pf}} c_{jk} x_{jk} \quad (2.29)$$

s.a.

$$\sum_{k=1}^{m_{pf}} x_{jk} = m_j, \quad \forall j = 1, \dots, n \quad (2.30)$$

$$\sum_{j=1}^n x_{jk} \leq 1, \quad \forall k = 1, \dots, m_{pf} \quad (2.31)$$

$$x_{jk} \in \{0,1\}, \quad \forall j = 1, \dots, n, k = 1, \dots, m_{pf} \quad (2.32)$$

La funzione obiettivo è, dunque, la minimizzazione del costo totale di assegnamento, che non è altro che il tempo totale di movimentazione dei prodotti presenti in magazzino. La quantità  $\sum_{k=1}^{m_{pf}} x_{jk}$  rappresenta il numero di posizioni di stoccaggio assegnate al  $j$ -esimo prodotto. Pertanto il gruppo di vincoli (2.30) garantisce che per ogni prodotto  $j$  il numero di posizioni che gli vengano attribuite sia esattamente pari a  $m_j$ , ossia al numero di punti di stoccaggio destinate al prodotto  $j$ . I vincoli (2.31) servono per assicurare che uno stesso *slot* non sia assegnato a più di un prodotto. Siccome il sistema a posizione fisse porta con se la possibilità di una sottoutilizzazione della capacità di stoccaggio a disposizione, un generico *slot* può anche rimanere vuoto ma, nel momento in cui viene assegnato, esso non può ospitare più di una tipologia/classe di prodotti. Infine, i vincoli (2.32) rappresentano i vincoli di binarietà della variabile decisionale.

Tale problema è un problema di programmazione binaria che non può essere risolto direttamente. Tuttavia, notando la particolare struttura dell'insieme dei vincoli (integrità del vettore dei termini noti e unimodularità della matrice dei vincoli), si possono rilassare, senza perdere l'integrità della soluzione, i vincoli (2.32) con i vincoli  $0 \leq x_{jk} \leq 1$  ottenendo così un problema di programmazione lineare. In realtà si può optare direttamente per un rilassamento continuo inserendo al posto dei vincoli (2.32) i vincoli di non negatività  $x_{jk} \geq 0$  che abbinati alla presenza dei vincoli (2.31) assicurano che gli unici valori possibili per le variabili decisionali siano 0 e 1. La matrice dei vincoli a componenti intere è unimodulare e il vettore delle risorse a componenti intere rappresentano una condizione sufficiente che permette di trovare la soluzione ottima del problema andando a calcolare la soluzione ottima di base del problema di PL associato, dal momento che ogni soluzione ammissibile di base (e quindi anche quella ottima) avrà componenti intere.

Il problema di assegnazione degli spazi è un problema altamente degenero pertanto è sconsigliata una sua risoluzione tramite il metodo del semplice.

Una sua semplificazione la si ottiene nel caso di magazzini che presentano un unico punto di ingresso/uscita. In questo caso, una proposta di procedura euristica è di seguito riportata.

Nel caso in cui il magazzino sia dotato di un unico punto di ingresso/uscita la soluzione del problema di assegnazione degli spazi può essere ulteriormente semplificata. Ciò dipende dal fatto che i vari costi di assegnamento di una generica posizione di stoccaggio ad una generica tipologia di prodotto assumono la seguente forma:

$$c_{jk} = \frac{p_{j1}}{m_j} t_{1k} = a_j b_k, \quad \forall j = 1, \dots, n, \quad k = 1, \dots, m_{pf} \quad (2.33)$$

in cui  $a_j = \frac{p_{j1}}{m_j}$  dipende solo dal prodotto  $j$ , mentre  $b_k = t_{1k}$  è funzione solo del punto di stoccaggio  $k$ .

La soluzione ottima del problema può essere determinata utilizzando il seguente schema [9]:

**Passo 1.** Costruisci il vettore  $\alpha$  di  $\sum_{j=1}^n m_j$  componenti, ottenuto realizzando  $m_j$  copie dello scalare  $a_j$ , per ogni  $j = 1, \dots, n$ . Ordina il vettore  $\alpha$  per valori non crescenti delle sue componenti. Sia  $\sigma_\alpha(i) = j$  l'indice relativo allo scalare  $a_j$ ,  $j = 1, \dots, n$ , corrispondente a  $\alpha_i$ ,  $i = 1, \dots, \sum_{j=1}^n m_j$  (gli indici  $\sigma_\alpha(i)$  consentono quindi di "risalire" alla corrispondenza esistente tra le componenti del vettore  $\alpha$  e le quantità  $a_j$ ).

**Passo 2.** Sia  $b$  il vettore di  $m_{pf}$  componenti, ciascuna delle quali corrispondente allo scalare  $b_k$ ,  $k = 1, \dots, m_{pf}$ . Ordina il vettore  $b$  per valori non decrescenti delle sue componenti. Sia  $\beta$  il vettore di  $\sum_{j=1}^n m_j$  componenti, ottenuto considerando le prime  $\sum_{j=1}^n m_j$  componenti del vettore  $b$  ordinato. Sia  $\sigma_\beta(i) = k$  l'indice relativo allo scalare  $b_k$ ,  $k = 1, \dots, m_{pf}$ , corrispondente a  $\beta_i$ ,  $i = 1, \dots, \sum_{j=1}^n m_j$  (gli indici  $\sigma_\beta(i)$  consentono di "risalire" alla corrispondenza esistente tra le componenti del vettore  $\beta$  e le quantità  $b_k$ ).

**Passo 3.** Determina la soluzione ottima del problema ponendo:

$$x_{\sigma_\alpha(i)\sigma_\beta(i)}^* = 1, \quad i = 1, \dots, \sum_{j=1}^n m_j$$

$$x_{jk}^* = 0, \quad \text{per tutte le rimanenti componenti.}$$

La procedura illustrata trae il suo fondamento teorico dall'osservazione che la minimizzazione del prodotto scalare di due generici vettori si ottiene ordinando un vettore per valori non crescenti e l'altro per valori non decrescenti.

### **3.3 Contributi presenti in letteratura per il PAP**

L'allocazione dei prodotti alle posizioni di stoccaggio è un problema che trova riscontro in un numero limitato di lavori nella letteratura scientifica di settore. Un buon numero di lavori presentano modelli che cercano di risolvere il problema congiuntamente con altri obiettivi quali ad esempio la determinazione del *layout* o il dimensionamento della zona di stoccaggio.

Tra gli autori che hanno proposto modelli e metodi per il problema dell'allocazione si pone in evidenza, per il loro contributo, Sanei, Nasiri, Marjani e Moattar Husseini [43] che nel loro articolo presentano un modello di allocazione degli spazi dal quale ha preso ispirazione questo lavoro di tesi; Muppani e Adil [32] che hanno proposto un modello di formazione delle classi e di allocazione denominato CFAM (*Class Formation and Allocation Model*); Larson, March e Kusiak [27] che hanno sviluppato un approccio euristico per la determinazione del *layout* di un magazzino basato su un sistema a classi in cui la terza fase riguarda proprio l'allocazione dello spazio alle varie classi; Van den Berg [47] che ha sviluppato un algoritmo polinomiale di programmazione dinamica che minimizza il tempo medio di immagazzinamento/prelievo; Önüt, Tuzkaya e Doğaç [34] i quali hanno ideato un modello e relativo algoritmo di risoluzione per la progettazione del *layout* di un magazzino multi-livello e l'allocazione dei prodotti nei vari ripiani, e infine Heragu, Du, Mantel e Schuur [20] che hanno proposto un modello matematico, e un algoritmo euristico, che congiuntamente determina la dimensione delle diverse aree e l'allocazione dei prodotti in quest'ultime.

Di seguito viene proposto lo studio di Önüt, Tuzkaya e Doğaç perché tratta un aspetto caratterizzante il presente lavoro di tesi (il multi-livello) e, successivamente viene illustrato in maniera dettagliata il lavoro di Sanei, Nasiri, Marjani e Moattar Husseini.

Lo studio di Önüt e gli altri è un'estensione del lavoro di Bassan, Roll e Rosenblatt [2] i quali hanno proposto un modello per la determinazione del *layout* interno di un magazzino considerando articoli omogenei caratterizzati dalla stessa probabilità di essere stoccati o prelevati in una struttura rettangolare 2D. Nello studio di Önüt gli articoli sono eterogenei ed inoltre viene presa in considerazione anche la terza dimensione e quindi la possibilità di stoccare e prelevare articoli anche lungo quest'ultima.

Il *throughput* annuale del magazzino è classificato in tre gruppi, A, B e C, in base al loro *turnover*. Sulla base del *turnover* e ai periodi di stoccaggio degli articoli, le probabilità di prelevare/stoccare ordini appartenenti alle classi A, B e C sono indicate rispettivamente con  $P_A$ ,  $P_B$  e  $P_C$ . La principale motivazione della suddivisione in classi degli articoli trattati risiede nell'intenzione di allocare i

prodotti nel magazzino in base alla vicinanza rispetto alla baia. Ciò che viene dato come parametri in *input* a tale modello sono la capacità totale del magazzino, il *throughput* annuale, lo spazio di stoccaggio riservato a ciascuna classe e la lunghezza dei corridoi e dei ripiani.

Altri assunti fondamentali riguardano la posizione del punto di I/O che è centrale, in modo tale che la distanza tra esso e le pareti dei due estremi sia  $u/2$ , e l'uguale probabilità di trasportare un *item* sulla parte destra oppure sulla parte sinistra della baia.

Di seguito è riportato il *layout* di riferimento di tale studio:

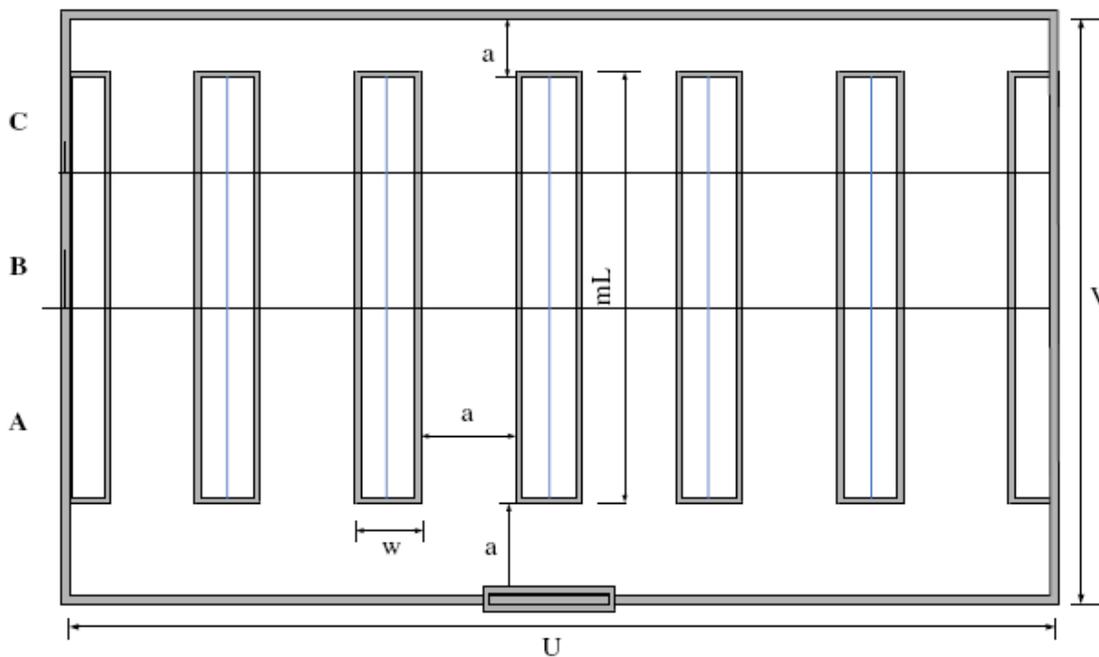


Figura 8 - *Layout* e relativi parametri [34]

L'obiettivo del modello è quello di trovare le dimensioni del magazzino che permettono di minimizzare il costo di movimentazione degli oggetti. A tal fine viene calcolata la distanza media totale percorsa nelle operazioni di stoccaggio e prelievo come somma di tre componenti, ciascuna rappresentante la distanza lungo un asse di direzione.

La distanza media lungo l'asse orizzontale è calcolata come media aritmetica tra lo spazio percorso nel caso migliore (quando la posizione di stoccaggio da raggiungere è nelle immediate vicinanze della porta di I/O) che viene assunto pari a 0 e lo spazio percorso nel caso peggiore (quando la posizione di stoccaggio da raggiungere è adiacente ad una delle pareti sull'estrema destra o sull'estrema sinistra del punto di I/O) che è pari proprio a  $u/2$ . Pertanto la distanza media lungo l'asse orizzontale è pari a

$$T_u = \frac{u}{4} \quad (2.3)$$

La distanza media lungo l'asse verticale dipende invece dall'ordine di probabilità di appartenenza dell'oggetto ad una delle tre classi individuate. Pertanto la sua espressione analitica è la seguente:

$$T_v = a + P_A \left( \frac{m_A L}{2} \right) + P_B \left( m_A L + \frac{m_B L}{2} \right) + P_C \left( m_A L + m_B L + \frac{m_C L}{2} \right) \quad (2.4)$$

Dopo opportuni passaggi e semplificazioni tale espressione diventa

$$T_v = a + \frac{m}{K} L \left[ (N_A + N_B) \left( 1 - \frac{P_A}{2} \right) + \frac{P_C}{2} (N_B + N_C) - \frac{N_B}{2} \right] \quad (2.5)$$

Infine la distanza media percorsa lungo la terza dimensione è anch'essa calcolata come media aritmetica tra lo spazio percorso nel caso migliore e lo spazio percorso nel caso peggiore.

$$T_h = \frac{h}{2} \quad (2.6)$$

La somma di questa tre distanze costituisce la distanza media totale. Moltiplicando quest'ultima per il costo unitario di movimentazione e per il *throughput* si ottiene il costo oggetto della minimizzazione:

$$C = 4dC_h \left[ a + \frac{m}{K} L \left[ (N_A + N_B) \left( 1 - \frac{P_A}{2} \right) + \frac{P_C}{2} (N_B + N_C) - \frac{N_B}{2} \right] + \frac{u}{4} + \frac{h}{2} \right] \quad (2.7)$$

Il coefficiente 4 tiene conto di questa situazione: il processo di movimentazione ha inizio quando un oggetto arriva alla porta di I/O; successivamente questo deve essere stoccato, il che vuol dire che l'addetto deve percorrere uno spazio pari a  $(T_u + T_v + T_h)$  all'andata e il medesimo spazio al ritorno. Quando lo stesso oggetto viene ordinato da un cliente, l'addetto deve provvedere al suo prelevamento e al suo spostamento verso la porta di I/O e quindi deve percorrere nuovamente 2 volte  $(T_u + T_v + T_h)$ . Pertanto lo spazio totale percorso sarà  $4(T_u + T_v + T_h)$ .

I vincoli del modello sono i seguenti:

$$K - 2mnh = 0 \quad (2.8)$$

$$m \geq 1 \quad (2.9)$$

$$n \geq 1 \quad (2.10)$$

$$h \geq 1 \quad (2.11)$$

$$u = n(w + a) \quad (2.12)$$

Il vincolo (2.8) serve per garantire il soddisfacimento della domanda annuale.  $2mnh$  è il numero totale di posizioni di stoccaggio del magazzino: ogni scaffale è caratterizzato da  $h$  livelli e ogni

livello contiene  $2m$  posizioni di stoccaggio; il numero di scaffali all'interno del magazzino è pari a  $n$ . I restanti vincoli forniscono il *design* del magazzino nelle tre dimensioni.

Le variabili decisionali pertanto sono  $m$ ,  $n$ , e  $h$ . La variabile decisionale  $n$ , come precedentemente detto, rappresenta il numero di scaffali del magazzino;  $m$  è il numero di posizioni di stoccaggio presenti in una facciata dello scaffale ( $v = 2a + mL$ ); infine  $h$  è il numero di livelli di ogni scaffale.

L'assegnamento delle SKU alle varie posizioni di stoccaggio è realizzato tenendo conto dell'ordine di probabilità di appartenenza ad una delle tre classi.

L'euristica proposta per la risoluzione di tale problema è basata sul *Particle Swarm Optimization* (PSO), una meta-euristica, proposta per la prima volta da Eberhart e Kennedy [7] e usata per la risoluzione di problemi *hard* di ottimizzazione globale.

Sanei e gli altri, invece, propongono un modello di programmazione lineare e un algoritmo euristico basato sull'algoritmo del *Branch and Bound* che risolve il problema dell'allocazione degli spazi tenendo conto di vari aspetti operativi volti principalmente a garantire che ogni tipo di prodotto sia concentrato in una determinata zona dell'area di stoccaggio al fine di semplificare la gestione delle scorte e l'utilizzazione dell'attrezzatura per la movimentazione dei prodotti.

Nel loro lavoro, gli autori assumono che le quantità da stoccare e la capacità necessaria per ogni tipo di prodotto siano informazioni note. Inoltre, prendono a riferimento un determinato *layout* costituito da  $h$  posizioni di stoccaggio in verticale (*layout multi livello*) e una porta di I/O posizionata sul lato sinistro del magazzino.

Dopo aver preso in esame diversi casi di gestione di magazzino, gli stessi autori sono arrivati a definire che la priorità per la maggior parte dei *manager* di magazzino è che se un prodotto deve essere allocato in più posizioni di stoccaggio allora è preferibile allocarlo primariamente in posizioni adiacenti, secondariamente in posizioni opposte, altrimenti in posizioni posteriori oppure, se nessuno di questi assegnamenti è fattibile, in posizioni generiche. Per tale ragione sono state definite delle penalità e l'obiettivo del modello è trovare l'allocazione ottimale dei prodotti, minimizzando la somma dei costi di penalità e dei costi di movimentazione.

## 4 Un modello matematico per il PAP in un magazzino multi-livello con vincoli di compatibilità fra i prodotti (MLPAP)

Alla base del lavoro di tesi effettuato vi è stata la formulazione del modello matematico per il problema dell'allocazione dei prodotti ai punti di stoccaggio in un magazzino multi livello e con vincoli di compatibilità fra prodotti adiacenti. Il modello proposto da Sanei [43] è stato rivistato in modo tale da cogliere alcune caratteristiche e vincoli che sono tipici di alcune realtà aziendali quali, ad esempio, i centri di distribuzione che operano nel settore della grande distribuzione organizzata. In tali contesti, infatti, l'area di stoccaggio è rappresentata da una scaffalatura multi livello e la natura dei prodotti trattati impone la gestione dei vincoli di compatibilità tra prodotti per evitare l'accostamento di prodotti tra loro non compatibili (è sufficiente pensare all'esigenza di collocare in posizioni non adiacenti i prodotti alimentari ed i prodotti per l'igiene personale o della casa).

### 4.1 Definizione del modello

Gli obiettivi che il modello matematico si propone di raggiungere sono la minimizzazione del costo totale di movimentazione e il decentramento dei prodotti a scaffale.

Di seguito sono introdotte le ipotesi di lavoro alla base della formulazione del modello matematico.

Il *layout* di riferimento è riportato in Figura 9:

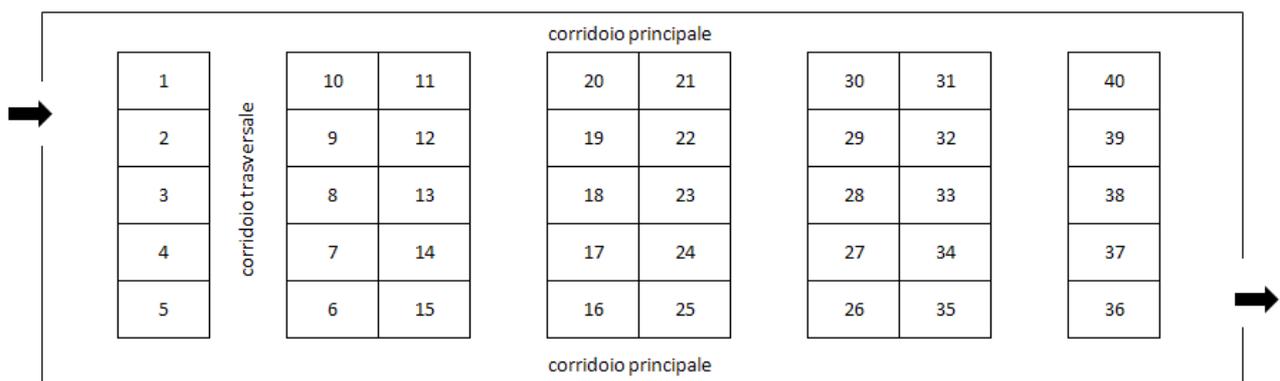


Figura 9 - *Layout* di riferimento

Il *layout* è caratterizzato da due porte di I/O che rappresentano la porta di ingresso (dove avvengono le operazioni in ingresso al magazzino) e la porta di uscita (dove avvengono le operazioni in uscita dal magazzino).

Sono presenti due corridoi principali e  $w/2$  corridoi trasversali, dove  $w$  (che nel caso in esempio è pari a 8) indica il numero di posizioni di stoccaggio presenti in linea orizzontale;  $w$  può assumere un

qualunque valore intero positivo purché rimanga sempre un numero pari in modo tale da preservare la struttura dei corridoi. Altri parametri caratteristici del *layout* sono il numero di posizioni di stoccaggio in linea verticale indicato con  $m$  (nel caso in esempio  $m=5$ ) ed il numero di livelli, indicato con  $h$ . Entrambi i parametri possono assumere un qualunque valore intero positivo senza nessun vincolo. La presenza di  $h$  livelli implica, ad esempio, che la posizione 1, in Figura 10, ha sopra di essa altre  $h-1$  posizioni di stoccaggio, così come tutti i restanti punti di stoccaggio del magazzino. Pertanto, ogni punto di stoccaggio (in seguito *slot*) è identificato da una coppia di indici  $(s,l)$  rappresentativi della posizione e del livello al quale appartiene la posizione stessa.

Infine, un altro importante aspetto riguardante il *layout*, così come introdotto in Sanei [43], riguarda la definizione delle posizioni di stoccaggio adiacenti, opposte e posteriori ad una determinata posizione. I concetti di posizione adiacente, opposta e posteriore sono stati introdotti al fine di minimizzare il decentramento dei prodotti.

Due *slot* sono adiacenti se appartengono alla stessa sezione (lato) di un corridoio trasversale e sono fisicamente contigui anche se posti su differenti livelli. Ad esempio, nel *layout* riportato in Figura 10, la posizione 1 ha come posizione adiacente la 2, così come la 2 ha come posizione adiacente la 3 e così via per tutte le posizioni tranne che per la 5 che non ha una posizione adiacente.

Due *slot* sono definiti opposti se appartengono ad uno stesso corridoio trasversale ma in sezione opposta e posizione opposta. Nel *layout* riportato in Figura 9, la posizione 1 che ha come opposto la posizione 6, la 2 che ha come opposta la posizione 7 e così via. Il concetto di *slot* opposto è esteso ovviamente al caso in cui gli *slot* sono entrambi posizionati su un livello diverso dal primo.

Due *slot* sono definiti posteriori se sono opposti ma appartengono a corridoi trasversali adiacenti come, ad esempio, la 6 che ha come posizione posteriore la 11, la 7 che ha la 12 ecc. Lo stesso concetto vale per le postazioni che occupano i livelli superiori al primo livello considerato nella in Figura 9.

Ulteriori ipotesi riguardano la politica di assegnamento adottata, la dimensione degli oggetti trattati, la capacità degli *slot* e la capacità complessiva del magazzino. Per quanto concerne la politica di assegnamento si è deciso per un sistema a classi non soffermandosi però sul criterio di determinazione delle classi perché ininfluente ai fini della generalità del modello. Ogni classe è caratterizzata da una domanda, nota a priori, espressa in unità di carico. Si è assunto che le unità di carico hanno tutte la stessa dimensione indipendentemente dalla classe di prodotto considerata. Inoltre, la capacità di uno *slot* è espressa in unità di carico ed è stata considerata omogenea all'interno dell'intero *layout*. In ogni singolo *slot* possono essere allocate anche due o più classi di

prodotto a patto che queste siano tra loro “compatibili”. La compatibilità tra prodotti è imposta anche tra *slot* adiacenti.

Al fine di una visione più schematica, sono di seguito sintetizzate l’insieme delle assunzioni intraprese:

- il *layout* di riferimento è rappresentato dal *layout* riportato in Figura 9, che impone un numero di corridoi trasversali pari a  $w/2$  e due diverse porte di I/O dislocate perpendicolarmente ai corridoi trasversali;
- la politica di assegnamento adottata è la politica *class-based*;
- la capacità complessiva del magazzino è tale da poter contenere tutti le classi prese in considerazione;
- il *layout* preso in considerazione è caratterizzato dalla presenza di ulteriori livelli (multi-livello) rappresentati attraverso il parametro  $h$  che indica l’espansione del magazzino in altezza in termini di numero di *slot*;
- uno *slot* può avere postazioni adiacenti nella sezione di un corridoio trasversale e nei livelli ad esso sovrastante; inoltre, può avere postazioni opposte e postazioni posteriori;
- due diverse classi di prodotti possono essere assegnati ad uno stesso slot o in slot adiacente se e solo se le due classi di prodotto sono fra esse compatibili;
- per le classi di prodotto che richiedono l’utilizzo di più *slot*, il modello deve prediligere un assegnamento che soddisfa il seguente ordine di priorità: in prima istanza occupare postazioni adiacenti nel medesimo blocco; successivamente si deve cercare uno *slot* adiacente nel medesimo corridoio per poi proseguire la ricerca nell’insieme degli *slot* opposti e posteriori.
- gli *slot* hanno tutti la stessa capacità (espressa in unità di carico);
- le unità di carico sono considerate omogenee in termini di dimensioni.

Preliminarmente alla definizione del modello matematico si introduce la seguente notazione:

- $N$  insieme delle classi di prodotti,  $i = 1, \dots, n$  ;
- $m$  numero di *slot* in verticale;
- $w$  numero di *slot* in orizzontale (in relazione al *layout* di riferimento);
- $h$  numero di livelli (*slot*) in altezza,  $l = 1, \dots, h$ ;
- $S$  insieme degli *slot* di primo livello;  $|S| = m * w$ ;
- $C$  matrice di compatibilità:  $c_{ij} = 1$  se la classe  $i$  è compatibile con la classe  $j$ ; 0 altrimenti;  $\forall i, j = 1, \dots, n$ ;

- $P$  vettore di sconto/penalità:  $p_1$  è lo sconto che si ottiene nella funzione obiettivo se una classe è assegnata a *slot* adiacenti nel medesimo blocco;  $p_2$  è lo sconto che si ottiene nella funzione obiettivo se una classe è assegnata a *slot* adiacenti nel medesimo corridoio;  $p_3$  è lo sconto che si ottiene nella funzione obiettivo se una classe è assegnata a *slot* opposti;  $p_4$  è lo sconto che si ottiene nella funzione obiettivo se una classe è assegnata a *slot* posteriori;  $p_5$  è la penalità da pagare se una classe occupa più di uno *slot*;
- $K$  insieme delle porte I/O,  $k = 1, \dots, |K|$  ;
- $D$  matrice delle distanze:  $d_{slk}$  distanza (in metri) dello *slot*  $s$ , di livello  $l$ , dalla porta  $k$ ,  $\forall s = 1, \dots, |S|, l = 1, \dots, h, k = 1, \dots, |K|$  ;
- $F$  matrice delle movimentazioni:  $f_{ik}$  numero medio giornaliero di movimentazioni (espresso in unità di carico) della classe  $i$  dalla porta  $k$ ,  $\forall i = 1, \dots, n, k = 1, \dots, |K|$  ;
- $r_i$  domanda, in unità di carico, della classe  $i$ ,  $\forall i = 1, \dots, n$  ;
- $Cap$  capacità di ogni singolo *slot* espressa in unità di carico;
- $T_1$  insieme di *slot* di primo livello che hanno uno *slot* adiacente nel medesimo corridoio. Con riferimento alla Figura 10, ad esempio:  $T_1 = \{1,2,3,4,6,7,8,9,11,12,13,14, \dots\}$ ;
- $T_2$  insieme di *slot* di primo livello che hanno uno *slot* opposto. Con riferimento alla Figura 10, ad esempio:  $T_2 = \{1,2,3,4,5,11,12,13,14,15,21,22,23,24,25, \dots\}$ ;
- $T_3$  insieme di *slot* di primo livello che hanno uno *slot* posteriore. Con riferimento alla Figura 10, ad esempio:  $T_3 = \{6,7,8,9,10,16,17,18,19,20, \dots\}$ ;
- $T_1 \cup T_2 \cup T_3 = S$ ;
- $\varepsilon$  costo per movimentare una unità di carico per un metro;

Inoltre, si definiscono le seguenti variabili decisionali:

- $y_{isl}$  numero di unità di carico di prodotto di classe  $i$  allocate allo *slot*  $s$  di livello  $l$ ,  $\forall i = 1, \dots, n, s = 1, \dots, |S|, l = 1, \dots, h$  ;
- $x_{isl}$  variabile binaria pari a 1 se la classe  $i$  è allocata allo *slot*  $s$  di livello  $l$ ; 0 altrimenti,  $\forall i = 1, \dots, n, s = 1, \dots, |S|, l = 1, \dots, h$  ;
- $z_{isl}^1$  variabile binaria pari a 1 se la classe  $i$  è allocata nello *slot* adiacente a  $s$  di livello  $l$  nello stesso blocco; 0 altrimenti,  $\forall i = 1, \dots, n, s = 1, \dots, |S|, l = 1, \dots, h$ ;
- $z_{isl}^2$  variabile binaria pari a 1 se la classe  $i$  è allocata nello *slot* adiacente a  $s$  di livello  $l$  nello stesso corridoio; 0 altrimenti,  $\forall i = 1, \dots, n, s = 1, \dots, |S|, l = 1, \dots, h$  ;
- $z_{isl}^3$  variabile binaria pari a 1 se la classe  $i$  è allocata nello *slot* opposto a  $s$  di livello  $l$ ; 0 altrimenti,  $\forall i = 1, \dots, n, s = 1, \dots, |S|, l = 1, \dots, h$  ;

- $z_{isl}^4$  variabile binaria pari a 1 se la classe  $i$  è allocata nello *slot* posteriore a  $s$  di livello  $l$ ; 0 altrimenti,  $\forall i = 1, \dots, n, s = 1, \dots, |S|, l = 1, \dots, h$ ;
- $b_{is}$  variabile binaria pari a 1 se la classe  $i$  è allocata in una posizione di stoccaggio appartenente al blocco  $s$ ; 0 altrimenti,  $\forall i = 1, \dots, n, s = 1, \dots, |S|$ ;

La formulazione matematica del modello è la seguente:

$$\min \quad Cost_F + Cost_P - Sav_P$$

s.a.

$$\sum_{s=1}^{|S|} \sum_{l=1}^h y_{isl} = r_i, \quad \forall i = 1, \dots, n \quad (3.1)$$

$$x_{isl} x_{i(s+1)l} = z_{isl}^1, \quad \forall i = 1, \dots, n, s = 1, \dots, |S|, l = 1, \dots, h-1 \quad (3.2)$$

$$z_{isl}^1 = 0, \quad \forall i = 1, \dots, n, s = 1, \dots, |S|, l = h \quad (3.3)$$

$$x_{isl} x_{i(s+1)l} = z_{isl}^2, \quad \forall i = 1, \dots, n, s \in T_1, l = 1, \dots, h \quad (3.4)$$

$$z_{isl}^2 = 0, \quad \forall i = 1, \dots, n, s \notin T_1, l = 1, \dots, h \quad (3.5)$$

$$x_{isl} x_{i(s+m)l} = z_{isl}^3, \quad \forall i = 1, \dots, n, s \in T_2, l = 1, \dots, h \quad (3.6)$$

$$z_{isl}^3 = 0, \quad \forall i = 1, \dots, n, s \notin T_2, l = 1, \dots, h \quad (3.7)$$

$$x_{isl} x_{i(s+m)l} = z_{isl}^4, \quad \forall i = 1, \dots, n, s \in T_3, l = 1, \dots, h \quad (3.8)$$

$$z_{isl}^4 = 0, \quad \forall i = 1, \dots, n, s \notin T_3, l = 1, \dots, h \quad (3.9)$$

$$b_{is} + b_{js} \leq 1 + c_{ij}, \quad \forall i = 1, \dots, n-1, j = i+1, \dots, n, s = 1, \dots, |S| \quad (3.10)$$

$$b_{is} + b_{j(s+1)} \leq 1 + c_{ij}, \quad \forall i = 1, \dots, n, j = 1, \dots, n, j \neq i, s \in T_1 \quad (3.11)$$

$$\sum_{i=1}^n y_{isl} \leq Cap, \quad \forall s = 1, \dots, |S|, l = 1, \dots, h \quad (3.12)$$

$$y_{isl} \leq M x_{isl}, \quad \forall i = 1, \dots, n, s = 1, \dots, |S|, l = 1, \dots, h \quad (3.13)$$

$$b_{is} \leq \sum_{l=1}^h x_{isl}, \quad \forall i = 1, \dots, n, s = 1, \dots, |S| \quad (3.14)$$

$$b_{is} \geq \frac{1}{h} \sum_{l=1}^h x_{isl} , \forall i = 1, \dots, n, s = 1, \dots, |S| \quad (3.15)$$

$$x_{isl} \in \{0,1\} , \forall i = 1, \dots, n, s = 1, \dots, |S|, l = 1, \dots, h \quad (3.16)$$

$$z_{isl}^1, z_{isl}^2, z_{isl}^3, z_{isl}^4 \in \{0,1\} , \forall i = 1, \dots, n, s = 1, \dots, |S|, l = 1, \dots, h \quad (3.17)$$

$$b_{is} \in \{0,1\} , \forall i = 1, \dots, n, s = 1, \dots, |S| \quad (3.18)$$

$$y_{isl} \geq 0 \text{ int} , \forall i = 1, \dots, n, s = 1, \dots, |S|, l = 1, \dots, h \quad (3.19)$$

dove

$$Cost_F = \varepsilon \sum_{k=1}^{|K|} \sum_{s=1}^{|S|} \sum_{l=1}^h d_{slk} \sum_{i=1}^n f_{ik} \frac{y_{isl}}{r_i} \quad (3.20)$$

$$Cost_P = \sum_{i=1}^n [(\sum_{s=1}^{|S|} \sum_{l=1}^h x_{isl}) - 1] p_5 \quad (3.21)$$

$$Sav_P = \sum_{i=1}^n \sum_{s=1}^{|S|} \sum_{l=1}^h (p_1 z_{isl}^1 + p_2 z_{isl}^2 + p_3 z_{isl}^3 + p_4 z_{isl}^4) \quad (3.22)$$

La funzione obiettivo è il *trade off* tra la minimizzazione dei costi di movimentazione e la minimizzazione del decentramento dei prodotti a scaffale. Il costo totale di movimentazione è rappresentato dalla componente  $Cost_F$ , che è influenzata dalle distanze che separano i vari *slot* dalle  $K$  porte di I/O e dal numero medio di operazioni di movimentazione per ciascun prodotto dalle relative porte. Il decentramento è perseguito con la differenza  $Cost_P - Sav_P$ . L'obiettivo primario è quello di allocare, se possibile, l'intera domanda di una generica classe in un unico *slot*. Quando tale domanda eccede la capacità dello *slot* allora si deve per forza occupare più di una posizione di stoccaggio e, secondo la politica di riduzione del decentramento, si predilige occupare dapprima *slot* adiacenti nello stesso blocco, poi adiacenti nello stesso corridoio, poi *slot* opposti, poi ancora *slot* posteriori e come ultima scelta *slot* generici. Questa scala di priorità è ottenuta assegnando valori alle componenti del vettore delle penalità tali per cui  $p_1 > p_2 > p_3 > p_4$ . La penalità  $p_5$  è impostata ad un valore molto superiore rispetto alle precedenti. Se fosse settata ad un valore confrontabile con quelli scelti per gli sconti potrebbe verificarsi una particolare allocazione di una generica classe tale per cui si combinerebbero tra loro diversi sconti la cui somma potrebbe essere superiore rispetto alla penalità  $p_5$ . Ciò spingerebbe il risolutore a creare queste situazioni non saturando ogni singolo *slot*. Con l'impostazione ad un valore nettamente superiore anche alla

somma di tutti gli sconti questo inconveniente viene evitato. Tale penalità è pagata quando una classe occupa più di uno *slot*: se la classe  $i$  occupa  $n$  *slot* allora viene pagata  $n-1$  volte. Questo legame è espresso nella funzione obiettivo dalla componente  $Cost_P$  in cui  $\sum_{s=1}^{|S|} \sum_{l=1}^h x_{isl}$  rappresenta il numero di *slot* occupati dalla generica classe  $i$ . Ciò spinge il risolutore a saturare la capacità di una posizione di stoccaggio prima di occuparne altre per una stessa classe. Nel momento in cui questa penalità viene pagata, il risolutore è incentivato ad allocare le restanti unità della classe in *slot* adiacenti nello stesso blocco (ottenendo una riduzione nella funzione obiettivo pari a  $p_1$ ), adiacenti nello stesso corridoio (con riduzione nella funzione obiettivo pari a  $p_2$ ), e così via seguendo l'ordine di priorità stabilito. Ciò permette di ottenere un'allocazione qualitativamente valida ai fini del decentramento dei prodotti.

I vincoli (3.1) sono i vincoli di soddisfacimento della domanda. I vincoli (3.2) sono i vincoli di legame tra le variabili  $x_{isl}$  e le variabili  $z_{isl}^1$  che rappresentano l'adiacenza nel medesimo blocco. Tale legame esprime che la variabile  $z_{isl}^1$  dovrà essere pari a 1 se e solo se una generica classe  $i$  è allocata in due *slot* adiacenti nel medesimo blocco; i vincoli (3.3) sono necessari per annullare le variabili  $z_{isl}^1$  che non hanno significato. Ci si riferisce a quelle variabili associate agli *slot* dell'ultimo livello (in caso di 4 livelli, ci si riferisce agli *slot* di quarto livello) che vanno annullate affinché non falsino la soluzione del problema. In assenza di questi vincoli, non essendo quest'ultime vincolate in nessun altro modo se non nei vincoli di binarietà, potrebbe accadere, infatti, che il risolutore assegni un valore pari a 1 a queste variabili, visto che comportano una riduzione nella funzione obiettivo, distorcendo il reale valore di quest'ultima. Per i vincoli (3.4)-(3.9) vale lo stesso discorso dei vincoli (3.2)-(3.3).

I vincoli (3.10)-(3.11) garantiscono, rispettivamente, che in uno stesso blocco e in blocchi adiacenti nel medesimo corridoio vengano inseriti soltanto classi di prodotti fra loro compatibili. Ad esempio, se un'azienda tratta sia prodotti di genere alimentare che prodotti per la cura dell'igiene personale, ci si aspetta che questi non siano stoccati in posizioni limitrofe. Lo stesso vale per articoli appartenenti alla catena del freddo. La considerazione di questi aspetti rende il modello matematico flessibile e adatto a tutte le realtà aziendali che presentano un portafoglio prodotti diversificato.

Si noti che il vincolo (3.10) viene scritto  $\forall i = 1, \dots, n-1, j = i+1, \dots, n$  (dove  $i$  e  $j$  rappresentano entrambi le classi di prodotti) e non  $\forall i = 1, \dots, n, j = 1, \dots, n$ . Questo per due motivi: il primo è che la matrice di compatibilità è una matrice simmetrica mentre il secondo è che una classe è sempre compatibile con se stessa. In tal modo, si evita l'inconveniente della ridondanza di alcuni vincoli e ciò permette di recuperare "memoria" ai fini della risoluzione di istanze di grandi

dimensioni. Il vincolo (3.11) invece viene scritto per  $j \neq i$ , sempre per il motivo che una classe è sempre compatibile con se stessa.

I vincoli (3.12) sono quelli sulla capacità di ogni singolo *slot*. I vincoli (3.13) sono i vincoli di legame tra le  $y_{isl}$  e le  $x_{isl}$ . La relazione da esprimere è:  $y_{isl} > 0 \Leftrightarrow x_{isl} = 1$ . In realtà, con questo vincolo, si garantisce soltanto che se  $y_{isl} > 0$  gioco forza la relativa  $x_{isl}$  deve essere pari a 1 ma non si garantisce la relazione inversa, ossia se  $x_{isl} = 1 \Rightarrow y_{isl} > 0$ . Tuttavia, all'ottimo, tale situazione non è riscontrabile in quanto il settaggio a 1 di una  $x_{isl}$  la cui relativa  $y_{isl}$  è 0 comporterebbe il pagamento della penalità  $p_5$  che potrebbe invece essere evitato. Di fatto viene dunque soddisfatta implicitamente anche la relazione inversa. Il gruppo di vincoli (3.14)-(3.15) garantisce il legame tra le variabili  $x_{isl}$  e le variabili  $b_{is}$ . La relazione che deve essere rispettata è che se  $\sum_{l=1}^h x_{isl} > 0$  vuol dire che nel blocco  $s$  è stata allocata la classe  $i$  e pertanto la relativa  $b_{is}$  deve essere posta pari a 1. Se, invece,  $\sum_{l=1}^h x_{isl} = 0$  allora la relativa  $b_{is}$  deve essere pari a 0. Infine i vincoli (3.16)-(3.19) sono i vincoli di binarietà e di non negatività e interezza.

Il modello, così formulato, rientra nella categoria dei problemi di PNL. Al fine di risolverlo con *CPLEX* è necessario trasformare il problema in forma lineare. Si è proceduto allora con la linearizzazione di quei vincoli che rendono appunto il problema non lineare. Più precisamente, al posto delle equazioni (3.2), (3.4), (3.6) e (3.8) sono state inserite rispettivamente le coppie di vincoli:

$$z_{isl}^1 \leq x_{isl} , \forall i = 1, \dots, n, s = 1, \dots, |S|, l = 1, \dots, h - 1 \quad (3.23)$$

$$z_{isl}^1 \leq x_{i(s+l+1)l} , \forall i = 1, \dots, n, s = 1, \dots, |S|, l = 1, \dots, h - 1 \quad (3.24)$$

$$z_{isl}^2 \leq x_{isl} , \forall i = 1, \dots, n, s \in T_1, l = 1, \dots, h \quad (3.25)$$

$$z_{isl}^2 \leq x_{i(s+1)l} , \forall i = 1, \dots, n, s \in T_1, l = 1, \dots, h \quad (3.26)$$

$$z_{isl}^3 \leq x_{isl} , \forall i = 1, \dots, n, s \in T_2, l = 1, \dots, h \quad (3.27)$$

$$z_{isl}^3 \leq x_{i(s+m)l} , \forall i = 1, \dots, n, s \in T_2, l = 1, \dots, h \quad (3.28)$$

$$z_{isl}^4 \leq x_{isl} , \forall i = 1, \dots, n, s \in T_3, l = 1, \dots, h \quad (3.29)$$

$$z_{isl}^4 \leq x_{i(s+m)l} , \forall i = 1, \dots, n, s \in T_3, l = 1, \dots, h \quad (3.30)$$

La relazione garantita dai vincoli sostituiti naturalmente continua a valere: le variabili  $z_{isl}^1, z_{isl}^2, z_{isl}^3, z_{isl}^4$  saranno pari a 1 se e solo se le relative variabili  $x_{isl}$  lo saranno anch'esse contemporaneamente. Teoricamente, riferendosi per esempio alla coppia (3.23)-(3.24), quando le relative variabili  $x_{isl}$  sono nel contempo pari a 1, la variabile  $z_{isl}^1$  può anche essere pari a 0. Tuttavia, all'ottimo, tale situazione non è possibile per questo motivo: si supponga di aver trovato una soluzione ottima in cui si verifica una situazione come quella appena descritta. Se la variabile  $z_{isl}^1$  fosse pari a 1 anziché 0, la coppia di vincoli (3.23)-(3.24) sarebbe ugualmente soddisfatta ed inoltre la funzione obiettivo sarebbe minore perché il settaggio a 1 della  $z_{isl}^1$  comporterebbe uno sconto pari a  $p_l$ . Ciò significa che è stata trovata una soluzione ammissibile con un valore di funzione obiettivo inferiore al valore della funzione obiettivo della soluzione ottima. E questo contraddice l'ipotesi iniziale di aver trovato una soluzione ottima.

#### **4.2 Implementazione del modello**

Il *software* scelto per la risoluzione del PAP con vincoli di compatibilità in un magazzino multi-livello è *ILOG CPLEX*, un ottimizzatore di problemi lineari o quadratici. Tale *software* permette di operare in tre modalità:

1. *Interattiva*: applicazione con la quale è possibile interagire tramite riga di comando. Si può descrivere il modello del problema (o importarlo da un file esterno) tramite una determinata sintassi, configurare il risolutore tramite l'impostazione di parametri opportuni e infine chiedere l'ottimizzazione del modello specificato;
2. *Concert Technology*: si tratta di un insieme di librerie che permette di estendere l'utilizzo di *ILOG CPLEX* alle applicazioni Java, C++ o .NET.
3. *Callable Library*: è una libreria del linguaggio di programmazione C che permette di estendere l'uso di *ILOG CPLEX* ad applicazioni scritte nei linguaggi C, Visual Basic, Fortran o in un qualsiasi altro linguaggio che può far riferimento alle funzioni del linguaggio C.

In questo caso si è optato per l'implementazione del modello con la seconda modalità, sfruttando l'ambiente di sviluppo *Eclipse* (un ambiente di sviluppo integrato multi-linguaggio, multipiattaforma e *open source*) e il linguaggio Java.

Con l'ausilio del linguaggio IDEF0 è, di seguito, schematizzato il processo eseguito per la risoluzione del problema.

# Un modello matematico per il PAP in un magazzino multi-livello con vincoli di compatibilità fra i prodotti (MLPAP)

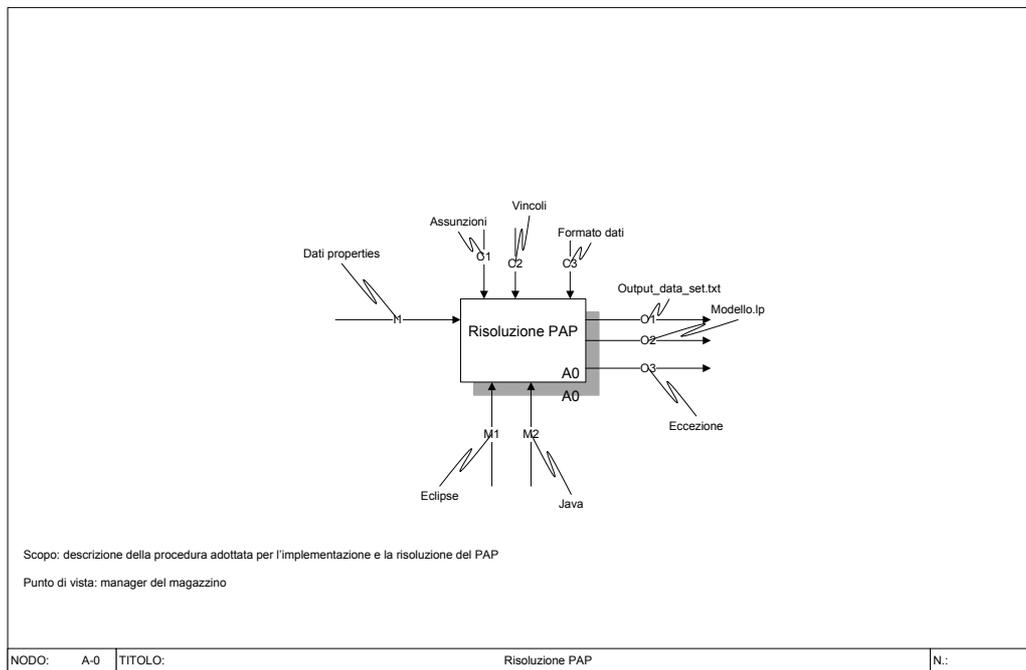


Figura 10 – Diagramma A-0

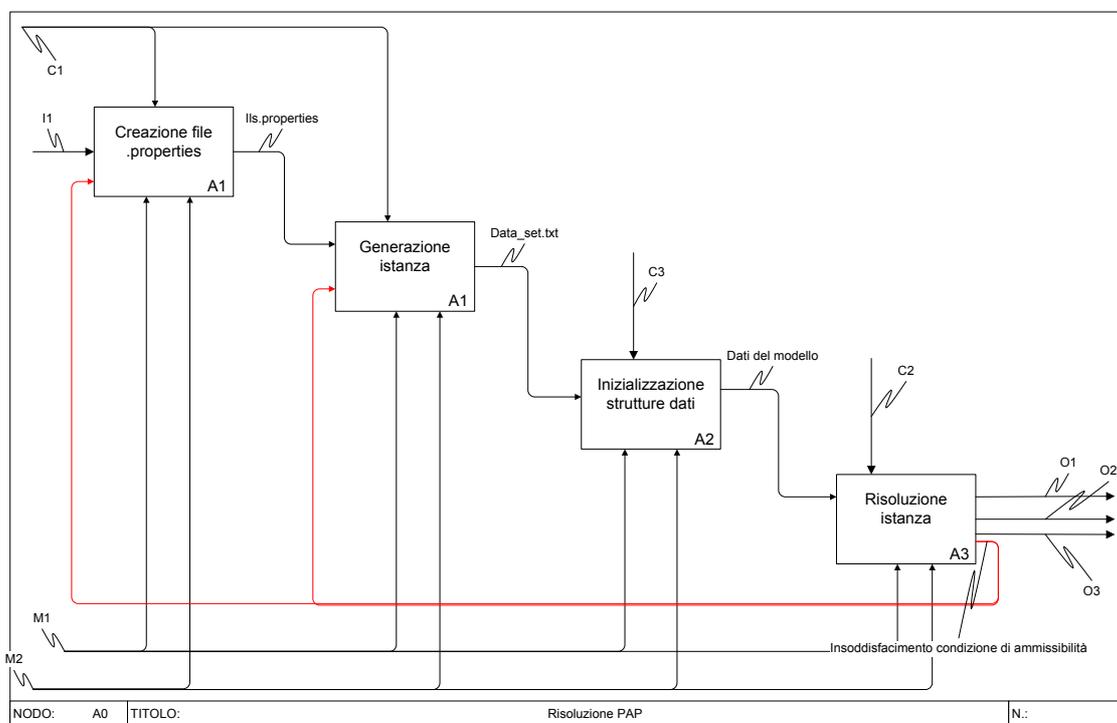


Figura 11 – Diagramma A0

La risoluzione di una istanza del problema di allocazione dei prodotti ai punti di stoccaggio passa attraverso l'attivazione di 4 processi:

1. creazione file *.properties*: viene creato un file in formato *properties* contenente il numero di classi di prodotti trattati, i parametri di *layout* dell'area di stoccaggio (numero di *slot* in verticale, numero di *slot* in orizzontale, numero di livelli, numero di porte, capacità degli *slot*), il costo unitario di movimentazione e i valori relativi agli sconti, ottenibili in base al posizionamento, e alla penalità, da sostenere nel caso di occupazione di più *slot* da parte di una generica classe;
2. Generazione istanza: attraverso l'implementazione di una classe denominata *GeneratorPap* viene letto il *file* creato con l'esecuzione del primo processo e vengono completati i parametri di *input* del modello con la generazione di ulteriori dati quali la domanda delle singole classi, la distanza di ogni *slot* dalla porta di ingresso e dalla porta di uscita, la matrice di compatibilità tra le classi e infine il numero medio giornaliero di movimentazioni da entrambe le porte di ogni classe di prodotto. Il tutto viene memorizzato in un *file* denominato *Data\_set.txt*;
3. inizializzazione strutture dati: la classe implementata nel terzo processo è denominata *DataSetManager* e ha la funzione di leggere il file *Data\_set.txt* e inizializzare tutte le strutture dati che compongono la classe *PAP* adibita alla risoluzione dell'istanza;
4. risoluzione istanza: nella classe *PAP* è implementato l'intero modello comprensivo dei dati di *input*, delle variabili decisionali e dei vincoli che delimitano la regione ammissibile dell'istanza. Qui viene richiamata la libreria *CPLEX* che permette di risolvere il problema. L'*output* di questa fase può assumere tre diverse forme:
  - 4.3.1 viene fornita la soluzione (inammissibile, ammissibile o ottima) dell'istanza trattata in un file denominato *Output\_data\_set.txt* - in cui vengono memorizzate anche altre informazioni quali la complessità dell'istanza (data dal numero di variabili e dal numero di vincoli), l'indice di saturazione della capacità totale del magazzino, il tempo impiegato per trovare la soluzione e la matrice di compatibilità tra le classi di prodotto - insieme alla forma esplicita del modello memorizzata in un file denominato *modello.lp*;
  - 4.3.2 viene stampato un messaggio nel file *Output\_data\_set.txt* che avvisa che la condizione di ammissibilità del problema non è soddisfatta. In tal caso, ai fini della generazione di istanze utili per la fase di *testing* del modello, si può procedere o

cambiando il range di valori all'interno del quale vengono generate le domande delle varie classi di prodotto oppure si può modificare la capacità degli *slot*;

- 4.3.3 viene lanciata un'eccezione di tipo *out of memory* che indica l'insufficienza di memoria nella risoluzione dell'istanza; ciò accade quando si cerca di risolvere istanze molto complesse.

### 4.3 Generazione delle istanze di test

Nel seguito, con il termine “istanza” si intende una particolarizzazione del modello in cui vengono fissati i parametri critici dello stesso: numero di prodotti da allocare ( $n$ ), capacità di ogni posizione di stoccaggio ( $Cap$ ), numero di *slot* in verticale ( $m$ ), orizzontale ( $w$ ) e in altezza ( $h$ ). Esistono poi altri parametri, comuni a tutte le istanze, che quantificano la penalità da sostenere nel caso di occupazione da parte di un prodotto/classe in più di uno *slot*, gli sconti per le posizioni adiacenti, opposte e posteriori e infine il costo unitario di movimentazione opportunamente determinato.

Ad oggi, non esiste in letteratura un insieme di *data set* consolidati per il PAP, pertanto la generazione delle istanze è stata realizzata utilizzando un apposito algoritmo codificato in Java, che, ricevuti i parametri caratterizzanti l'istanza, genera i rimanenti dati di *input* del modello matematico: vettore delle domande delle classi di prodotti, matrice di compatibilità delle stesse, matrice delle movimentazioni che i prodotti subiscono dalle porte di I/O, matrice delle distanze degli *slot* dalle porte di I/O.

Più specificatamente, il vettore delle domande dei prodotti si ottiene per mezzo di una funzione che genera una distribuzione uniformemente casuale di numeri interi nell'intervallo  $[10, 100]$ <sup>1</sup>; la stessa funzione, nell'intervallo  $[50, 100]$ , è utilizzata per la generazione della matrice delle movimentazioni; le distanze vengono generate attraverso una semplice formula del tipo  $offset + x + y + z$  per la porta 1 e  $offset + x + m - y + z$  per la porta 2 (si ricorda che il *layout* preso a riferimento prevede 2 porte di I/O), dove:

- *offset* è un numero fisso costante e fissato a priori;
- $x, y$  e  $z$  sono le coordinate nel piano cartesiano che variano rispettivamente tra  $[1, w]$ ,  $[1, m]$  e  $[1, h]$ .

---

<sup>1</sup> Per maggiori dettagli visitare <http://www.iro.umontreal.ca/~simardr/ssj/doc/html/index.html>, classe *UniformIntGen*.

Infine, la matrice delle compatibilità, al fine di testare la robustezza del modello matematico, viene prodotta sulla base di tre diversi scenari come di seguito descritto.

#### 4.4 Definizione degli scenari e settaggio dei parametri di controllo

Al fine di verificare l'incidenza della compatibilità tra le classi di prodotti sul modello, sono stati individuati tre scenari di complessità crescente:

- **Scenario S:** più semplice, con nessun grado di incompatibilità tra i prodotti;
- **Scenario MC:** mediamente complesso, con un medio grado di incompatibilità tra i prodotti;
- **Scenario C:** più complesso con un elevato grado di incompatibilità tra i prodotti.

A monte dello studio di sensitività, è stata eseguita una fase di pre analisi per il settaggio opportuno di due parametri di controllo: il costo unitario di movimentazione ( $\epsilon$ ) e il tempo limite entro il quale il solutore *CPLEX* deve fornire una soluzione. Sono state individuate 6 istanze del modello che si differenziano in base al parametro  $n$ :

- *Prob5*:  $n= 5$ ;
- *Prob10*:  $n= 10$ ;
- *Prob20*:  $n= 20$ ;
- *Prob50*:  $n= 50$ ;
- *Prob80*:  $n= 80$ ;
- *Prob100*:  $n= 100$ .

Ognuna di queste istanze è stata contestualizzata soltanto nello scenario S ed è caratterizzata dai seguenti parametri di *layout*:

Parametri layout	
m	2
w	4
h	2
k	2
P1	10
P2	8
P3	6
P4	4
P5	10000

Tabella 2 - Parametri prefissati per ogni istanza

Un modello matematico per il PAP in un magazzino multi-livello con vincoli di compatibilità fra i prodotti (MLPAP)

Per la determinazione di  $\epsilon$ , inoltre, sono stati impostati altri due parametri fondamentali che determinano il raggio d'azione del solutore *CPLEX*:

- il tempo limite, denominato *CPU budget* (descritto in precedenza): 7.200 secondi (2 ore);
- la memoria limite, denominata *Tree budget* (ossia la dimensione massima in MB dell'albero di *Branch&Bound* costruito dal solutore): 1000 MB.

I risultati ottenuti sono riportati in **Tabella 3**:

Problema\eps	0,1			0,2			0,3		
	Tempo (s)	Costo (€)	Gap	Tempo (s)	Costo (€)	Gap	Tempo (s)	Costo (€)	Gap
<b>Prob5_S</b>	0,984375	10611,1	0,07%	0,75	11232,2	0,01%	0,59375	11846,99	0,01%
<b>Prob10_S</b>	195,5938	41246,902	0,01%	71,921875	42531,81	0,01%	42,70313	43812,71	0,01%
<b>Prob20_S</b>	7154,625	42776,854	0,12%	7170,921875	45583,82	0,17%	7078,063	48395,73	0,20%
<b>Prob50_S</b>	7181,047	7168	0,42%	7183,328125	14336	0,27%	7180,422	21513,9	0,26%
<b>Prob80_S</b>	7179,703	11356,1	0,48%	7183,59375	22715	0,23%	7127,594	34059,9	0,19%
<b>Prob100_S</b>	7168,453	14154,89999	0,43%	7185,234375	28292,6	0,10%	7108,875	42438,3	0,09%
<b>media</b>	4813,401	21218,976	0,255%	4799,291667	27448,57	0,132%	4756,375	33677,92	0,127%
<b>differenziale</b>	4813,401	0	0,128%	4799,291667	6229,595	0,005%	4756,375	12458,95	0,000%

0,4			0,5			0,6			0,7		
Tempo (s)	Costo (€)	Gap	Tempo (s)	Costo (€)	Gap	Tempo (s)	Costo (€)	Gap	Tempo (s)	Costo (€)	Gap
0,578125	12462,66	0,46%	0,5625	13078,32	0,00%	0,5625	13693,99	0,57%	0,625	14309,65	0,09%
55,765625	45095,61	0,01%	38,60938	46378,51	0,01%	49,78125	47661,42	0,01%	47,82813	48944,32	0,01%
7137,09375	51210,63	0,24%	7167,938	54012,492	0,26%	7137,438	56823,39	0,27%	7181,375	59636,29	0,30%
7182,109375	28698,8	0,28%	6873,688	35840	0,17%	6601,453	43041,6	0,26%	7178,422	50176	0,20%
7138,25	45464,8	0,29%	7144,828	56796	0,26%	7150,922	68062,2	0,09%	7157,938	79408	0,10%
7181,109375	56598	0,11%	7146,844	70731,5	0,09%	7189	81882,6	0,09%	7148,047	99047,9	0,11%
4782,484375	39921,75	0,232%	4728,745	46139,47033	0,132%	4688,193	51860,87	0,215%	4785,706	58587,03	0,135%
4782,484375	18702,77	0,105%	40,55208	24920,49434	0,005%	0	30641,89	0,088%	97,51302	37368,05	0,008%

0,8			0,9			1		
Tempo (s)	Costo (€)	Gap	Tempo (s)	Costo (€)	Gap	Tempo (s)	Costo (€)	Gap
0,578125	14925,32	0,46%	0,609375	15540,98	0,55%	0,640625	16156,65	0,75%
53,78125	50227,22	0,01%	73,75	51510,122	0,01%	56,35938	52793,03	0,01%
7184,0625	62447,26	0,33%	7147,469	62439,188	0,31%	7161,797	68058,99	0,37%
7182,921875	57344	0,17%	7182,781	64512	0,17%	7178,891	71747	0,25%
7140,74	90862,4	0,23%	7159,484	102213,9	0,22%	7178,047	113521	0,18%
7183,296875	113227,2	0,14%	7152,35	127353,6	0,11%	7192,969	141508	0,12%
4790,896771	64838,9	0,223%	4786,074	70594,965	0,228%	4794,784	77297,44	0,280%
102,7040625	43619,92	0,097%	97,88125	49375,989	0,102%	106,5911	56078,47	0,153%

Tabella 3 - Determinazione dell'*epsilon best*

Il parametro  $\epsilon$  è stato determinato tenendo in considerazione il tempo di calcolo impiegato dal solutore, la qualità della soluzione (in rosso le istanze per cui è stata trovata la soluzione ottima, in nero quelle per cui la soluzione è ammissibile), il costo totale della stessa e il *gap* esistente tra la soluzione fornita da *CPLEX* e la migliore soluzione ottenibile nel nodo in cui il solutore stesso ha trovato la soluzione. Considerando che il tempo medio minimo registrato è di 4.688,193 secondi, il costo medio minimo è di 21.218,98 euro e il *gap* medio minimo è di 0,127%, il giusto

Un modello matematico per il PAP in un magazzino multi-livello con vincoli di compatibilità fra i prodotti (MLPAP)

compromesso ( $\varepsilon$  best), individuato considerando i differenziali dei valori medi dai valori minimi, è  $\varepsilon = 0,6$ . Quest'ultimo è stato utilizzato per la determinazione del *CPU budget best* da assegnare al solutore per i successivi *test*. I risultati utili alla determinazione del *CPU budget* sono riportati in **Tabella 4**.

Problema\CPU budget	1h			2h			3h		
	Tempo (s)	Costo (€)	Gap	Tempo (s)	Costo (€)	Gap	Tempo (s)	Costo (€)	Gap
<b>Prob5_S</b>	0,5625	13693,988	0,57%	0,5625	13693,99	0,57%	0,578125	13693,99	0,57%
<b>Prob10_S</b>	49,46875	47661,4151	0,01%	49,78125	47661,42	0,01%	49,59375	47661,42	0,01%
<b>Prob20_S</b>	3589,9531	56828,945	0,30%	7137,438	56823,39	0,27%	10771,72	56823,39	0,27%
<b>Prob50_S</b>	3577,9063	43044	0,26%	6601,453	43041,6	0,26%	10731,64	43033,8	0,23%
<b>Prob80_S</b>	3584,0156	68062,2	0,10%	7150,922	68062,2	0,09%	10703,02	68062,2	0,09%
<b>Prob100_S</b>	3592,9844	84883,2	0,09%	7189	81882,6	0,09%	10775,88	84874,2	0,08%
<b>media</b>	2399,1484	52362,29135	0,222%	4688,193	51860,87	0,215%	7172,07	52358,17	0,208%
<b>differenziale</b>	0	501,4253517	0,013%	2289,044	0	0,007%	4772,922	497,2997	0,000%

4h			5h			6h		
Tempo (s)	Costo (€)	Gap	Tempo (s)	Costo (€)	Gap	Tempo (s)	Costo (€)	Gap
0,59375	13693,99	0,57%	0,59375	13693,988	0,57%	0,578125	13693,99	0,57%
49,09375	47661,42	0,01%	49,34375	47661,4151	0,01%	49,45313	47661,42	0,01%
14361,5625	56823,39	0,26%	17952,55	56823,391	0,26%	21540,8	56823,39	0,25%
14346,14063	43041,6	0,25%	17888,61	43041,6	0,25%	21522,31	43041,6	0,25%
14261,90625	68062,2	0,09%	17934,72	68062,2	0,09%	21401,5	68062,2	0,09%
14362,79688	84876	0,08%	17945,86	84876,59999	0,08%	21541,36	84880,8	0,08%
9563,682292	52359,77	0,210%	11961,95	52359,86568	0,210%	14342,67	52360,57	0,208%
7164,533854	498,8997	0,002%	9562,797	498,9996833	0,002%	11943,52	499,6997	0,000%

Tabella 4 - Determinazione del *CPU budget best*

Anche in questo caso, considerando che il tempo medio minimo registrato è pari a 2.399,15 secondi, il costo medio minimo è pari a 51.860,86 euro e il *gap* medio minimo è dello 0,208%, la scelta si è basata sul miglior *trade off* tra queste misure ed è ricaduta sulle 2 ore, ossia 7.200 secondi.

Il *Tree budget*, invece, non è stato determinato in maniera scrupolosa come i due parametri sopradescritti, in quanto non particolarmente incidente per gli scopi dell'analisi. Per tale ragione si è scelto di impostarlo per tutti i *test* ad un valore pari a 1000 MB.

#### 4.5 Identificazione dei parametri chiave del modello

I parametri ritenuti maggiormente incidenti sulla *performance* del modello sono stati individuati nel numero di classi di prodotti trattati, numero di *slot* in verticale, numero di *slot* in orizzontale, numero di livelli, infine, capacità di ogni singolo *slot*.

Al variare dei suddetti parametri, si è analizzato il comportamento del modello matematico per mezzo di indicatori di *performance* quali la complessità del modello, la qualità della soluzione fornita dal solutore, il costo computazionale e infine il grado di decentramento della soluzione. In particolare, la complessità del modello è misurata nel numero di variabili e di vincoli dell'istanza presa in considerazione; per qualità della soluzione si intende soluzione ottima oppure ammissibile (nelle tabelle che seguiranno, la convenzione adottata prevede che le istanze per cui si ha la soluzione ottima siano evidenziate in rosso, mentre quelle per cui si ha soltanto una soluzione ammissibile siano scritte in nero); il costo computazionale è rappresentato dal tempo (in secondi) impiegato dal solutore per fornire la soluzione dell'istanza; infine il grado di decentramento della soluzione è valutato attraverso un coefficiente determinato come:

$$1 - \left( \frac{\text{valore peggiore} - \text{valore corrente}}{\text{valore peggiore}} \right) \quad (4.1)$$

dove:

- il *valore peggiore* rappresenta il valore più alto di decentramento ottenibile nell'istanza considerata ed è calcolato come

$$n[(mwh - 1) * p_5] \quad (4.2)$$

in cui  $p_5$  è la penalità pagata per l'occupazione di due o più *slot* da parte di una classe di prodotto;

- il *valore corrente* è la differenza tra il costo di penalità e lo sconto dell'istanza considerata.

#### **4.6 Comportamento del modello al variare del numero di classi di prodotti**

Di seguito è riportato l'esito dello studio di sensitività al variare del numero di classi di prodotti trattati. La **Tabella 5** mostra i parametri che in questa fase sono prefissati per tutte le istanze, mentre in **Tabella 6** è riportato il *range* di variabilità del parametro  $n$  (numero delle classe di prodotto).

Un modello matematico per il PAP in un magazzino multi-livello con vincoli di compatibilità fra i prodotti (MLPAP)

---

Parametri	
m	4
w	6
h	4
k	2
Cap	100
Eps best	0,6
P1	10
P2	8
P3	6
P4	4
P5	10000

Tabella 5 - Parametri predefiniti

Istanza	n	N° variabili	N° vincoli	Scenario	Tempo di calcolo (s)
<b>Prob5</b>	5	3000	7461	S	0,875
				MC	1,234
				C	2,063
<b>Prob10</b>	10	6000	16326	S	2,750
				MC	8,469
				C	47,313
<b>Prob20</b>	20	12000	38556	S	46,000
				MC	105,234
				C	5675,563
<b>Prob50</b>	50	30000	141246	S	7179,438
				MC	7175,609
				C	7188,484
<b>Prob80</b>	80	48000	297936	S	7162,094
				MC	7183,984
				C	7172,344
<b>Prob100</b>	100	60000	432396	S	7159,313
				MC	7177,547
				C	7157,375

Tabella 6 – Range di variazione del parametro n

Analizzando i risultati ottenuto è possibile evincere le seguenti informazioni:

- in risposta alla variazione di  $n$ , il modello si comporta come un classico modello *NP-hard*, in quanto, la complessità cresce esponenzialmente all'aumentare del parametro considerato (vedi grafico riportato in **Figura 12**).

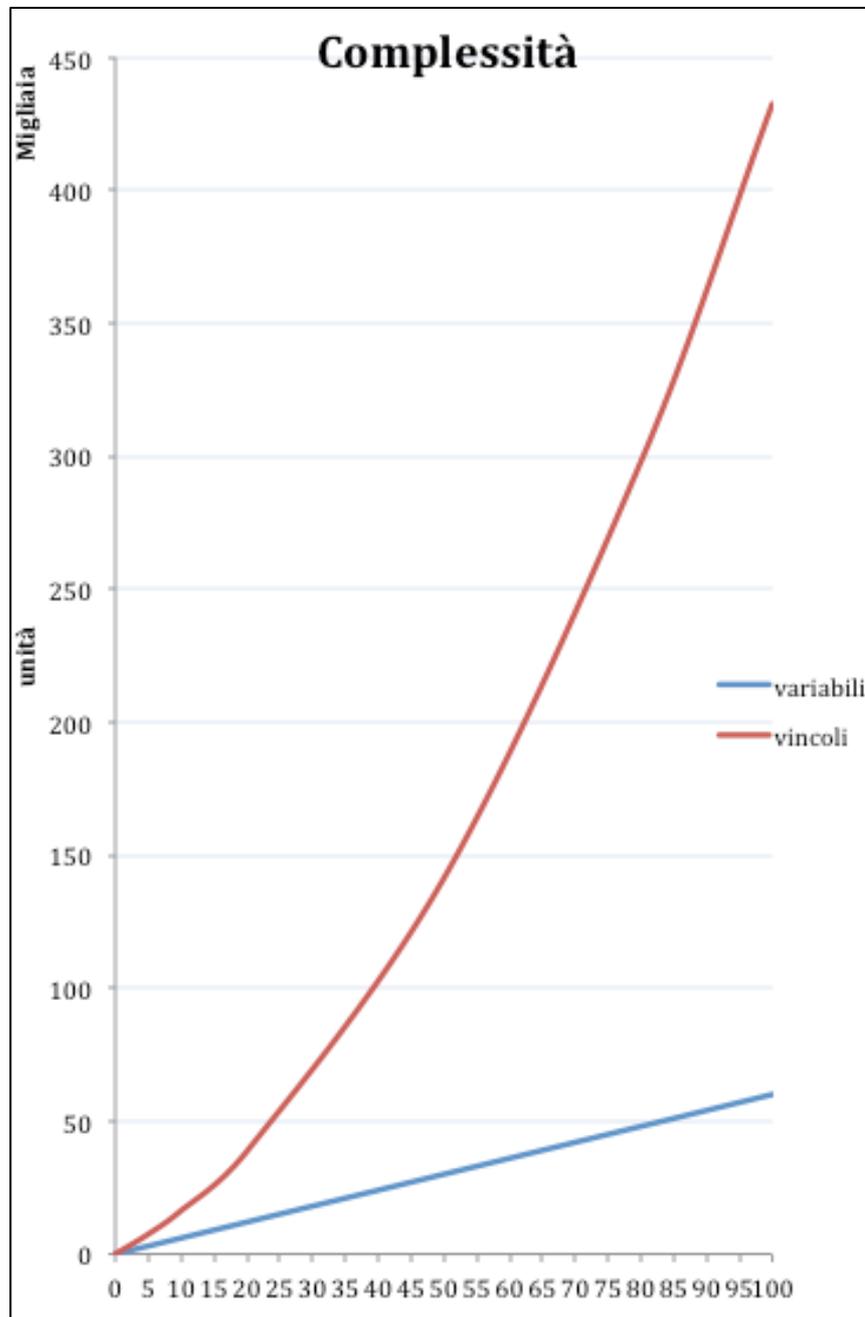


Figura 12 - Complessità del modello al variare di  $n$

- La crescente complessità si riflette sulla qualità del lavoro svolto da *CPLEX* il quale, fino ad  $n$  uguale a 20, opera in maniera ottimale, indipendentemente dallo scenario preso in considerazione e in tempi più che ragionevoli; oltre 20, fornisce la miglior soluzione ammissibile che riesce a calcolare nell'intervallo temporale prefissato. Il costo

computazionale, infatti, si attesta intorno al limite massimo consentito sia che si tratti dello scenario S, MC o C. L'incidenza dei vari scenari, e quindi del grado di compatibilità, sulle *performance* del modello è evidente in corrispondenza dell'istanza *Prob20*, nella quale lo scarto tra il tempo impiegato per calcolare la soluzione ottima negli scenari S e MC e il tempo impiegato per calcolare la soluzione ottima nello scenario C è particolarmente elevato.

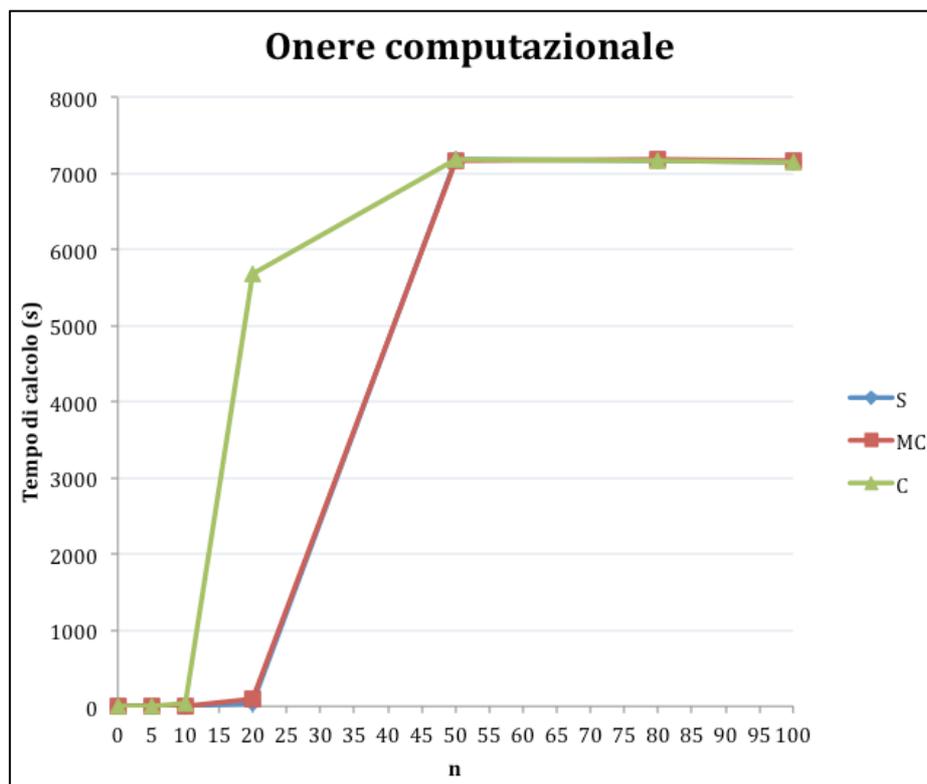


Figura 13 - Tempo di calcolo della soluzione

Il grado di decentramento delle soluzioni non è stato preso in considerazione in quanto la capacità degli *slot* impostata è tale da rendere costante questo tipo di informazione e dunque ininfluenza in questa fase dell'analisi.

#### 4.7 Comportamento del modello al variare della capacità degli slot

L'analisi del modello matematico relativa al parametro capacità è stata condotta sull'istanza *Prob20*, perché ritenuta sufficiente a mostrare il comportamento medio del modello al variare del suddetto parametro. I valori assunti dalla capacità sono stati definiti in modo casuale a partire da un valore minimo opportunamente calcolato e derivante direttamente dalla condizione di ammissibilità del modello per l'ottimizzazione del PAP in un magazzino multi-livello e con vincoli di

compatibilità fra i prodotti. Considerando che si è ipotizzato una capacità omogenea per tutte le posizioni di stoccaggio, tale condizione si traduce in forma analitica in:

$$Cap \geq \left\lceil \frac{\sum_{i=1}^n r_i}{mwh} \right\rceil \quad (4.3)$$

dove il secondo membro della disequazione rappresenta il valore minimo che il parametro capacità può assumere.

Tra le misure di *performance* non è stata presa in considerazione la complessità del modello in quanto la capacità non incide sul numero di variabili e conseguentemente sul numero di vincoli.

Nella **Tabella 7** sono riportati i valori costanti per le istanze prese in considerazione, mentre in **Tabella 8** è sintetizzato l'esito di questa fase di studio.

Parametri	
n	20
m	4
w	6
h	4
k	2
Eps best	0,6
P1	10
P2	8
P3	6
P4	4
P5	10000

Tabella 7 – Parametri con valori costanti

Capacità (unità di carico)	Scenario	Tempo di calcolo (s)	Grado decentramento (%)
11	S	7035,047	4,050%
	MC	7176,766	4,103%
	C	7151,766	4,102%
20	S	7175,344	1,999%
	MC	7160,078	1,999%
	C	7111,234	1,999%
30	S	7175,672	1,210%
	MC	7176,844	1,210%
	C	7176,594	1,210%
50	S	7177,063	0,421%
	MC	7164,563	0,421%
	C	7179,156	0,421%

80	S	7176,766	0,158%
	MC	7154,438	0,158%
	C	7136,375	0,158%
100	S	46,000	0,000%
	MC	105,234	0,000%
	C	5675,563	0,000%
150	S	48,344	0,000%
	MC	232,359	0,000%
	C	829,422	0,000%
200	S	45,594	0,000%
	MC	75,406	0,000%
	C	313,828	0,000%

Tabella 8 – Range di variabilità del parametro *Cap* (capacità)

È ragionevole pensare che all'aumentare della capacità di ogni singolo *slot* diminuisca il grado di difficoltà che il solutore incontra nel calcolare l'allocazione ottimale, in quanto, avendo più spazio a disposizione in ogni *slot*, è maggiore la probabilità di allocare l'intera domanda di una generica classe di prodotto all'interno di un'unica posizione di stoccaggio. Ciò comporta una riduzione delle potenziali combinazioni da analizzare e quindi una riduzione dell'albero di *Branch&Bound* costruito dal solutore che si riflette nell'ottenimento di una soluzione di qualità migliore. Con una capacità che va dalle 100 unità di carico in poi, infatti, *CPLEX* trova la soluzione ottima, per l'istanza corrente, indipendentemente dal grado di compatibilità in un tempo ragionevole. È da notare inoltre, come lecitamente ci si attendeva, la diminuzione del grado di decentramento all'aumentare della capacità.

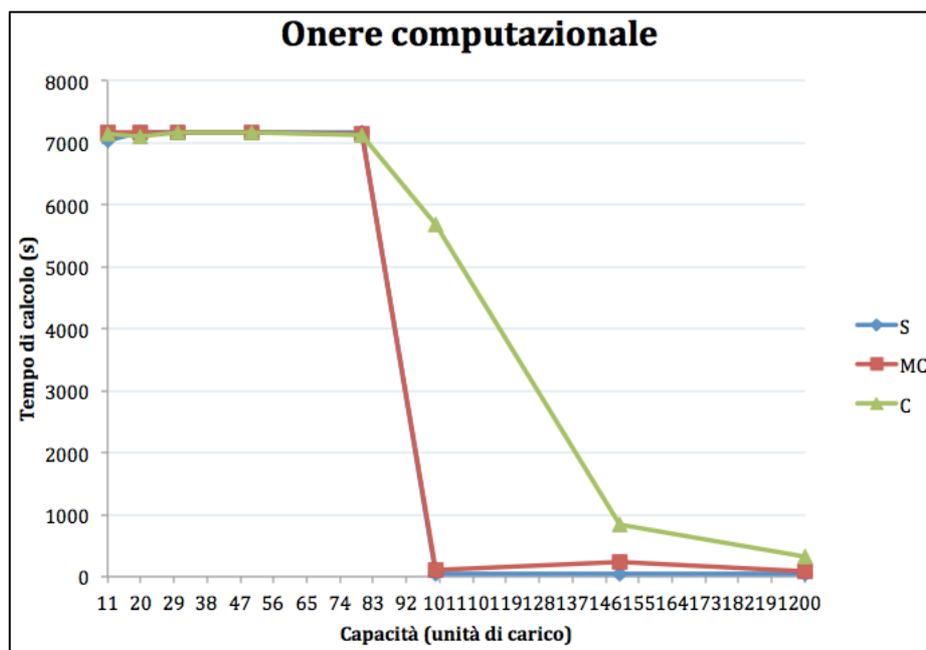


Figura 14 - Costo computazionale

In **Figura 14** e in **Figura 15** è rappresentata rispettivamente l'evoluzione dell'onere computazionale e del grado di decentramento della soluzione per i tre scenari di compatibilità.

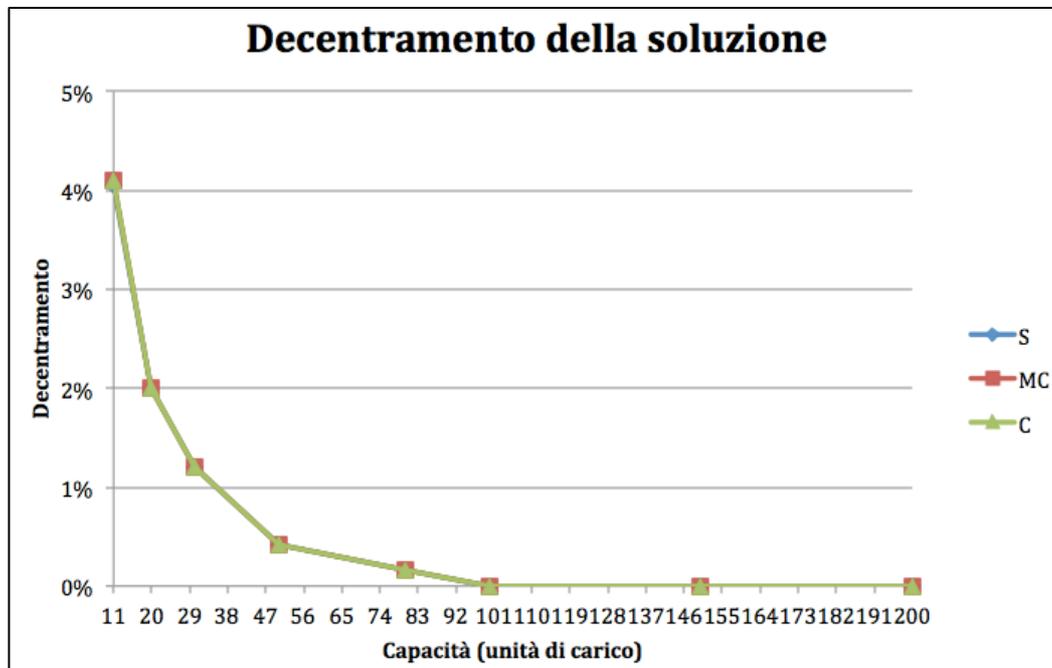


Figura 15 – Decentramento dei prodotti a scaffale

#### 4.8 Comportamento del modello al variare dei parametri $m$ , $w$ e $h$

Nella sezione seguente viene valutato l'impatto che la variazione dei parametri  $m$ ,  $w$  e  $h$  ha sul costo computazionale sostenuto dal solutore e sulla qualità della soluzione ottenuta. Vengono prese in considerazione soltanto queste due misure di *performance* in quanto le altre sono poco significative per gli scopi di questa fase di studio. In particolare, la complessità, in termini di numero di variabili e di vincoli, aumenta in maniera lineare con l'aumentare del valore assegnato ai parametri in esame, mentre per il decentramento vale lo stesso discorso fatto in precedenza, ossia la capacità degli *slot* è impostata ad un valore tale da permettere che il grado di decentramento rimanga costante e quindi ininfluente ai fini del controllo di *performance*.

I *test* sono stati eseguiti considerando tutte le istanze (da *Prob5* a *Prob100*) e tutti e tre gli scenari di compatibilità. Nelle tabelle seguenti sono riportati i valori assegnati ai parametri costanti e quelli in corrispondenza della variazione dei parametri  $m$ ,  $w$  e  $h$ .

Un modello matematico per il PAP in un magazzino multi-livello con vincoli di compatibilità fra i prodotti (MLPAP)

---

Parametri al variare di <b>m</b>		Parametri al variare di <b>w</b>		Parametri al variare di <b>h</b>	
w	6	m	4	m	4
h	4	h	4	w	6
k	2	k	2	k	2
Cap	100	Cap	100	Cap	100
Eps best	0,6	Eps best	0,6	Eps best	0,6
P1	10	P1	10	P1	10
P2	8	P2	8	P2	8
P3	6	P3	6	P3	6
P4	4	P4	4	P4	4
P5	10000	P5	10000	P5	10000

Tabella 9 - Parametri costanti al variare del parametro in esame

Nelle tabelle sottostanti sono riportati i risultati ottenuti al variare del parametro  $m$ ,  $w$  e  $h$ .

Istanza	m	Scenario	Tempo di calcolo (s)	
Prob5	4	S	0,875	
		MC	1,234	
		C	2,063	
	5	S	1,078	
		MC	1,859	
		C	2,469	
	6	S	1,469	
		MC	4,188	
		C	4,422	
	7	S	1,625	
		MC	3,438	
		C	3,453	
	8	S	1,953	
		MC	3,344	
		C	4,594	
	9	S	2,297	
		MC	3,891	
		C	6,391	
	10	S	2,672	
		MC	4,375	
		C	6,047	
	Prob10	4	S	2,750
			MC	8,469
			C	47,313
		5	S	5,328
			MC	17,641
			C	37,922
6		S	7,203	
		MC	10,969	
		C	24,500	
7		S	5,297	
		MC	8,625	
		C	15,469	
8		S	5,938	
		MC	9,906	
		C	23,188	
9		S	7,031	
		MC	12,266	
		C	19,391	
10		S	8,578	
		MC	14,141	
		C	23,750	
Prob20		4	S	46,000
			MC	105,234
			C	5675,563
		5	S	86,938
			MC	197,438
			C	1081,000
	6	S	108,781	
		MC	159,688	
		C	940,938	
	7	S	380,766	
		MC	513,578	
		C	2397,359	
	8	S	268,063	
		MC	304,625	
		C	1567,563	
	9	S	326,797	
		MC	333,234	
		C	326,797	
	10	S	344,813	
		MC	268,219	
		C	550,953	
	Prob50	4	S	7179,438
			MC	7175,609
			C	7188,484
		5	S	7183,297
			MC	7183,281
			C	7183,313
6		S	7111,734	
		MC	7059,828	
		C	6740,750	
7		S	7184,578	
		MC	7186,188	
		C	7185,531	
8		S	6853,719	
		MC	7185,984	
		C	7187,141	
9		S	7187,297	
		MC	7176,516	
		C	7188,438	
10		S	7185,234	
		MC	7185,859	
		C	7132,641	
Prob80		4	S	7162,094
			MC	7183,984
			C	7172,344
		5	S	7099,719
			MC	7023,844
			C	7189,000
	6	S	7006,125	
		MC	7165,750	
		C	7135,719	
	7	S	7119,563	
		MC	7172,375	
		C	7139,828	
	8	S	7000,906	
		MC	7176,406	
		C	7155,094	
	9	S	7183,547	
		MC	7130,781	
		C	7174,438	
	10	S	7154,672	
		MC	6987,391	
		C	7128,344	
	Prob100	4	S	7159,313
			MC	7177,547
			C	7157,375
		5	S	7162,656
			MC	7177,672
			C	7065,750
6		S	6952,000	
		MC	7173,281	
		C	7167,281	
7		S	7151,750	
		MC	7116,969	
		C	7179,281	
8		S	7120,125	
		MC	7179,450	
		C	7052,734	
9		S	7177,875	
		MC	7165,250	
		C	7121,078	
10		S	7199,266	
		MC	7172,125	
		C	7194,203	

Tabella 10 - Variazione del parametro  $m$

Istanza	w	Scenario	Tempo di calcolo (s)	Istanza	w	Scenario	Tempo di calcolo (s)
Prob5	6	S	0,875	Prob50	6	S	7179,438
		MC	1,234			MC	7175,609
		C	2,063			C	7188,484
	8	S	1,125		8	S	6818,844
		MC	1,984			MC	7181,375
		C	3,766			C	7182,203
	10	S	1,516		10	S	7184,781
		MC	2,641			MC	7182,266
		C	5,141			C	7183,063
	12	S	1,828		12	S	7184,000
		MC	4,563			MC	7173,516
		C	6,594			C	7181,750
	14	S	2,375		14	S	7184,234
		MC	4,000			MC	7184,453
		C	9,609			C	7184,281
	16	S	2,891		16	S	7184,922
		MC	4,750			MC	7178,656
		C	6,563			C	7185,719
18	S	3,422	18	S	7187,938		
	MC	5,641		MC	7187,266		
	C	18,719		C	7187,859		
Prob10	6	S	2,750	Prob80	6	S	7162,094
		MC	8,469			MC	7183,984
		C	47,313			C	7172,344
	8	S	5,359		8	S	7008,984
		MC	16,516			MC	7181,453
		C	62,406			C	6766,969
	10	S	7,969		10	S	7084,953
		MC	22,563			MC	7154,547
		C	91,250			C	7092,969
	12	S	9,313		12	S	7143,109
		MC	27,984			MC	7165,016
		C	95,688			C	7192,688
	14	S	10,453		14	S	7153,625
		MC	27,813			MC	7057,594
		C	181,750			C	7182,734
	16	S	16,016		16	S	7149,625
		MC	34,016			MC	7131,875
		C	170,375			C	7183,656
18	S	16,797	18	S	7197,109		
	MC	44,766		MC	7192,000		
	C	212,672		C	7169,563		
Prob20	6	S	46,000	Prob100	6	S	7159,313
		MC	105,234			MC	7177,547
		C	5675,563			C	7157,375
	8	S	97,063		8	S	7129,594
		MC	222,797			MC	7174,016
		C	7161,641			C	6988,531
	10	S	90,266		10	S	7169,256
		MC	301,281			MC	7190,422
		C	7180,281			C	7158,141
	12	S	108,469		12	S	7156,594
		MC	311,656			MC	7141,890
		C	7145,672			C	7156,188
	14	S	172,750		14	S	7164,516
		MC	306,516			MC	7091,563
		C	7181,359			C	7129,200
	16	S	150,234		16	S	7172,656
		MC	537,109			MC	7193,484
		C	7166,406			C	7189,594
18	S	372,031	18	S	6961,750		
	MC	475,797		MC	7006,672		
	C	7152,219		C	7158,828		

Tabella 11 - Variazione del parametro  $w$

Istanza	h	Scenario	Tempo di calcolo (s)	Istanza	h	Scenario	Tempo di calcolo (s)		
Prob5	4	S	0,875	Prob50	4	S	7179,438		
		MC	1,234			MC	7175,609		
		C	2,063			C	7188,484		
	5	S	1,031		5	S	7180,734		
		MC	2,328			MC	7180,516		
		C	3,094			C	7180,172		
	6	S	1,109		6	S	7178,234		
		MC	3,344			MC	7180,844		
		C	4,063			C	7180,438		
	7	S	1,328		7	S	7178,750		
		MC	3,984			MC	6652,047		
		C	5,938			C	6887,406		
	8	S	1,609		8	S	7003,875		
		MC	4,375			MC	7184,234		
		C	6,688			C	7182,500		
	9	S	1,781		9	S	7181,359		
		MC	5,297			MC	7181,438		
		C	7,031			C	7181,375		
	10	S	2,000		10	S	7181,578		
		MC	4,291			MC	7178,281		
		C	10,063			C	7181,328		
	Prob10	4	S		2,750	Prob80	4	S	7162,094
			MC		8,469			MC	7183,984
			C		47,313			C	7172,344
		5	S		3,813		5	S	7182,594
			MC		13,172			MC	7182,969
			C		39,797			C	7182,234
6		S	3,516	6	S		7185,422		
		MC	24,766		MC		7182,891		
		C	50,641		C		7183,922		
7		S	4,344	7	S		6958,938		
		MC	24,813		MC		7184,813		
		C	58,781		C		7184,125		
8		S	5,391	8	S		7185,094		
		MC	23,219		MC		7180,250		
		C	54,531		C		7185,188		
9		S	7,500	9	S		7185,219		
		MC	26,125		MC		7184,609		
		C	69,969		C		7185,453		
10		S	8,344	10	S		7183,516		
		MC	32,219		MC		7187,234		
		C	84,391		C		7185,578		
Prob20		4	S	46,000	Prob100		4	S	7159,313
			MC	105,234				MC	7177,547
			C	5675,563				C	7157,375
		5	S	66,500			5	S	7177,063
			MC	187,750				MC	7190,125
			C	7106,016				C	7148,063
	6	S	142,109	6		S	6974,031		
		MC	257,891			MC	6489,516		
		C	7162,250			C	7055,141		
	7	S	103,422	7		S	6920,484		
		MC	374,016			MC	7186,203		
		C	7180,266			C	7161,125		
	8	S	106,891	8		S	7160,734		
		MC	443,734			MC	7140,500		
		C	7136,531			C	7134,960		
	9	S	161,375	9		S	7126,078		
		MC	404,063			MC	6985,609		
		C	7158,078			C	6655,156		
	10	S	153,156	10		S	7085,406		
		MC	494,922			MC	7151,047		
		C	7143,625			C	7183,563		

Tabella 12 - Variazione del parametro  $h$

Dalla **Tabella 10**, **Tabella 11** e **Tabella 12**, dal punto di vista della qualità della soluzione, si evince come, per le istanze che presentano un numero maggiore di venti classi di prodotti, il solutore non riesca a convergere verso l'ottimo globale, indipendentemente dal valore assunto dai tre parametri in esame. Ciò testimonia, in maniera ancora più evidente, la notevole sensibilità del modello matematico, in termini di complessità, al numero di prodotti trattati. Per le istanze con un valore di  $n$  al di sotto delle 20 unità, invece, *CPLEX* riesce a trovare una soluzione ottima con qualsiasi grado di compatibilità soltanto al variare del numero di *slot* in verticale. Per i restanti due parametri, infatti, accade che lo scenario C, in corrispondenza di determinati valori, inizia ad incidere in modo tale da richiedere un tempo maggiore di quello concesso al solutore per trovare l'ottimo globale.

Per quanto concerne il costo computazionale, invece, è stato calcolato, sia per  $m$ , che per  $w$  e  $h$ , il tempo medio impiegato per il calcolo della soluzione, distinguendo i tre scenari di compatibilità. I grafici sottostanti testimoniano come la variazione del parametro  $m$  sia la meno influenzante la performance del modello matematico, con un costo computazionale medio che rimane più o meno costante per gli scenari S e MC mentre, per lo scenario complesso, tende, in linea di massima, a diminuire all'aumentare di  $m$ , per poi stabilizzarsi intorno ad un valore medio prossimo a quello degli scenari meno complessi. Ciò non accade al variare sia di  $w$  che di  $h$ , dove lo scenario C condiziona in maniera nettamente superiore la *performance* del modello matematico.

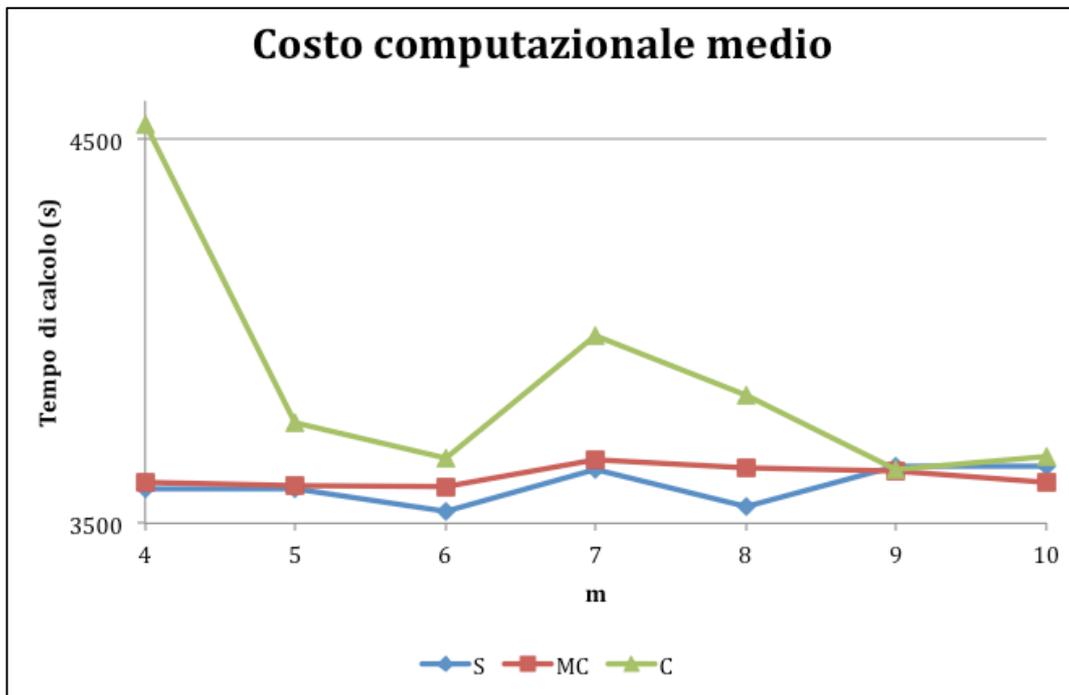


Figura 16 - Tempo di calcolo medio al variare del parametro  $m$

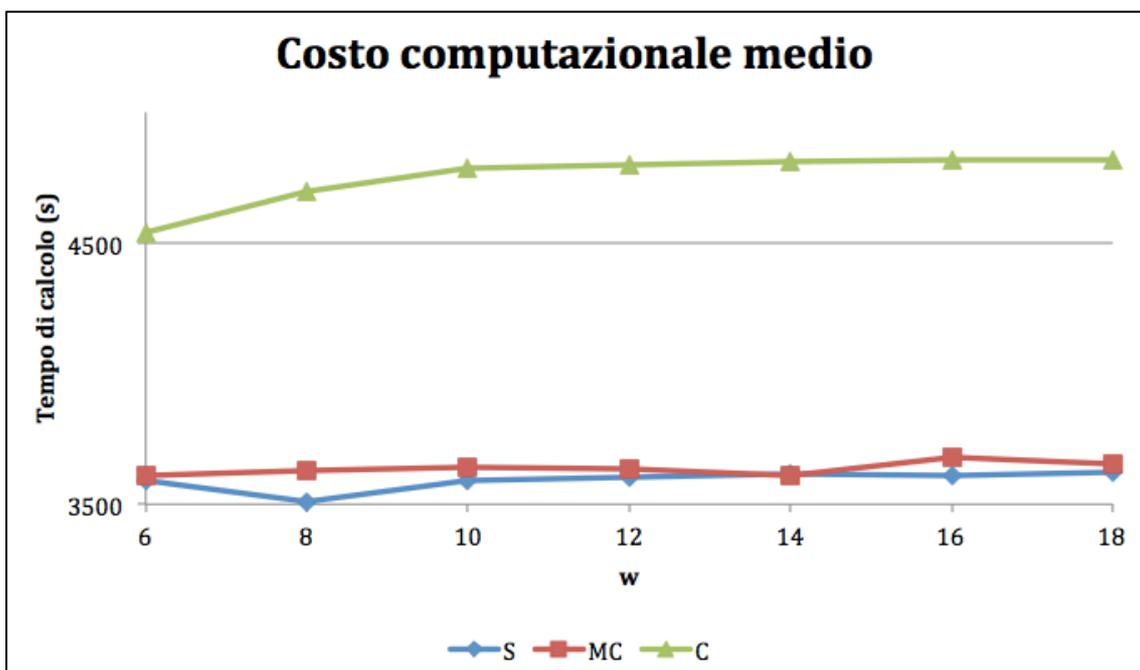


Figura 17 - Tempo di calcolo medio al variare del parametro  $w$

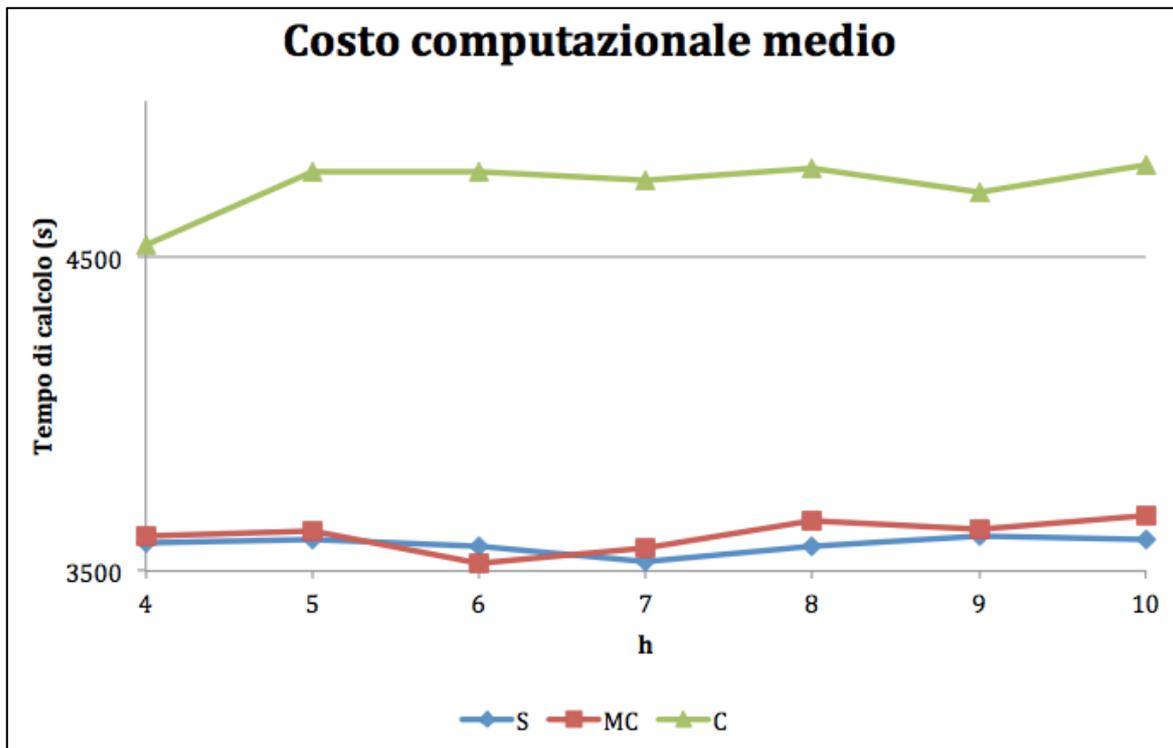


Figura 18 - Tempo di calcolo medio al variare del parametro h

L'analisi di sensitività sul modello implementato è stata condotta utilizzando un processore core2 Duo P7350, 3 GB di RAM e 250 GB di HD. Sono stati individuati tre potenziali scenari e per ognuno di essi sono state testate una serie di istanze al variare dei parametri più incisivi ed influenzanti le performance del modello.

Per ogni *test*, sono state registrate tutte le informazioni necessarie a valutare la qualità della soluzione ottenuta e la complessità della particolare istanza trattata.

## 5 Approcci euristici risolutivi per il MLPAP

### 5.1 *Procedure euristiche e metaeuristiche*

Nell'affrontare un problema di ottimizzazione possono essere utilizzati vari approcci, a seconda, sia della difficoltà specifica e delle dimensioni del problema in esame sia degli obiettivi reali che si vogliono ottenere.

Una strada alternativa dall'utilizzo di metodi esatti, che può diventare l'unica percorribile se le dimensioni e/o la complessità del problema sono elevate, è quella che consiste nel cercare soluzioni di tipo euristico, ottenute cioè applicando un algoritmo concepito in modo da produrre soluzioni di buona qualità, ma senza garanzie a priori sulla vicinanza all'ottimo.

Un algoritmo euristico, o semplicemente un'euristica, deve essere in grado di produrre una soluzione in tempo relativamente breve. Anche se è possibile progettare euristiche specifiche per qualunque problema di Ottimizzazione Combinatoria, negli ultimi anni hanno acquisito importanza sempre maggiore alcuni approcci euristici di tipo generale, detti metaeuristiche. In questucasi, la struttura e l'idea di fondo di ciascuna metaeuristica sono sostanzialmente fissate, ma la realizzazione delle varie componenti dell'algoritmo dipende dai singoli problemi.

Tutti i diversi approcci metaeuristici possono essere considerati in modo omogeneo, come generalizzazioni di un unico approccio fondamentale, che è quello della ricerca locale. La ricerca locale si basa su quello che è, per certi versi, l'approccio più semplice ed istintivo: andare per tentativi. Va comunque detto che a fronte di questa relativa semplicità, l'applicazione di una qualunque metaeuristica ad un problema di Ottimizzazione Combinatoria richiede comunque una messa a punto accurata e talvolta molto laboriosa.

Consideriamo un problema di minimizzazione ed una sua soluzione realizzabile  $x$ , con associato il valore della funzione obiettivo  $f(x)$ . La ricerca locale consiste nel definire un intorno di  $x$  (nella terminologia della ricerca locale, detto anche vicinato), e nell'esplorarlo in cerca di soluzioni migliori, se ve ne sono. Se, in questo vicinato di  $x$ , si scopre una soluzione  $y$  per cui  $f(y) < f(x)$ , allora ci si sposta da  $x$  a  $y$  e si riparte con l'esplorazione del vicinato. Se invece nel vicinato di  $x$  non si scopre nessuna soluzione migliore, allora vuol dire che  $x$  è un minimo locale .

Nella ricerca locale classica, arrivati ad un minimo locale, l'algoritmo si ferma e restituisce questo minimo come soluzione al problema. Ovviamente, in generale, non si ha alcuna garanzia che tale valore costituisca una soluzione ottima del problema; anzi, tipicamente esso può essere anche molto

distante dall'ottimo globale. Le metaeuristiche, in effetti, sono state concepite con l'obiettivo di cercare di ovviare al problema di rimanere intrappolati in un minimo locale.

Un grosso pregio della ricerca locale sta nella sua semplicità concettuale e realizzativa, pur consentendo tale metodo di ottenere risultati molto interessanti in termini della qualità delle soluzioni fornite.

Negli ultimi venti anni la ricerca si è indirizzata verso la definizione di approcci euristici che generalizzano la ricerca locale cercando di ovviare ai suoi principali difetti e proprio da tale ricerca sono nate le tecniche metaeuristiche, oggi molto apprezzate e largamente utilizzate.

Spesso queste metaeuristiche traggono ispirazione (se non legittimazione) da alcune analogie con la natura fisica. Ad esempio, nella tecnica metaeuristica nota con il nome di *Simulated Annealing* (SA) si paragona il processo di risoluzione di un problema di Ottimizzazione Combinatoria, in cui si passa iterativamente da una soluzione realizzabile ad un'altra via via migliorando la funzione obiettivo, al processo con cui un solido, raffreddandosi, si porta in configurazioni con via via minor contenuto di energia.

Ricerca il minimo globale di una funzione di costo con molti gradi di libertà è un problema molto complesso se questa ammette un grande numero di minimi locali. Uno dei principali obiettivi dell'ottimizzazione è proprio quello di evitare di rimanere intrappolati in un minimo locale. Questo è uno dei limiti più grandi delle tecniche di ricerca locale.

I metodi basati sul SA applicano un meccanismo probabilistico che consente alla procedura di ricerca di fuggire da questi minimi locali. L'idea è quella di accettare, in certi casi, oltre alle transizioni che corrispondono a miglioramenti nella funzione obiettivo, anche quelle transizioni che portano a peggioramenti nel valore di questa funzione di valutazione. La probabilità di accettare tali deterioramenti varia nel corso del processo di ricerca, e discende lentamente verso zero. Verso la fine della ricerca, quando vengono accettati solo miglioramenti, questo metodo diventa una semplice ricerca locale. Tuttavia, la possibilità di transitare in punti dello spazio di ricerca che deteriorano la soluzione ottima corrente, consente di abbandonare i minimi locali ed esplorare meglio l'insieme delle soluzioni ammissibili.

Altra analogia con la natura fisica è palese negli Algoritmi Genetici (AG), approcci metaeuristici in cui un insieme di soluzioni realizzabili è visto come un insieme di individui di una popolazione che, accoppiandosi e combinandosi tra loro, danno vita a nuove soluzioni che, se generate in base a criteri di miglioramento della specie, possono risultare migliori di quelle da cui sono state generate.

Gli AG fanno parte della categoria dei cosiddetti *weak methods*, costituita da quei metodi di risoluzione di problemi che si basano su poche assunzioni (o conoscenze) relative alle particolari strutture e caratteristiche dei problemi stessi, motivo per il quale questi metodi sono applicabili ad una vasta classe di problemi. Gli AG rientrano pienamente in questa categoria, dal momento che essi sono in grado di compiere una efficiente ricerca anche quando tutta la conoscenza a priori è limitata alla sola procedura di valutazione che misura la qualità di ogni punto dello spazio di ricerca (misura data, ad esempio, dal valore della funzione obiettivo).

Questa caratteristica conferisce a tali algoritmi una grande robustezza, ovvero una grande versatilità che li rende applicabili a diverse categorie di problemi, al contrario dei metodi convenzionali che, in genere, non trovano altra applicazione che quella relativa alla classe di problemi per cui sono stati ideati.

Un'altra tecnica metaeuristica largamente usata nella risoluzione di problemi di Ottimizzazione Combinatoria è la *Tabu Search* (TS). Nella ricerca locale classica, ogni qual volta si esplora il vicinato (o intorno) di una soluzione, l'unica informazione relativa alla storia dell'algoritmo fino a quel momento è la migliore soluzione corrente ed il corrispondente valore della funzione obiettivo. L'idea della TS è invece quella di mantenere una memoria di alcune informazioni sulle ultime soluzioni visitate, orientando la ricerca in modo tale da permettere di uscire da eventuali minimi locali. L'idea di base della TS è dunque quella di utilizzare le informazioni sulle ultime mosse effettuate, memorizzate in una coda detta *tabu list*, per proibire quelle mosse per un certo tempo al fine di prevenire ricadute in punti già visitati, evitando così di ripercorrere soluzioni già esplorate.

La maggior parte degli analisti di OR hanno abitualmente considerato di usare le tecniche euristiche per ottenere buone soluzioni per problemi considerati troppo complessi per essere in grado di ottenere soluzioni ottime.

Tuttavia, numerosi codici commerciali, il cui scopo è o di provare l'ottimalità o di terminare non appena una soluzione si dimostra essere all'interno di una specificata tolleranza di ottimalità, applicano abitualmente algoritmi euristici durante tutta la procedura in modo da trovare prima possibile dei buoni limiti nell'algoritmo.

Così, le euristiche servono a due scopi importanti: fornire buone soluzioni a problemi per i quali algoritmi esatti sono incapaci di provare l'ottimalità in un tempo ragionevole ed aiutare un algoritmo esatto nella ricerca di una soluzione.

I metodi esatti e le tecniche di ricerca locale sono viste tradizionalmente come due approcci generali ma distinti per la risoluzione efficiente di problemi di Ottimizzazione Combinatoria, ognuno con i propri vantaggi e svantaggi.

È solo piuttosto recentemente che sono stati proposti veri algoritmi ibridi, che prendono idee da ambedue i campi cercando di combinarli al meglio per superare i rispettivi difetti. Nel caso di un problema di minimizzazione, per esempio, il più semplice approccio ibrido consiste nell'uso di un algoritmo di ricerca locale per calcolare limiti superiori che permettano una potatura immediata di soluzioni non ottime nella procedura guidata da un algoritmo esatto.

Lo sviluppo di algoritmi euristici è mirato a fornire una soluzione di buona qualità ad un problema difficile con un limitato tempo di calcolo. Nella maggior parte dei casi infatti, non si ha il tempo necessario per applicare metodi esatti proposti dai modelli matematici esposti in precedenza.

Nel corso del lavoro di tesi si è avuto modo di progettare ed implementare tre diverse euristiche per il problema di allocazione dei prodotti ai punti di stoccaggio. La prima soluzione proposta si basa sul *framework* della ricerca locale iterativa mentre la seconda soluzione utilizza la tecnica del *rollout*. Una terza soluzione è stata proposta sulla base di una strategia *ad hoc* che cerca di ridurre il decentramento delle classi di prodotto compatibili sulla base di una procedura di clusterizzazione così come descritto nelle sezioni successive.

### **5.2 La procedura di ricerca locale**

La tecnica della ricerca locale è una semplice strategia che può determinare dei buoni risultati soprattutto quando si vogliono trovare soluzioni anche non ottimali ma in tempi brevi. Affinché essa sia applicabile è necessario che, data un'istanza di un problema di ottimizzazione, sull'insieme delle soluzioni dell'istanza stessa sia definito un concetto di *neighborhood* (vicinato).

Gli algoritmi di ricerca locale si differenziano sia per le modalità con cui viene definito il vicinato, sia per le modalità di esplorazione del vicinato stesso alla ricerca di una soluzione migliore di quella corrente. A volte la ricerca locale può avere costo esponenziale o perché il vicinato ha taglia esponenziale o perché l'algoritmo effettua un numero esponenziale di iterazioni. Il caso a cui siamo interessati, ovviamente, è quello che ci permette lo "spostamento" dalla soluzione corrente  $S$  verso una nuova  $S'$  che presenta un valore di funzione obiettivo migliore, ovvero di costo minore. Sostanzialmente esistono due strategie di esplorazione del vicinato della soluzione corrente: la *first improvement* e la *best improvement*. La prima impone che non appena la ricerca trova una soluzione migliore di quella corrente, effettua lo scambio; la seconda, invece, considera tutti i possibili scambi ed effettua quello che conduce al massimo miglioramento, qualora esista.

Nel presente lavoro di tesi, è stata adottata la strategia *first improvement* in modo tale da accettare tutte le mosse che provocano una diminuzione della funzione di costo.

Per quanto riguarda la definizione delle mosse sono state individuate due diverse tipologia di mosse:

- una mossa di *remove* mediante la quale un prodotto contenuto in uno *slot*  $S_1$  viene rimosso per essere inserito in un altro *slot*  $S_2$  in grado di soddisfare i vincoli di capacità e di compatibilità;
- una mossa di *swap* mediante la quale due diversi prodotti sono scambiati nel rispetto dei vincoli di capacità e di compatibilità.

Per quanto riguarda la definizione del vicinato di una soluzione la strategia adottata è stata quella di prendere in considerazione l'insieme delle configurazioni ottenibile attraverso una operazione di *swap* o di *remove*.

Obiettivo di una mossa è quello di ridurre il numero di *slot* occupati in modo tale da evitare il decentramento dei prodotti e allo stesso tempo soddisfare i vincoli di compatibilità tra prodotti.

L'esecuzione delle mosse avviene sulla base di opportune strutture appositamente predisposte che riguardano in particolare la lista (*chain*) degli *slot* che contengono un determinato prodotto e la lista degli *slot* vuoti.

Successivamente, per ogni classe di prodotti, utilizzando le relative catene di *slot* si mettono in atto le mosse con l'obiettivo di:

- ridurre la frammentazione di una stessa classe di prodotti (vedi **Figura 19** e **Figura 20**) riducendo il numero di nodi che costituiscono la catena;
- ridurre il decentramento tra le classi compatibili in modo tale da posizionare le unità di carico di una medesima classe negli *slot* che consentono di ridurre le penalità nella funzione obiettivo privilegiando le posizioni adiacenti, le posizioni opposte e infine quelle posteriori (vedi **Figura 21**).

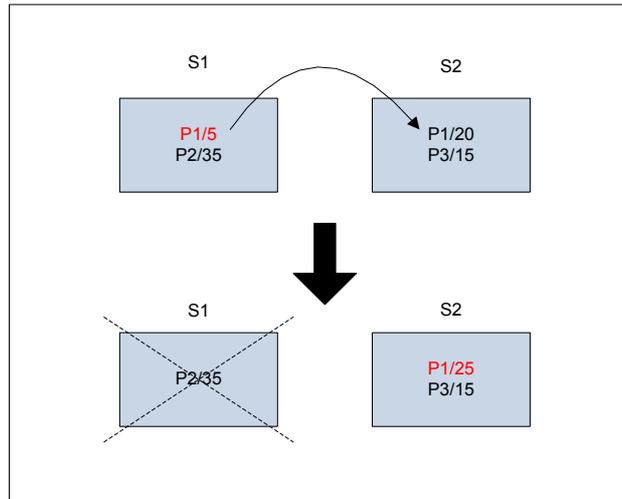


Figura 19 – Riduzione della catena di una classe di prodotti attraverso una mossa di *remove*

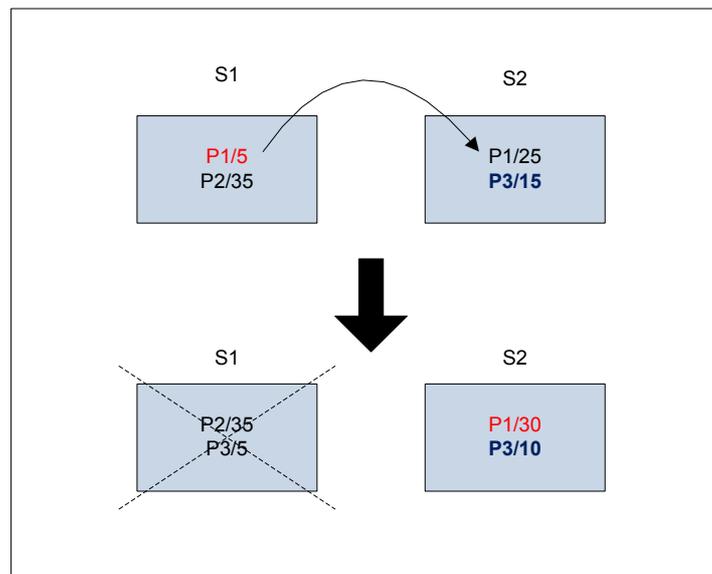


Figura 20 - Riduzione della catena di una classe di prodotti attraverso una mossa di *swap*

L'applicazione di una mossa potrebbe portare anche ad una configurazione peggiore di quella corrente dal punto di vista del costo totale. Se così fosse, allora, si annullano le mosse applicate (*rollback*) e si tenta di farne altre. Con l'obiettivo di rendere la procedura più performante, il calcolo della funzione di costo è effettuata in modo tale da prendere in considerazione solo i contributi derivanti dall'applicazione di una mossa.

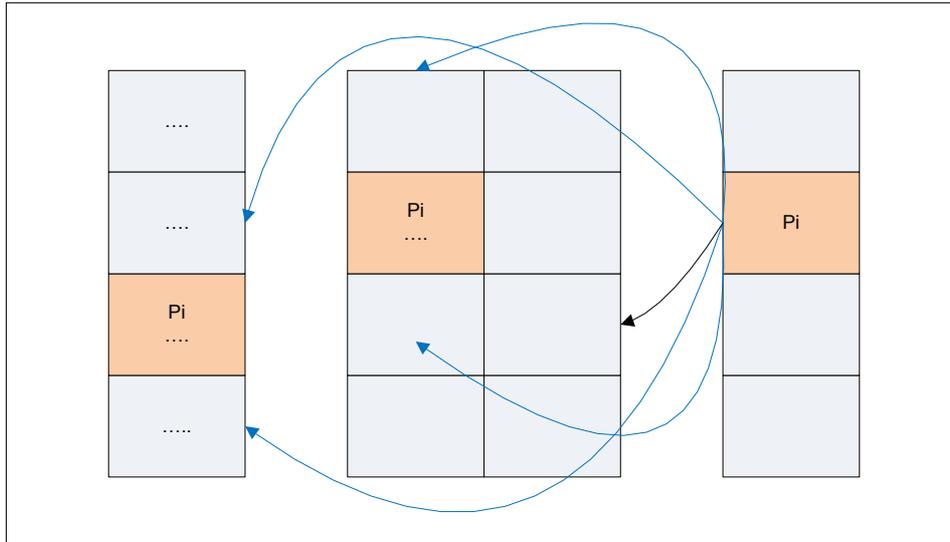


Figura 21 - Recupero delle vicinanze

### 5.3 Progettazione di un procedura euristica basata sulla Ricerca locale Iterativa

Dall'analisi dei risultati ottenuti tramite lo studio di sensitività condotto sul modello matematico, si evince che il problema di allocazione dei prodotti ai punti di stoccaggio, con vincoli di compatibilità fra le classi dei prodotti in un magazzino a più livelli, risulta essere *NP-hard*. Pertanto, non può essere risolto, in tempi ragionevoli, in maniera esatta, su istanze realistiche che risultano essere molto complesse dal punto di vista computazionale (elevato numero di vincoli e di variabili in relazione all'elevato numero di prodotti da allocare ed al numero complessivo di *slot* a disposizione). È stato, quindi, ritenuto indispensabile progettare e implementare degli algoritmi euristici al fine di risolvere, sebbene in maniera approssimata, il MLPAP su istanze di grandi dimensioni, fornendo soluzioni *sub ottime* ma comunque di buona qualità in confronto a quelle ottime, ottenute dal modello.

Nel seguito è descritta la procedura euristica implementata sulla base del *framework* della ricerca locale iterativa (ILS).

La ILS è una procedura di esplorazione del vicinato (tramite l'applicazione di mosse) che è stata introdotta per la prima volta da *Lourenço et al.* (2002).

La procedura euristica definita esplora il vicinato di una soluzione iniziale mediante l'applicazione di una serie di mosse che consentono di muoversi in maniera locale alla soluzione corrente. Al fine di superare il problema dei minimi locali, è stata definita una fase di perturbazione della soluzione corrente che, stravolgendola (completamente in alcuni casi), avvia la ricerca da una nuova e

differente soluzione. Il procedimento può essere riassunto schematicamente nei seguenti passi principali:

1. determina una soluzione iniziale  $s$  e inizializza la migliore soluzione  $s^*$  con  $s$ ;
2. applica una procedura di ricerca locale ad  $s$  per ottenere una nuova soluzione migliorata  $s'$ ;
3. fino a quando non è soddisfatto un criterio di arresto:
  - a) applica un meccanismo di perturbazione su  $s'$  per ottenere una nuova soluzione  $s''$ ;
  - b) applica il meccanismo di ricerca locale su  $s''$  per ottenere una soluzione ottima locale  $s'''$ ;
  - c) se  $s'''$  soddisfa un criterio di accettazione, sostituisci  $s'$  con  $s'''$ ;
  - d) se il costo di  $s'''$  è migliore di  $s^*$ , sostituisci  $s^*$  con  $s'''$ .

Il criterio di arresto è raggiunto quando la combinazione di mosse e di perturbazioni non migliora ulteriormente la soluzione corrente per un numero fissato di iterazioni.

Si pone in evidenza che tale schema algoritmico è del tutto generale e che si adatta al problema da risolvere mediante la specializzazione delle mosse della ricerca locale e di conseguenza della fase di perturbazione. Inoltre, mentre il processo di ricerca locale (avviato invocando la relativa procedura) è comunque sempre considerato al fine di esplorare il vicinato di una soluzione, quello di perturbazione è preso in considerazione solo se il costo della nuova soluzione, ottenuta a valle della ricerca locale, uguaglia quello della soluzione corrente. In questo caso, infatti, si è in presenza di una soluzione che rappresenta un minimo locale per il problema in esame e che dovrebbe essere “superato” per evitare che la procedura termini fornendo in *output* una soluzione distante da quella ottima. Il processo di perturbazione della soluzione corrente serve, quindi, a stravolgerne la sua struttura e ad avviare la ricerca partendo da una nuova soluzione. La procedura euristica definita è una procedura di ricerca locale in grado di fornire risultati di qualità in considerazione della complessità delle istanze reali. Essa rappresenta il giusto compromesso tra qualità delle soluzioni fornite in *output* e tempi di calcolo complessivamente richiesti. Il diagramma di flusso della procedura di ricerca locale iterativa implementata è riportato in **Figura 22**.

La soluzione iniziale è ottenuta attraverso un ordinamento di tipo *random* dell'insieme dei prodotti da allocare e l'utilizzazione dell'euristica di assegnamento di base. La fase di perturbazione è ottenuta effettuando una riassegnazione *random* dei prodotti.

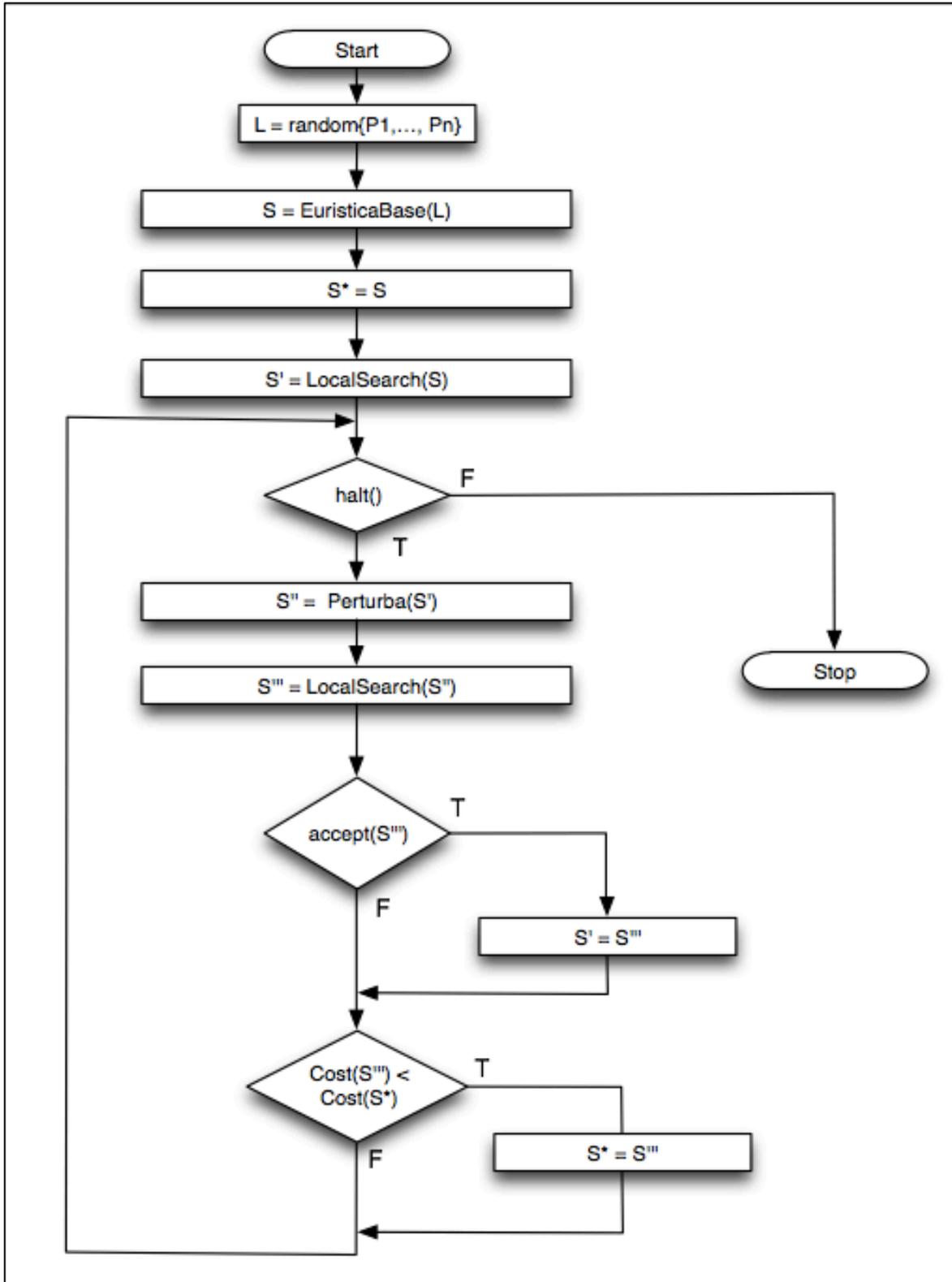


Figura 22 – Diagramma di flusso procedura di ricerca locale iterativa

#### **5.4 Progettazione di un procedura euristica basata sulla tecnica del Rollout**

In questa sezione, si intende descrivere un approccio euristico per il problema di allocazione dei prodotti, al fine di confrontare i risultati ottenuti attraverso la ILS sia in riferimento alla qualità della soluzione finale sia al tempo di elaborazione. Tale approccio alternativo si basa sull'algoritmo del *Rollout*, come di seguito dettagliato.

L'algoritmo del *Rollout* (proposto, per la prima volta, da *Bertsekas* per la soluzione di problemi di ottimizzazione combinatoria) è una procedura che, costruttivamente, determina una soluzione per un determinato problema in esame. Esso, partendo da una soluzione parziale, la amplia iterativamente. Durante la procedura costruttiva, le diverse opzioni di scelta ammissibili vengono valutate per mezzo di un algoritmo euristico B (di seguito anche indicato come euristica di base).

Pertanto, un algoritmo di *Rollout* basato sull'euristica di base B (denominato come RB) è definibile come un algoritmo che applica B sequenzialmente.

L'esperienza accumulata nell'uso del *Rollout*, soprattutto, nel campo dei problemi di ottimizzazione combinatoria, ha mostrato che questo metodo migliora significativamente le prestazioni, in termini di qualità della soluzione, dell'euristica di base impiegata e che esso è più robusto di altre tecniche come il *Tabu Search* ed il *Simulated Annealing*, dal momento che non esistono parametri da definire i cui valori possono influire sulle prestazioni dell'algoritmo.

Facendo riferimento al problema di allocazione dei prodotti, così come descritto nel capitolo precedente, l'algoritmo di *Rollout* può essere formalmente descritto come segue.

Sia  $L$  la lista delle classi di prodotti da allocare nel magazzino in esame. Si assume, inoltre, che:

- l'istanza presa in considerazione sia ammissibile (ovvero che la capacità complessiva del magazzino è tale da contenere tutte le classi) e, quindi, che esiste almeno una possibile configurazione di allocazione di questi classi anche in considerazione dei vincoli di compatibilità fra di esse;
- i prodotti della lista  $L$  siano ordinati sulla base di un criterio prefissato.

L'algoritmo di *Rollout* costruisce una configurazione di allocazione, partendo da una determinata classe, e aggiungendone, ad ogni iterazione, una nuova. Tale procedura è quindi ripetuta finché tutte le classi sono state definitivamente allocate (ovvero fino a quando la lista  $L$  risulta essere vuota).

Indicando, quindi, con  $S_m=(p_1, p_2, p_3, \dots, p_m)$  una configurazione di allocazione delle classi parziale, la generica iterazione  $m+1$  dell'algoritmo di *Rollout* può essere descritta come di seguito:

1. determina il vicinato  $V_{p_m}$  della classe  $p_m$ , ovvero l'insieme delle classi di prodotti, non ancora allocate e quindi ancora nella lista  $L$ ;
2. per ogni classe  $j$  appartenente a  $V_{p_m}$ , genera una nuova soluzione in modo tale che:
  - i. la classe  $j$  sia la prima ad essere allocata attraverso l'euristica di base  $B$ ;
  - ii. ogni altra classe  $i$  appartenente a  $V_{p_m}$  sia successivamente allocata attraverso  $B$  sino ad ottenere una completa assegnazione dei prodotti
  - iii. applicare la procedura di ricerca locale sulla soluzione completa e memorizzarne il costo della funzione obiettivo
3. dall'insieme delle soluzioni ottenute al passo precedente, scegli la classe di prodotto  $p_{m+1}$  che ha fornito la configurazione di allocazione a costo minimo e contestualmente rimuovi tale classe da  $L$ ;
4. Aggiorna la soluzione parziale corrente  $S_{m+1}=(p_1, p_2, p_3, \dots, p_m, p_{m+1})$ . Se la lista  $L$  è vuota, allora l'algoritmo fornisce in *output* tale soluzione come definitiva; altrimenti ritorna al passo 1.

Le proprietà teoriche di un algoritmo di *Rollout* dipendono fortemente dalle caratteristiche dell'euristica di base. La maggior parte delle euristiche, utilizzate per risolvere problemi di ottimizzazione combinatoria, hanno la caratteristica di essere sequenzialmente consistenti. Ad esempio, gli algoritmi di tipo "goloso" (*greedy*) spesso godono di questa proprietà.

Si può dimostrare che se l'euristica di base usata è sequenzialmente consistente allora l'algoritmo di *Rollout* progettato termina in un numero finito di iterazioni. L'approccio risolutivo progettato prevede anche l'applicazione di una procedura di Ricerca Locale, per come descritto nella precedente sezione. Ciò al fine di migliorare, per quanto possibile, il costo della soluzione finale restituita in *output*.

L'*outline*, sotto forma di digramma di flusso, dell'approccio euristico progettato è mostrato in **Figura 23**. L'insieme  $LC$  rappresenta l'insieme dei prodotti candidati ancora da assegnare mentre l'insieme  $LA$  rappresenta l'insieme dei prodotti per i quali è già stato stabilito l'ordine di assegnamento (di inserimento all'interno del *layout*). La strategia è quella di valutare, ad ogni iterazione  $i$ , l'insieme delle possibili configurazioni ottenibili interscambiano il prodotto candidato ad occupare l'*i-esima* posizione di allocazione. Un ruolo importante è svolto dalla procedura di assegnazione dei prodotti alle postazioni di stoccaggio indicata con *EuristicaBase* che si occupa di allocare i prodotti ai punti di stoccaggio utilizzando come criterio di individuazione degli *slot* sia la distanza dalle porte di *input* ed *output* sia i criteri di adiacenza delle celle.

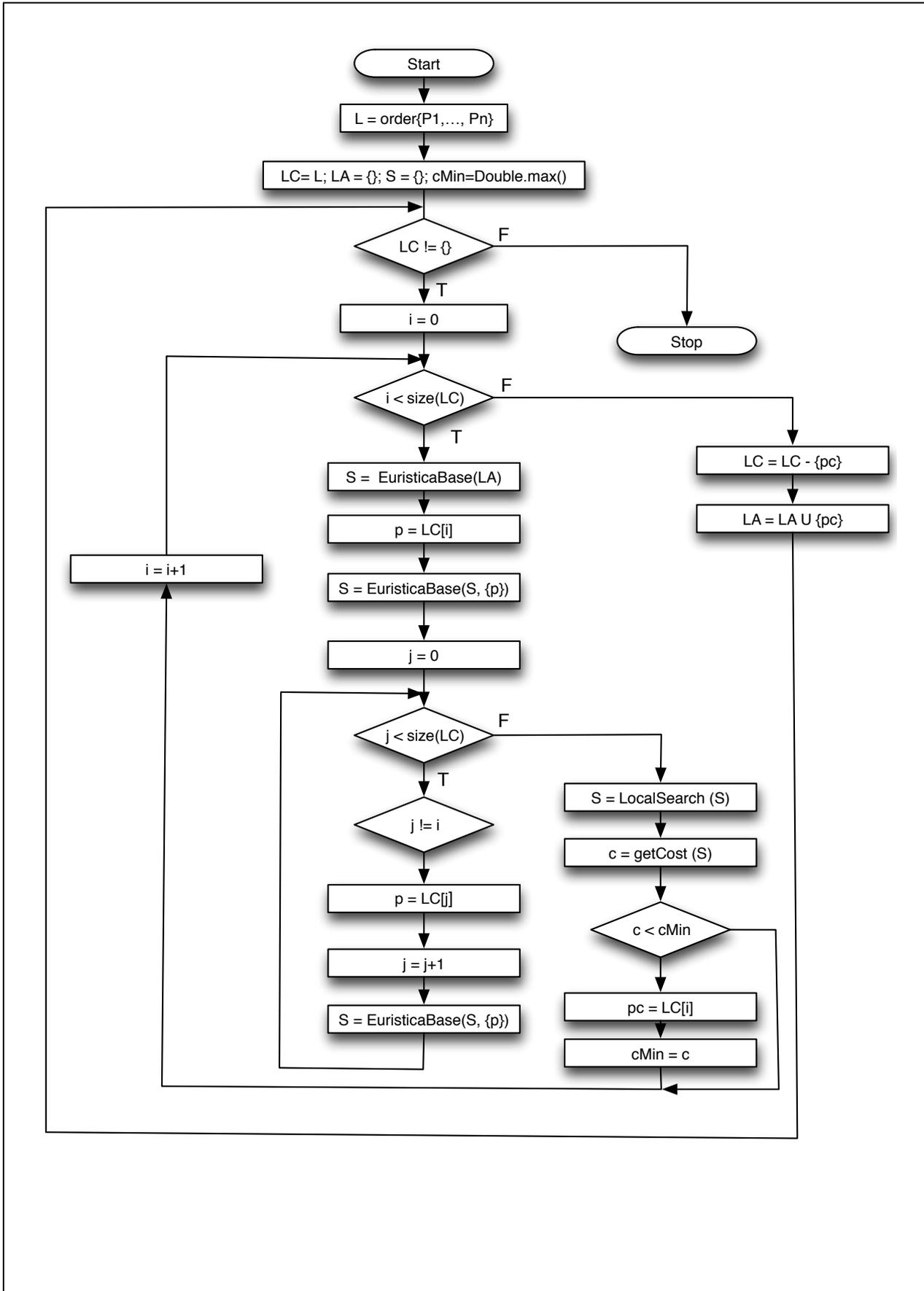


Figura 23 – Diagramma di flusso della procedura euristica di *rollout*

### 5.5 Progettazione di un procedura euristica basata su un processo di Clustering

La procedura euristica proposta in questa sezione è stata realizzata sulla base di favorire l'aggregazione delle classi di prodotto tra loro compatibili attraverso un processo di *clustering* teso ad individuare le partizioni dell'insieme di prodotti caratterizzati da prodotti compatibili.

Per capire meglio il filo conduttore di questa progettazione, a più alto livello, si supponga di avere 4 classi di prodotti {P1,P2,P3, P4} e la seguente matrice di compatibilità:

$$C = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

Ciò significa che P1 non è compatibile con P2; P2 non è compatibile con P4. In una semplice procedura di allocazione iniziale di prodotti, si potrebbe generare la seguente situazione sfavorevole per come indicato in **Figura 24**.

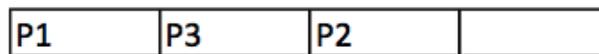


Figura 24 – configurazione sfavorevole

Ogni volta che una classe deve essere allocata si cerca, partendo sempre dall'inizio del blocco, la prima postazione libera ammissibile (tenendo, cioè, conto della compatibilità). Quindi, ad esempio, P2 non è stata allocata nella seconda postazione perché non compatibile con P1, postazione che, invece, è stata correttamente occupata da P3. Tuttavia, la classe P4 non può essere né allocata nell'ultima postazione né, qualora ci fosse spazio libero, nella seconda postazione (ammettendo che non ci sia spazio residuo nella prima). Analizzando, però, visivamente la situazione, è facile individuare una migliore strategia di allocazione, come indicato in **Figura 25**. Quindi, si conclude che la precedente strategia di allocazione non tiene in giusta considerazione la minimizzazione delle cosiddette discontinuità, ovvero situazioni di “buchi”, postazioni libere non occupate da alcuna classe di prodotto (come in **Figura 24**). La causa principale risiede nel fatto che la procedura non è in grado di gestire e manipolare opportunamente le condizioni di compatibilità (la procedura è, infatti, definita di tipo *blind*).

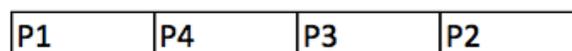


Figura 25 – configurazione favorevole

Alla luce di quanto detto, la procedura di allocazione dei prodotti può essere riassunta nei seguenti passi:

1. Clusterizzazione delle classi;
2. Costruzione del grafo ausiliario GA;
3. Individuazione delle componenti connesse di GA;
4. Per ogni componente connessa, visita in profondità del grafo (ed allocazione dei prodotti mediante l'utilizzo dell'euristica base) scegliendo il nodo figlio sulla base della domanda dei nodi figli. Il nodo radice è quello a domanda maggiore.

**Clusterizzazione delle classi.** Si definisce *cluster* di classi di prodotti o di prodotti, un insieme di classi fra loro compatibili. E' evidente che una classe potrebbe, teoricamente afferire anche a più di un *cluster*. Nell'esempio indicato prima, quindi, si individua un primo *cluster*  $C1=\{P1, P3, P4\}$  in quanto queste tre classi sono fra loro compatibili e  $C2=\{P2, P3\}$  in quanto fra loro compatibili. Più formalmente, quindi, un *cluster*  $Cl$  è definito come segue:

$$Cl = \{i, j, \dots, k | i, j, \dots, k \in N \wedge cij = \dots = cik = 1\}$$

Pertanto, l'intersezione di una coppia di *cluster* ( $Cl_1, Cl_2$ ) è definita come segue:

$$Cl_1 \cap Cl_2 = \overline{Cl_1} \cup \overline{Cl_2}$$

dove  $\overline{Cl_1}$  e  $\overline{Cl_2}$  sono i rispettivi sottoinsiemi tali che esiste almeno un prodotto del primo compatibile con almeno uno del secondo. Nell'esempio:

$$C1 \cap C2 = \overline{C1} \cup \overline{C2} = \{P1, P2, P3, P4\}$$

**Costruzione del grafo ausiliario.** Si definisce *grafo ausiliario non orientato*  $G=<V, E>$  un grafo in cui esiste un vertice per ogni *cluster* individuato al passo precedente ed esiste uno spigolo per ogni coppia di *cluster* se e solo se la loro intersezione è non vuota. Nell'esempio,  $V=\{C1, C2\}$  e  $E=\{(C1, C2)\}$ , come indicato in **Figura 26**:

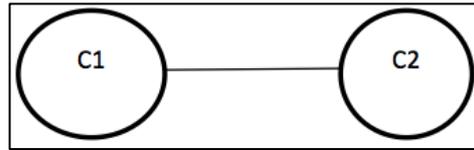


Figura 26 – esempio di grafo ausiliario

**Individuazione delle componenti connesse.** Si definisce componente connessa un sottografo massimale di  $G$ . Un sottografo connesso  $H$  di  $G$  è “massimale” se non si possono aggiungere ad  $H$  altri vertici o archi in modo che l’ $H$  risultante sia ancora un sottografo connesso di  $G$  (**Figura 27**). Per la ricerca delle componenti connesse in un grafo non orientato è possibile usare una visita in profondità del grafo stesso.

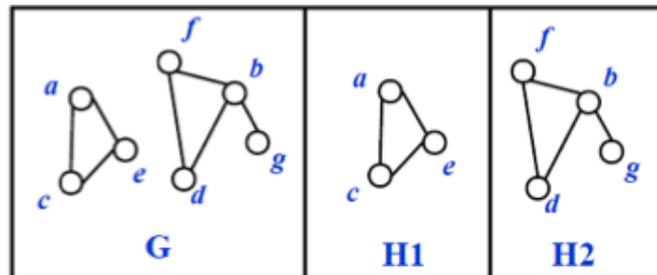


Figura 27 – Esempio di componenti connesse

Ad esempio, nel caso dei *cluster*, si potrebbe avere una situazione come mostrata in **Figura 28**.

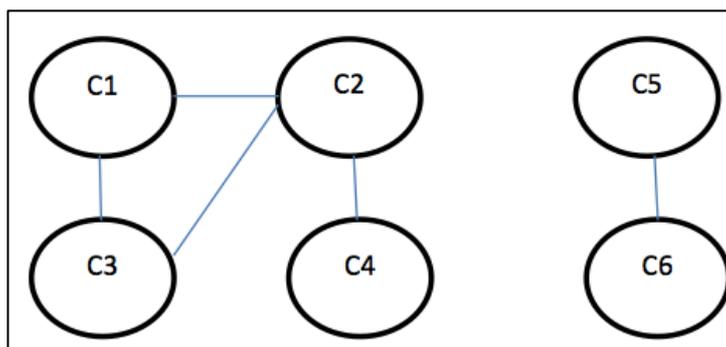


Figura 28 - Esempio di componenti connesse nella clusterizzazione

In questo caso, si sono individuate due componenti connesse  $G' = \langle V', E' \rangle$  e  $G'' = \langle V'', E'' \rangle$  dove  $V' = \{C1, C2, C3, C4\}$  e  $E' = \{(C1, C2), (C2, C3), (C1, C3), (C2, C4)\}$  mentre  $V'' = \{C5, C6\}$  e  $E'' = \{(C5, C6)\}$ . E' evidente che ogni coppia di componente connessa identifica una discontinuità nell'allocazione delle classi a scaffale che comunque non può essere superata ed evitata. Quindi, nel

caso in esame avremo una discontinuità.

**Per ogni componente connessa, visita in profondità del grafo.** Per ciascuna delle componenti connesse così individuate, si definisce la sequenza di visita di ciascun nodo, che corrisponde anche alla strategia di allocazione iniziale delle classi in magazzino, effettuando una visita in profondità del grafo e scegliendo sempre il nodo figlio a domanda maggiore rispetto agli altri nodi figli di un generico nodo. Il nodo radice è scelto come quello a domanda maggiore. Con riferimento ad una componente connessa  $\{C1, C2, C3, C4\}$  riportata in **Figura 29**, se la domanda totale del *cluster* C1 è pari a 10; di C2 è pari a 8; di C3 è pari a 6 e di C4 è pari a 4, la visita in profondità, partendo dalla radice C1, avendo domanda maggiore, genererebbe l'albero in **Figura 30**.

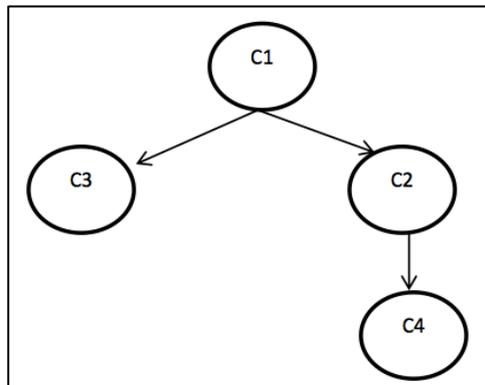


Figura 29 - Minimo albero ricoprente

Tuttavia, il nodo figlio è sempre selezionato come quello a domanda maggiore, pertanto si ottiene:

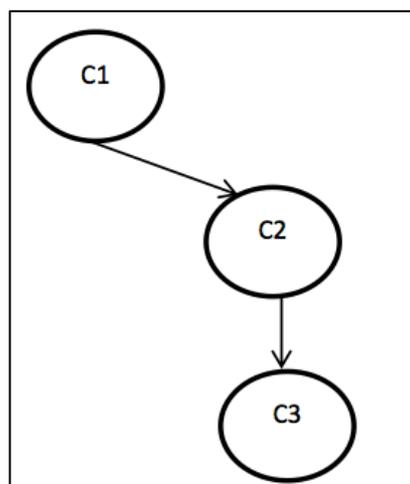


Figura 30 – Configurazione finale

In questo caso, si evince che, comunque, almeno una discontinuità è inevitabile. D'altronde, si tratta di una procedura di inizializzazione la cui soluzione rappresenta la fase di avvio di una ricerca migliorativa attraverso l'utilizzo della procedura di ricerca locale.

## 6 Analisi dei risultati sperimentali

Al fine di verificare la qualità delle soluzioni fornite in *output* dalle tre diverse procedure euristiche realizzate è stata condotta una approfondita analisi dei risultati ottenuti rispetto sia al valore della funzione obiettivo sia ai tempi di elaborazione.

Per ognuno degli scenari individuati (scenario S, MC e C) sono state testate una serie di istanze al variare dei parametri più incisivi ed influenzanti le *performance* del modello.

L'insieme completo dei risultati computazionali dello studio effettuato è riportato in Appendice A - Risultati sperimentali.

### 6.1 Analisi delle prestazioni in termini della funzione di costo

Per ogni *data set* utilizzato nella sperimentazione si è avuto modo di calcolare il *gap*, in termini percentuali, tra il costo della funzione obiettivo ed il valore ottenuto attraverso il solutore CPLEX, ovvero valutando il rapporto:

$$\frac{\text{costo totale euristica} - \text{costo totale cplex}}{\text{costo totale cplex}} * 100$$

Si riporta in **Tabella 13** il quadro sintetico dei risultati ottenuti (in valore assoluto), dal quale si evince che le euristiche implementate forniscono soluzioni di buona qualità che mediamente migliorano le soluzioni sub ottime ottenute dal risolutore CPLEX.

Parametro	CPLEX	ILS	Rollout FlussoMedio	Rollout Domanda	Rollout Random	Cluster
Numero prodotti	43.535,34	42.964,33	42.611,50	43.267,40	42.604,80	43.417,03
Capacità slot	204.954,70	204.207,77	199.523,46	199.817,48	198.593,02	201.473,10
W (numero di slot in orizzontale)	38.082,22	33.039,29	32.813,90	33.118,46	32.819,84	33.426,48
H (numero di livelli)	31.794,39	29.855,24	29.646,34	30.019,92	29.644,29	30.133,86
M (numero di slot di verticale)	44.507,72	41.489,07	41.172,74	41.702,30	41.170,83	41.342,34

Tabella 13 – Average gap tra cplex ed euristica

In particolare, così come evidenziato dai grafici riportati in **Figura 31**, l'euristica di *rollout* con un ordinamento iniziale dei prodotti di tipo casuale è la strategia che consente di minimizzare il costo di assegnamento dei prodotti.

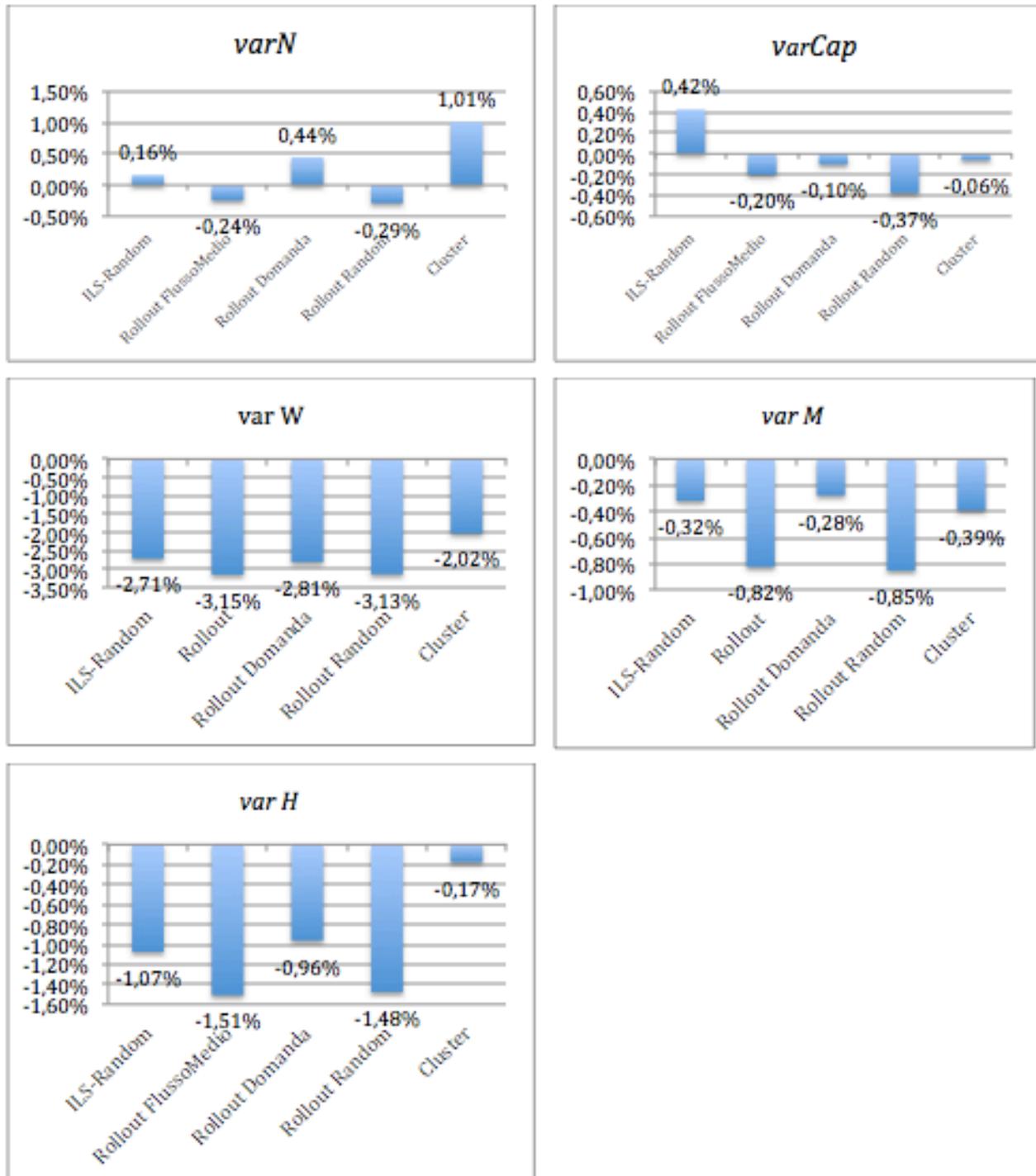


Figura 31 - Average gap tra *cplex* e procedura euristica

La tecnica del *rollout* risulta essere la strategia migliore indipendentemente dallo scenario considerato così come evidenziato in **Figura 32**. Nel caso dello scenario C è possibile osservare che l'euristica che fornisce un miglior valore di *average gap* è l'euristica di *Rollout* con le classi di prodotto ordinate sulla base del flusso medio di movimentazione.

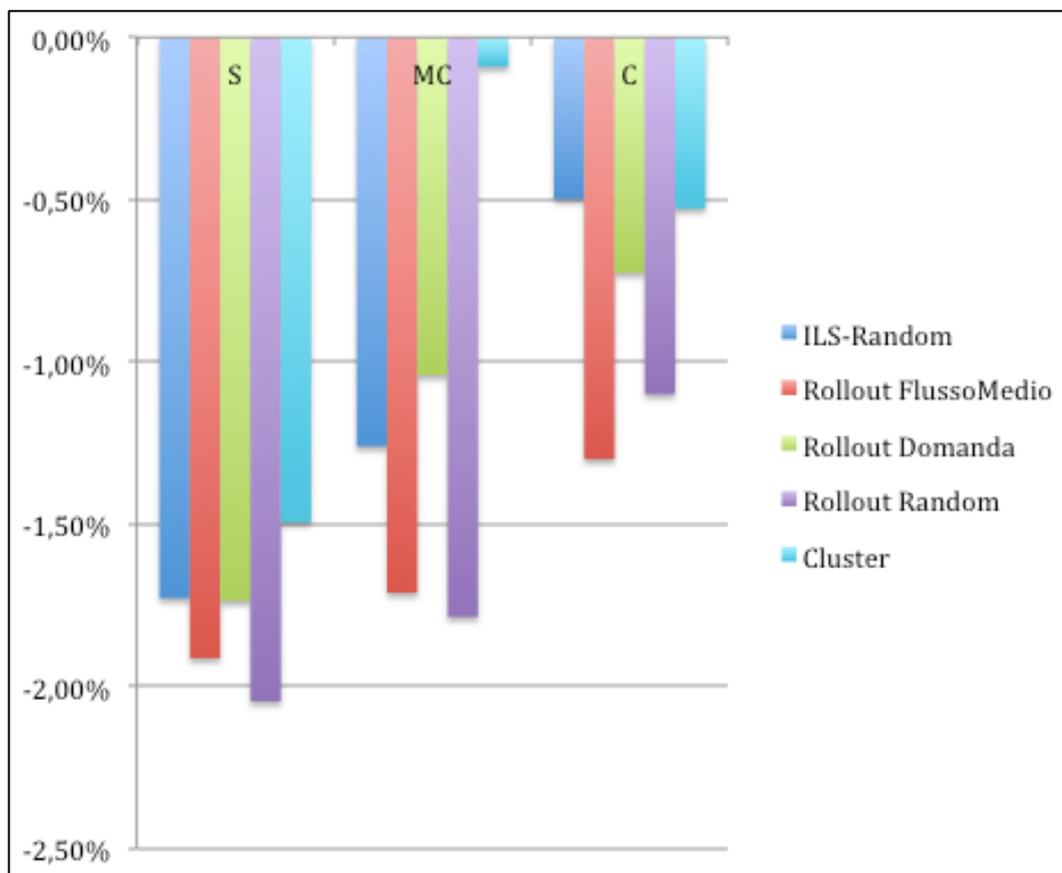


Figura 32 – Average gap per scenario

Per quanto riguarda la strategia ILS si è avuto modi valutare i benefici introdotti dalla ricerca locale effettuando il confronto sul *gap* percentuale tra il costo della soluzione iniziale ed il costo della soluzione finale ottenuta attraverso l'utilizzo della procedura di ricerca locale.

Nelle Figura 33 e Figura 34 è possibile valutare il *gap* percentuale tra le due soluzioni nel caso di due diverse insiemi di *dataset* (significati per lo scenario, il parametro variabile ed il numero di prodotti utilizzati).

dataset	CPLEX	Costo iniziale	CostoFinale	gap iniziale %	gap finale %
C_N100_H4_data_set.txt	78.642,00	83.334,00	76.343,40	5,97%	-2,92%
C_N100_H5_data_set.txt	57.102,60	69.062,60	56.170,00	20,94%	-1,63%
C_N100_H6_data_set.txt	58.847,39	71.427,39	55.650,00	21,38%	-5,43%
C_N100_H7_data_set.txt	58.867,79	66.859,79	57.287,60	13,58%	-2,68%

C_N100_H8_data_set.txt	57.357,00	63.344,00	57.045,20	10,44%	-0,54%
C_N100_H9_data_set.txt	58.739,40	63.705,40	57.051,00	8,45%	-2,87%
C_N100_H10_data_set.txt	57.801,00	66.725,00	56.949,80	15,44%	-1,47%

In particolare, è possibile osservare che la ricerca locale influisce in maniera decisiva sulla qualità della soluzione iniziale ottenuta attraverso l'utilizzo dell'euristica base. Infatti, mentre il costo della soluzione iniziale così ottenuta è sempre peggiorativa rispetto alla soluzione *sub* ottima fornita da *cplex*, la soluzione finale, nel caso dello scenario complesso con parametro variabile H e N=100, evidenzia un average *gap* rispetto a *cplex* di circa il 2.50%.

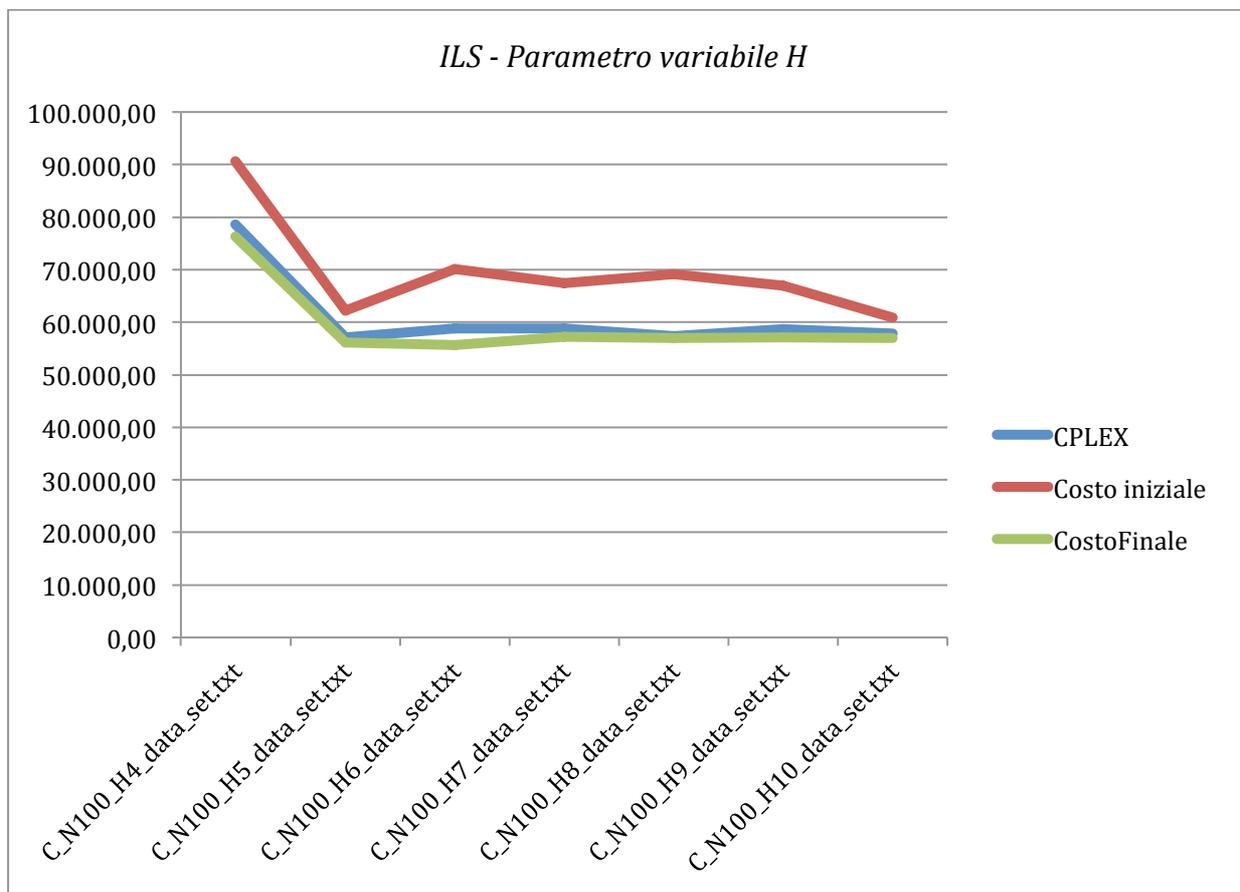


Figura 33 - Confronto tra la soluzione iniziale ed finale della ILS

Per quanto riguarda l'analisi delle istanze dello scenario complesso con parametro variabile W e N=100 si è avuto modo di registrare un guadagno medio rispetto a *cplex* di circa il 3.7%.

Dataset	CPLEX	Costo iniziale	Costo finale	gap iniziale %	gap finale %
S_N100_W6_data_set.txt	59.641,60	63.688,60	56.550,60	6,79%	-5,18%
S_N100_W8_data_set.txt	59.308,20	64.062,20	57.810,20	8,02%	-2,53%

## Risultati sperimentali

S_N8100_W10_data_set.txt	58.497,00	65.816,00	55.390,00	12,51%	-5,31%
S_N100_W12_data_set.txt	60.833,40	65.150,40	58.342,40	7,10%	-4,09%
S_N100_W14_data_set.txt	69.306,63	73.933,63	66.613,63	6,68%	-3,89%
S_N100_W16_data_set.txt	70.216,32	78.967,32	67.371,32	12,46%	-4,05%
S_N100_W18_data_set.txt	72.804,86	77.131,86	69.368,86	5,94%	-4,72%
C_N100_W6_data_set.txt	73.104,11	82.757,11	71.648,11	13,20%	-1,99%
C_N100_W8_data_set.txt	73.906,45	84.779,45	70.553,45	14,71%	-4,54%
C_N100_W10_data_set.txt	75.306,98	79.263,98	71.399,98	5,25%	-5,19%
C_N100_W12_data_set.txt	82.140,41	88.476,41	78.670,41	7,71%	-4,22%
C_N100_W14_data_set.txt	83.549,32	91.592,32	81.343,32	9,63%	-2,64%
C_N100_W16_data_set.txt	85.776,76	91.678,76	83.733,76	6,88%	-2,38%
C_N100_W18_data_set.txt	89.294,43	101.281,43	87.195,43	13,42%	-2,35%

E' possibile, quindi, notare come la Ricerca Locale sia efficace anche rispetto ai tempi di elaborazione così come evidenziato nella sezione successiva.

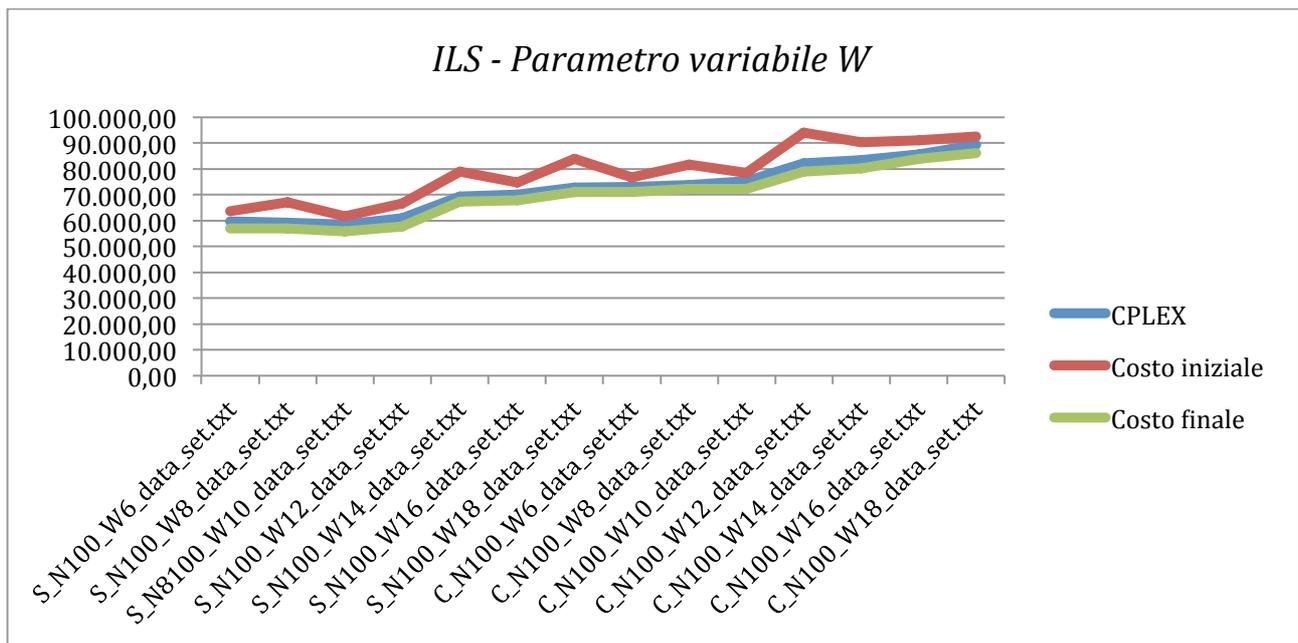


Figura 34 – Confronto tra la soluzione iniziale ed finale della ILS

## 6.2 Analisi dei tempi di elaborazioni

Per quanto riguarda i tempi di elaborazione è possibile osservare come il tempo di calcolo richiesto dalle procedure euristiche è nettamente inferiore rispetto al risolutore *cplex*, con uno scostamento (*tempo di calcolo cplex – tempo di calcolo euristica*) elevato (vedi **Tabella 15**) che si riduce man mano che ci si avvicina alle soluzioni ottime calcolate da *CPLEX*.

Tempi medi di elaborazione (ms)						
Parametro	CPLEX	ILS-Random	Rollout FlussoMedio	Rollout Domanda	Rollout Random	Cluster
Numero prodotti	3.913.645,28	316,56	28.433,28	27.292,28	27.255,67	7.027,22
Capacità <i>slot</i>	5.042.819,79	222,79	1.150,50	960,50	1.068,13	3.104,75
W (numero di <i>slot</i> in orizzontale)	3.993.639,28	567,02	64.920,78	52.295,68	54.166,97	34.443,24
H (numero di livelli)	3.715.171,08	456,34	39.796,22	37.535,72	33.249,71	17.389,28
M (numero di <i>slot</i> di verticale)	3.697.567,56	517,38	50.127,34	48.234,63	47.283,12	19.869,12

Tabella 14 – Tempi medi di elaborazione

Tramite il quadro completo delle informazioni, si può affermare, quindi, che gli approcci euristici forniscono soluzioni di buona qualità, soprattutto quando si tratta di risolvere istanze complesse, per le quali il tempo di calcolo impiegato da *CPLEX* è nettamente superiore rispetto ai tempi registrati per le euristiche realizzate.

Average Gap tempi di calcolo					
Parametro	ILS-Random	Rollout FlussoMedio	Rollout Domanda	Rollout Random	Cluster
Numero prodotti	-99,99%	-99,27%	-99,30%	-99,30%	-99,82%
Capacità <i>slot</i>	-100,00%	-99,98%	-99,98%	-99,98%	-99,94%
W (numero di <i>slot</i> in orizzontale)	-99,99%	-98,37%	-98,69%	-98,64%	-99,14%
H (numero di livelli)	-99,99%	-98,93%	-98,99%	-99,11%	-99,53%
M (numero di <i>slot</i> di verticale)	-99,99%	-98,64%	-98,70%	-98,72%	-99,46%

Tabella 15 – Average gap tempi di calcolo rispetto a *cplex*

Un confronto diretto dei tempi di calcolo delle diverse euristiche utilizzate evidenzia come la procedura basata sul *framework* di ricerca locale sia la tecnica che permette di ottenere le prestazioni migliori in termini di tempi di calcolo. Infatti, risulta essere la procedura che fornisce sempre il minor tempo di calcolo. Si riporta, in **Figura 35**, il *gap* percentuale tra il valore medio di elaborazione delle euristiche rispetto al minimo valore medio di calcolo.

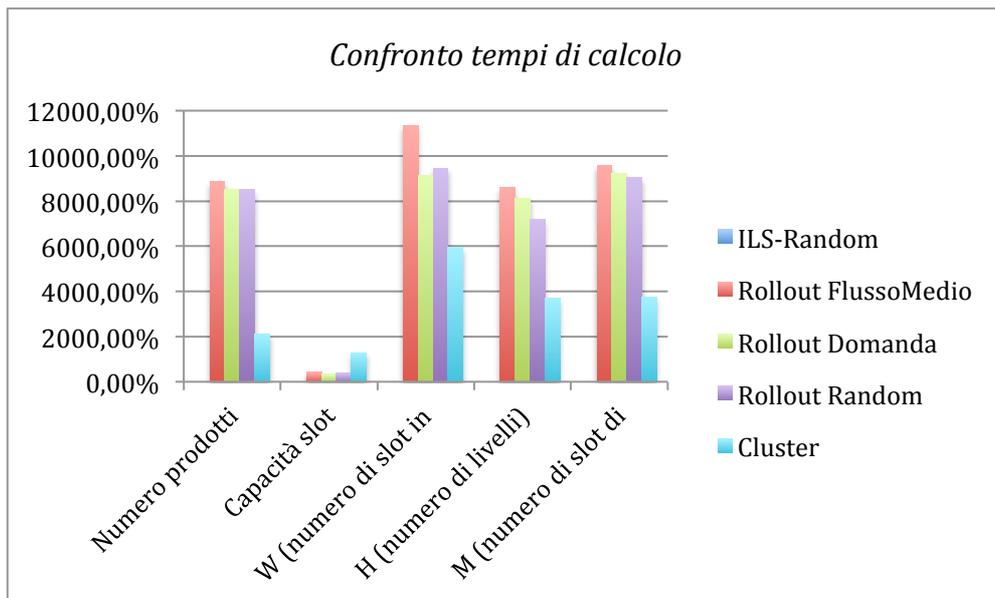


Figura 35 – Confronto tempi medi di calcolo euristiche

E' importante sottolineare che i tempi riportati nella tabella 14 sono relativi ai tempi in cui si riscontra per la prima volta il valore minimo.

A causa della particolare struttura della tecnica di *rollout* i valori rilevati hanno evidenziato il *gap* esistente con la procedura euristica basata sulla tecnica della ricerca locale iterativa. Tale distanza è tale da indurre da una parte a valutare il *trade off* tra la qualità della soluzione fornita ed i tempi di calcolo e dall'altra di esplorare strategie alternative per migliorare i tempi di calcolo del *rollout*.

## 7 Conclusioni e sviluppi futuri

Obiettivo del lavoro di è stato quello di formalizzare il problema dell'allocazione dei prodotti ai punti di stoccaggio in un magazzino multilivello e con vincoli di compatibilità tra le diverse classi di prodotto. I vincoli operativi del problema rappresentato sono stati individuati sulla base delle specifiche esigenze delle aziende che operano nel settore della grande distribuzione.

La versione generale del problema di allocazione dei prodotti ai punti di stoccaggio, nota in letteratura come *Storage Location Assignment Problem* o, equivalentemente, come *Product Allocation Problem*, rappresenta, insieme a quello della gestione efficiente dell'inventario, uno dei *topic* più importanti nell'ambito del *Warehouse Managament*. Lo scopo principale è quello di ridurre, per quanto possibile, i tempi di consegna degli ordini; le scorte in magazzino; i costi logistici complessivi e garantire, allo stesso tempo, più elevati livelli di servizio (e di conseguenza un elevato grado di soddisfazione dei clienti finali).

Il modello matematico proposto nel lavoro di tesi si prefigge la minimizzazione del *trade off* tra i costi di movimentazione dei prodotti e il decentramento di questi a scaffale, e può ritenersi una rivisitazione e un'estensione del modello proposto in Sanei et al. (2011), attraverso la formalizzazione matematica di due aspetti aggiuntivi: il multi-livello (con riferimento al *layout* di magazzino) e i vincili di compatibilità tra le classi di prodotti trattati.

Il modello matematico è stato implementato nel linguaggio di programmazione Java in ambiente di sviluppo *Eclipse*, in modo tale da essere risolto tramite il software di ottimizzazione *ILOG CPLEX*.

Non esistendo in letteratura, ad oggi, un insieme di *data set* per il problema studiato, si è avuto modo di generare, per mezzo di una apposita procedura *Java*, un insieme di istanze al fine di effettuare una scrupolosa analisi di sensitività sul modello matematico lineare. Grazie a questa analisi condotta, si è potuto constatare che il PAP in esame presenta le tipiche caratteristiche di un problema *NP-hard*.

Infine, le attività di ricerca si sono concentrati sulla progettazione di tre diversi strategie risolutive, alternative al metodo esatto, basati sulla ricerca locale, che permettono di risolvere il problema in tempi di calcolo ridotti rispetto al risolutore *CPLEX*, anche su istanze particolarmente complesse, e che forniscono risultati di buona qualità.

In particolare, sono state oggetto di progettazione tre diverse procedure euristiche sulla base di tre diversi *framework*:

- la prima euristica proposta è stata ispirata dalla strategia alla base della ricerca locale iterativa che si prefigge di superare il problema dei minimi locali attraverso l'implementazione di una specifica procedura in grado di effettuare una perturbazione della soluzione corrente;
- sulla base della strategia basata sulla tecnica del *rollout*, si è avuto modo di progettare una meta euristica in grado di ottenere soluzioni di buona qualità sulla base di un processo iterativo che consente di valutare, per ogni fase del processo di *rollout*, la migliore soluzione sulla base dei diversi scenari generati;
- un terzo approccio è stato ispirato dalla filosofia organizzativa che considera la compatibilità tra i prodotti il fattore di familiarità, (*Family Grouping*); sostanzialmente consiste nel creare *cluster* di prodotti o classi di prodotto, fra di loro compatibili per ottenere una soluzione iniziale che cerca di massimizzare la concentrazione dei prodotti compatibili all'intero del *layout*.

Dall'analisi dei dati sperimentali è possibile evincere che la tecnica del *rollout* consente di raggiungere i migliori risultati in termini di valore della funzione obiettivo, mentre risulta essere particolarmente oneroso rispetto ai tempi di calcolo soprattutto se confrontati con i tempi della ricerca locale iterativa. La tecnica del *rollout* rappresenta la miglior soluzione indipendentemente dal criterio iniziale di allocazione dei prodotti. Inoltre, per quanto riguarda la tecnica del *rollout* si è avuto modo di verificare come l'ordinamento dei prodotti iniziale sia poco influente sui risultati.

La tecnica di cluterizzazione si è rilevata poco efficace per il problema in esame sia rispetto alla ricerca locale iterativa sia rispetto alla tecnica di *rollout*, anche se in quest'ultimo caso ha fatto registrare tempi di elaborazione inferiori.

I benefici derivanti dall'utilizzo di approcci euristici sono riscontrabili sia nello scenario semplice, mediamente complesso e complesso. Lo scenario complesso è contraddittorio da un elevato grado di incompatibilità tra le classi di prodotto.

Per quanto riguarda gli sviluppi futuri è in fase di valutazione la possibilità di implementare la procedura di *rollout* su architetture parallele in modo tale da poter ridurre in maniera significativa i tempi di calcolo. Inoltre, per quanto riguarda la ricerca locale iterativa, l'obiettivo è quello di individuare una strategia di individuazione della soluzione iniziale ammissibile a partire da soluzioni non ammissibili. Attraverso l'individuazione di specifici criteri (ad esempio ordinamento delle classi di prodotto) da utilizzare nella fase di allocazione iniziale dei prodotti e opportune mosse locali l'obiettivo è quello di raggiungere una soluzione iniziale ammissibile di buona qualità.

Un ulteriore sviluppo possibile riguarda il trattamento di dati incerti sulla domanda di prodotti in magazzino. In questo caso, si tratta di ipotizzare che il numero di posizioni di stoccaggio richieste per ciascun prodotto non siano note con certezza.

Per rappresentare tale incertezza, si può utilizzare un insieme discreto di variabili aleatorie le cui possibili realizzazioni congiunte costituiscono i cosiddetti scenari, ognuno caratterizzato da una specifica proprietà di occorrenza. Sulla base di questi scenari sono allo studio possibili decisioni di ricorso che rappresentano le reazioni all'osservazione dell'evoluzione di parametri in incerti.

## 8 Appendice A - Risultati sperimentali

Le sperimentazione è stata condotta su un elaboratore che utilizza un processore core2 *Duo P7350*, *3 GB di RAM* e *250 GB di HD*.

Sono stati individuati tre potenziali scenari e per ognuno di essi sono state testate una serie di istanze al variare dei parametri più incisivi ed influenzanti le performance del modello.

Per ogni *test*, sono state registrate tutte le informazioni necessarie a valutare la qualità della soluzione ottenuta e la complessità della particolare istanza trattata.

Nelle sezioni sottostanti sono riportati i risultati computazionali dello studio effettuato sulla base dei *data set* individuati nella fase di analisi della sensibilità del modello con l'obiettivo di mettere a confronto i risultati ottenuti con i risultati forniti dal solutore *CPLEX*.

### 8.1 *Variazione del numero di classi di prodotti*

Sono stati ottenuti i seguenti valori della funzione obiettivo:

VarN – Funzione obiettivo						
dataset	CPLEX	ILS-Random	Rollout FlussoMedio	Rollout Domanda	Rollout Random	Cluster
s_5_data_set.txt	3856,19	3856,2	3930,6	3856,2	3856,2	3856,2
s_10_data_set.txt	7745,39	7884,6	7884,6	7884,6	7884,6	7884,6
s_20_data_set.txt	16616,39	16808,4	16904,4	16725,6	16858,2	16783,8
s_50_data_set.txt	45881,4	46900,2	46264,8	46696,8	46417,2	47068,2
s_80_data_set.txt	78247,8	79420,8	78671,4	80252,4	79311,6	80569,8
s_100_data_set.txt	100862,4	102271,8	101645,4	102954	101026,2	102964,2
c_5_data_set.txt	3881,39	3881,4	3881,4	3881,4	3944,4	3888,6
c_10_data_set.txt	7944,59	8038,2	7914	7922,4	7922,4	7916,4
c_20_data_set.txt	16835,99	16861,2	16822,8	16846,2	16851,6	17029,8
c_50_data_set.txt	46468,79	47014,8	46877,4	47284,2	46545,6	47583,6
c_80_data_set.txt	79641,59	80172	79045,2	82023,6	79157,4	81041,4
c_100_data_set.txt	102031,8	102516,6	101746,2	103117,8	101645,4	104005,8
mc_5_data_set.txt	3869,39	3871,2	3871,2	3881,4	3871,2	4084,8
mc_10_data_set.txt	7784,99	7908	7932,6	7911,6	7908	7942,2
mc_20_data_set.txt	16669,2	16818,6	16887,6	16926	16828,8	16936,8
mc_50_data_set.txt	45994,2	47122,2	46485,6	47056,2	46468,2	47422,8
mc_80_data_set.txt	78642	79881	78902,4	80395,8	79416	79866
mc_100_data_set.txt	120662,59	102130,8	101339,4	103197	100973,4	104661,6
<i>valore medio</i>	<b>43535,33833</b>	<b>42964,33</b>	<b>42611,5</b>	<b>43267,4</b>	<b>42604,8</b>	<b>43417,03</b>

Per quanto riguarda i tempi di elaborazione sono stati rilevati i seguenti valori:

VarN - Tempi di elaborazione (ms)						
dataset	CPLEX	ILS-Random	Rollout FlussoMedio	Rollout Domanda	Rollout Random	Cluster
s_5_data_set.txt	875	385	250	172	184	1101
s_10_data_set.txt	2750	154	13	148	252	1332
s_20_data_set.txt	46000	189	1266	1022	1320	2085
s_50_data_set.txt	7179430	420	14839	14989	17085	7109
s_80_data_set.txt	7162090	558	58095	60635	57862	14039
s_100_data_set.txt	7159310	722	114120	107710	97482	18698
c_5_data_set.txt	2060	56	15	2	7	498
c_10_data_set.txt	47310	95	133	152	171	953
c_20_data_set.txt	5675560	128	1039	894	1046	2179
c_50_data_set.txt	7188480	301	9035	13079	12694	7169
c_80_data_set.txt	7172340	430	35152	32279	50520	13706
c_100_data_set.txt	7157370	559	102801	95297	85299	17748
mc_5_data_set.txt	1230	49	22	2	6	395
mc_10_data_set.txt	8460	81	144	55	28	931
mc_20_data_set.txt	105230	139	157	1054	338	1654
mc_50_data_set.txt	7175600	381	13634	6700	15180	6986
mc_80_data_set.txt	7183980	468	57745	55917	54472	12278
mc_100_data_set.txt	7177540	583	103339	101154	96656	17629
<i>valore medio</i>	<b>3.913.645,28</b>	<b>316,56</b>	<b>28.433,28</b>	<b>27.292,28</b>	<b>27.255,67</b>	<b>7.027,22</b>

## 8.2 *Variatione della capacità degli slot*

Sono stati ottenuti i seguenti valori della funzione obiettivo:

dataset	CPLEX	ILS-Random	Rollout FlussoMedio	Rollout Domanda	Rollout Random	Cluster
S_N20_CAP11_data_set.txt	790811,36	767946,2	791216,55	791340,58	791227,5	790599,45
S_N20_CAP20_data_set.txt	399520,01	394019,61	399386,57	399337,97	399325,54	399046,1
S_N20_CAP30_data_set.txt	248257,12	247378,76	248412	248432,07	248447,17	248324,05
S_N20_CAP50_data_set.txt	97500,41	97655,96	97617,29	97688,82	97669,9	97669,59
S_N20_CAP80_data_set.txt	46827,8	47111,47	46989,37	47008,39	47019,18	47102,85
S_N20_CAP100_data_set.txt	16616,39	16808,4	16904,4	16725,6	16783,8	16783,8
S_N20_CAP150_data_set.txt	16275,6	16589,4	16512,6	16495,8	16500,6	16388,4
S_N20_CAP200_data_set.txt	16042,19	16483,8	16474,8	16422	16402,2	16144,2
C_N20_CAP11_data_set.txt	801013,93	821908,37	698856,43	707034,87	677635,27	738412,36
C_N20_CAP20_data_set.txt	399590,69	398126,44	399343,2	399423,16	399394,17	399138,01
C_N20_CAP30_data_set.txt	248789,16	247579,87	248641,82	248481,41	248562,19	248483,73
C_N20_CAP50_data_set.txt	97802,59	97879,07	97767,73	97854,16	97867,18	97937,01

## Appendice A – Risultati sperimentali

C_N20_CAP80_data_set.txt	47204,93	46659,3	47090,46	47180,97	47203,1	47487,34
C_N20_CAP100_data_set.txt	16835,99	17031,6	16822,8	16846,2	16874,4	17029,8
C_N20_CAP150_data_set.txt	16466,4	16714,2	16529,4	16665	16572,6	16583,4
C_N20_CAP200_data_set.txt	16271,99	16588,8	16450,8	16514,4	16350	16348,8
MC_N20_CAP11_data_set.txt	801329,04	796612,7	771335,64	769586,76	770007,68	779958,07
MC_N20_CAP20_data_set.txt	399447,01	396871,43	399294,83	399326,26	399374,82	399157,74
MC_N20_CAP30_data_set.txt	248762,15	247652,08	248492	248568,9	248567,29	248473,38
MC_N20_CAP50_data_set.txt	97504,99	96118,24	97670,65	97670,22	97718,14	97657,37
MC_N20_CAP80_data_set.txt	46954,75	47302,47	46982,4	47043,71	47103,13	47023,31
MC_N20_CAP100_data_set.txt	16669,2	16851,6	16887,6	16926	16753,2	16936,8
MC_N20_CAP150_data_set.txt	16338,59	16668,6	16446,6	16592,4	16497,6	16440
MC_N20_CAP200_data_set.txt	16080,59	16428	16437	16453,8	16375,8	16228,8
valore medio	<b>204954,70</b>	<b>204207,77</b>	<b>199523,46</b>	<b>199817,48</b>	<b>198593,02</b>	<b>201473,10</b>

Per quanto riguarda i tempi di elaborazione sono stati rilevati i seguenti valori:

dataset	CPLEX	ILS-Random	Rollout FlussoMedio	Rollout Domanda	Rollout Random	Cluster
S_N20_CAP11_data_set.txt	7151765,625	498	1495	109	1681	6974
S_N20_CAP20_data_set.txt	7111234,375	441	2569	1359	1337	5250
S_N20_CAP30_data_set.txt	7176593,75	317	2050	1966	1698	4427
S_N20_CAP50_data_set.txt	7179156,25	217	1455	702	963	3001
S_N20_CAP80_data_set.txt	7136375	183	1343	1121	402	2360
S_N20_CAP100_data_set.txt	5675562,5	192	1250	973	1062	1783
S_N20_CAP150_data_set.txt	829421,875	171	1051	890	1259	2233
S_N20_CAP200_data_set.txt	313828,125	236	999	1084	1454	1637
C_N20_CAP11_data_set.txt	7035040	169	1455	1278	1247	4591
C_N20_CAP20_data_set.txt	7175340	246	476	684	1308	3832
C_N20_CAP30_data_set.txt	7175670	216	1607	1498	1325	3269
C_N20_CAP50_data_set.txt	7177060	160	1345	669	1194	2659
C_N20_CAP80_data_set.txt	7176760	146	1186	1070	1074	2058
C_N20_CAP100_data_set.txt	46000	139	1027	923	911	1852
C_N20_CAP150_data_set.txt	48340	186	1188	329	744	1852
C_N20_CAP200_data_set.txt	45590	142	656	1015	1171	2054
MC_N20_CAP11_data_set.txt	7151765,625	273	1104	1110	364	5913
MC_N20_CAP20_data_set.txt	7111234,375	360	157	78	706	4552
MC_N20_CAP30_data_set.txt	7176593,75	248	747	1447	1547	3502
MC_N20_CAP50_data_set.txt	7179156,25	197	818	892	1160	2642
MC_N20_CAP80_data_set.txt	7136375	152	1146	1003	737	2237
MC_N20_CAP100_data_set.txt	5675562,5	139	144	1068	1044	2094
MC_N20_CAP150_data_set.txt	829421,875	163	1250	690	402	1830

## Appendice A – Risultati sperimentali

MC_N20_CAP200_data_set.txt	313828,125	156	1094	1094	845	1912
valore medio	<b>5042819,79</b>	<b>222,79</b>	<b>1150,50</b>	<b>960,50</b>	<b>1068,13</b>	<b>3104,75</b>

### 8.3 *Variazione del parametro w*

Sono stati ottenuti i seguenti valori della funzione obiettivo:

dataset	CPLEX	ILS-Random	Rollout FlussoMedio	Rollout Domanda	Rollout Random	Cluster
S_N5_W6_data_set.txt	3856,19	3856,2	3930,6	3856,2	3934,2	3856,2
S_N5_W8_data_set.txt	3856,19	3934,2	3930,6	3856,2	3934,2	3856,2
S_N5_W10_data_set.txt	3856,19	3856,2	3930,6	3856,2	3930,6	3856,2
S_N5_W12_data_set.txt	3856,19	3934,2	3930,6	3856,2	3934,2	3856,2
S_N5_W14_data_set.txt	3856,19	3856,2	3930,6	3856,2	3934,2	3856,2
S_N5_W16_data_set.txt	3856,19	3856,2	3930,6	3856,2	3856,2	3856,2
S_N5_W18_data_set.txt	3856,19	3934,2	3930,6	3856,2	3934,2	3856,2
C_N5_W6_data_set.txt	3881,39	3881,4	3881,4	3881,4	3885,6	3888,6
C_N5_W8_data_set.txt	3881,39	3885,6	3881,4	3881,4	3944,4	3888,6
C_N5_W10_data_set.txt	3881,39	3881,4	3881,4	3881,4	3881,4	3888,6
C_N5_W12_data_set.txt	3881,39	3881,4	3881,4	3881,4	3944,4	3888,6
C_N5_W14_data_set.txt	3881,39	3885,6	3881,4	3881,4	3881,4	3888,6
C_N5_W16_data_set.txt	3881,39	3881,4	3881,4	3881,4	3944,4	3888,6
C_N5_W18_data_set.txt	3881,39	3881,4	3881,4	3881,4	3944,4	3888,6
MC_N5_W6_data_set.txt	3869,39	3881,4	3871,2	3881,4	3871,2	4084,8
MC_N5_W8_data_set.txt	3869,39	3876	3871,2	3881,4	3871,2	4084,8
MC_N5_W10_data_set.txt	3869,39	3881,4	3871,2	3881,4	3871,2	4084,8
MC_N5_W12_data_set.txt	3869,39	3876	3871,2	3881,4	3881,4	4084,8
MC_N5_W14_data_set.txt	3869,39	3871,2	3871,2	3881,4	3871,2	4084,8
MC_N5_W16_data_set.txt	3869,39	3881,4	3871,2	3881,4	3871,2	4084,8
MC_N5_W18_data_set.txt	3869,39	3871,2	3871,2	3881,4	3871,2	4084,8
S_N10_W6_data_set.txt	7745,39	7884,6	7884,6	7902,6	7884,6	7810,8
S_N10_W8_data_set.txt	7745,39	7884,6	7884,6	7902,6	7824,6	7810,8
S_N10_W10_data_set.txt	7745,39	7884,6	7884,6	7902,6	7824,6	7810,8
S_N10_W12_data_set.txt	7745,39	7887,6	7884,6	7902,6	7884,6	7810,8
S_N10_W14_data_set.txt	7745,39	7884,6	7884,6	7902,6	7884,6	7810,8
S_N10_W16_data_set.txt	7745,39	7884,6	7884,6	7902,6	7884,6	7810,8
S_N10_W18_data_set.txt	7745,39	7902,6	7884,6	7902,6	7884,6	7810,8
C_N10_W6_data_set.txt	7944,59	7983	7914	7922,4	7975,8	7916,4
C_N10_W8_data_set.txt	7944,59	8009,4	7914	7922,4	7954,2	7916,4
C_N10_W10_data_set.txt	7944,59	8020,8	7914	7922,4	7968	7916,4
C_N10_W12_data_set.txt	7944,59	8011,8	7914	7922,4	7965,6	7916,4
C_N10_W14_data_set.txt	7944,59	7973,4	7914	7922,4	7981,8	7916,4

## Appendice A – Risultati sperimentali

C_N10_W16_data_set.txt	7944,59	8008,2	7914	7922,4	7920,6	7916,4
C_N10_W18_data_set.txt	7944,59	7965,6	7914	7922,4	7969,2	7916,4
MC_N10_W6_data_set.txt	7785	7947,6	7932,6	7911,6	7911,6	7942,2
MC_N10_W8_data_set.txt	7785	7978,2	7932,6	7911,6	7869,6	7942,2
MC_N10_W10_data_set.txt	7785	7974,6	7932,6	7911,6	7913,4	7942,2
MC_N10_W12_data_set.txt	7785	7975,8	7932,6	7911,6	7913,4	7942,2
MC_N10_W14_data_set.txt	7785	7962,6	7932,6	7911,6	7946,4	7942,2
MC_N10_W16_data_set.txt	7785	7924,8	7932,6	7911,6	7915,8	7942,2
MC_N10_W18_data_set.txt	7785	7920,6	7932,6	7911,6	7911,6	7942,2
S_N20_W6_data_set.txt	16616,39	16827	16904,4	16725,6	16720,8	16783,8
S_N20_W8_data_set.txt	16616,39	16876,2	16904,4	16725,6	16812,6	16783,8
S_N20_W10_data_set.txt	16616,39	16798,8	16904,4	16725,6	16810,8	16783,8
S_N20_W12_data_set.txt	16616,39	16827	16904,4	16725,6	16793,4	16783,8
S_N20_W14_data_set.txt	16616,39	16807,2	16904,4	16725,6	16732,2	16783,8
S_N20_W16_data_set.txt	16616,39	16894,2	16904,4	16725,6	16723,2	16783,8
S_N20_W18_data_set.txt	16616,39	16887,6	16904,4	16725,6	16797,6	16783,8
C_N20_W6_data_set.txt	16836	16999,2	16822,8	16846,2	16795,2	17029,8
C_N20_W8_data_set.txt	16836	16929,6	16822,8	16846,2	16834,8	17029,8
C_N20_W10_data_set.txt	16869,6	16969,8	16822,8	16846,2	16893	17029,8
C_N20_W12_data_set.txt	16843,8	17017,8	16822,8	16846,2	16948,8	17029,8
C_N20_W14_data_set.txt	16836	17016	16822,8	16846,2	16878	17029,8
C_N20_W16_data_set.txt	16836	17008,8	16822,8	16846,2	16851	17029,8
C_N20_W18_data_set.txt	16837,19	17086,2	16822,8	16846,2	16889,4	17029,8
MC_N20_W6_data_set.txt	16669,2	16879,2	16887,6	16926	16765,8	16936,8
MC_N20_W8_data_set.txt	16669,2	16891,8	16887,6	16926	16726,8	16936,8
MC_N20_W10_data_set.txt	16669,2	16952,4	16887,6	16926	16716	16936,8
MC_N20_W12_data_set.txt	16669,2	16832,4	16887,6	16926	16812	16936,8
MC_N20_W14_data_set.txt	16669,2	16967,4	16887,6	16926	16752	16936,8
MC_N20_W16_data_set.txt	16669,2	16927,2	16887,6	16926	16740,6	16936,8
MC_N20_W18_data_set.txt	16669,2	16890	16887,6	16926	16752,6	16936,8
S_N50_W6_data_set.txt	45881,4	46826,4	46264,8	46696,8	46275	47068,2
S_N50_W8_data_set.txt	45889,79	46920,6	46264,8	46696,8	46290,6	47068,2
S_N50_W10_data_set.txt	45898,79	46659	46264,8	46696,8	46406,4	47068,2
S_N50_W12_data_set.txt	45893,4	46837,2	46264,8	46696,8	46576,8	47068,2
S_N50_W14_data_set.txt	45867,6	46887	46264,8	46696,8	46405,8	47068,2
S_N50_W16_data_set.txt	45884,4	46803,6	46264,8	46696,8	46491,6	47068,2
S_N50_W18_data_set.txt	45913,2	47085	46264,8	46696,8	46617,6	47068,2
C_N50_W6_data_set.txt	46468,79	47003,4	46877,4	47284,2	46792,8	47583,6
C_N50_W8_data_set.txt	46647	47374,2	46877,4	47284,2	46476	47583,6
C_N50_W10_data_set.txt	46396,2	47158,8	46877,4	47284,2	46647,6	47583,6
C_N50_W12_data_set.txt	46869,6	47080,2	46877,4	47284,2	46650	47583,6
C_N50_W14_data_set.txt	47310	47145	46877,4	47284,2	46615,2	47583,6
C_N50_W16_data_set.txt	46736,4	47163,6	46877,4	47284,2	46660,2	47583,6
C_N50_W18_data_set.txt	46848	47182,8	46877,4	47284,2	46667,4	47583,6
MC_N50_W6_data_set.txt	45994,2	46891,2	46485,6	47056,2	46491,6	47422,8

Appendice A – Risultati sperimentali

MC_N50_W8_data_set.txt	46162,2	46750,8	46485,6	47056,2	46534,2	47422,8
MC_N50_W10_data_set.txt	46140	46840,2	46485,6	47056,2	46497,6	47422,8
MC_N50_W12_data_set.txt	46128,6	46885,2	46485,6	47056,2	46455	47422,8
MC_N50_W14_data_set.txt	45953,4	46909,2	46485,6	47056,2	46576,8	47422,8
MC_N50_W16_data_set.txt	46105,8	46888,2	46485,6	47056,2	46404	47422,8
MC_N50_W18_data_set.txt	46111,79	46926	46485,6	47056,2	46425	47422,8
S_N80_W6_data_set.txt	78247,8	79498,2	78671,4	80252,4	78961,8	80569,8
S_N80_W8_data_set.txt	57037,2	57795	57212,4	57839,4	57597,6	58721,4
S_N80_W10_data_set.txt	57109,8	58015,8	57212,4	57839,4	57521,4	58721,4
S_N80_W12_data_set.txt	56866,2	57551,4	57212,4	57839,4	57282,6	58721,4
S_N80_W14_data_set.txt	67509,17	57806,4	57212,4	57839,4	57440,4	58721,4
S_N80_W16_data_set.txt	57306,6	57885	57212,4	57839,4	57291	58721,4
S_N80_W18_data_set.txt	67628,93	57804,6	57212,4	57839,4	57484,2	58721,4
C_N80_W6_data_set.txt	79641,6	80301,6	79045,2	82023,6	79761,6	81041,4
C_N80_W8_data_set.txt	59308,2	57843,6	57457,2	57912,6	57808,2	58620
C_N80_W10_data_set.txt	58497	58126,2	57457,2	57912,6	57523,8	58620
C_N80_W12_data_set.txt	60833,4	58167,6	57457,2	57912,6	57611,4	58620
C_N80_W14_data_set.txt	89306,63	58108,2	57457,2	57912,6	57691,2	58620
C_N80_W16_data_set.txt	0	57931,8	57457,2	57912,6	57772,8	58620
C_N80_W18_data_set.txt	0	57945,6	57457,2	57912,6	57483,6	58620
MC_N80_W6_data_set.txt	78642	79787,4	78902,4	80395,8	79482	79866
MC_N80_W8_data_set.txt	57010,2	57862,8	57220,2	58128,6	57589,8	58919,4
MC_N80_W10_data_set.txt	59332,8	57990,6	57220,2	58128,6	57600,6	58919,4
MC_N80_W12_data_set.txt	57271,79	58071	57220,2	58128,6	57412,8	58919,4
MC_N80_W14_data_set.txt	67305,96	57703,8	57220,2	58128,6	57583,2	58919,4
MC_N80_W16_data_set.txt	78984,8	57865,2	57220,2	58128,6	57774	58919,4
MC_N80_W18_data_set.txt	69362,95	57519	57220,2	58128,6	57608,4	58919,4
S_N100_W6_data_set.txt	100862,4	102306	101645,4	102954	100777,2	102964,2
S_N100_W8_data_set.txt	72892,8	72843	72975	73104	72613,2	74235
S_N100_W10_data_set.txt	82999,53	72546,6	72975	73104	72721,8	74235
S_N100_W12_data_set.txt	83774,8	72981,6	72975	73104	72391,8	74235
S_N100_W14_data_set.txt	83981,2	73251	72975	73104	72841,8	74235
S_N100_W16_data_set.txt	143273,63	73318,2	72975	73104	72414,6	74235
S_N100_W18_data_set.txt	191445,59	72796,2	72975	73104	72715,8	74235
C_N100_W6_data_set.txt	102031,8	102073,2	101746,2	103117,8	101029,8	104005,8
C_N100_W8_data_set.txt	73638,59	73621,2	72751,2	74425,8	72985,8	74748
C_N100_W10_data_set.txt	204441,14	73046,4	72751,2	74425,8	72935,4	74748
C_N100_W12_data_set.txt	0	73589,4	72751,2	74425,8	72982,2	74748
C_N100_W14_data_set.txt	0	73099,2	72751,2	74425,8	72605,4	74748
C_N100_W16_data_set.txt	0	73122	72751,2	74425,8	72552,6	74748
C_N100_W18_data_set.txt	0	73157,4	72751,2	74425,8	72729,6	74748
MC_N100_W6_data_set.txt	120662,59	101731,2	101339,4	103197	101269,8	104661,6
MC_N100_W8_data_set.txt	74323,79	73383	72833,4	73250,4	72679,8	74763,6
MC_N100_W10_data_set.txt	73397,4	73057,8	72833,4	73250,4	72952,2	74763,6
MC_N100_W12_data_set.txt	223123,01	73020,6	72833,4	73250,4	72681	74763,6

## Appendice A – Risultati sperimentali

MC_N100_W14_data_set.txt	0	73019,4	72833,4	73250,4	72766,2	74763,6
MC_N100_W16_data_set.txt	0	72765,6	72833,4	73250,4	72907,2	74763,6
MC_N100_W18_data_set.txt	0	73093,2	72833,4	73250,4	72629,4	74763,6
	<b>35362,05</b>	<b>35661,27</b>	<b>35425,75</b>	<b>35778,88</b>	<b>35431,35</b>	<b>36122,39</b>

Per quanto riguarda i tempi di elaborazione sono stati rilevati i seguenti valori:

dataset	CPLEX	ILS- Random	Rollout FlussoMedio	Rollout Domanda	Rollout Random	Cluster
S_N5_W6_data_set.txt	875	38	10	7	1	592
S_N5_W8_data_set.txt	1125	58	18	11	28	846
S_N5_W10_data_set.txt	1515,625	63	18	9	17	983
S_N5_W12_data_set.txt	1828,125	68	27	10	33	1534
S_N5_W14_data_set.txt	2375	97	28	20	46	2605
S_N5_W16_data_set.txt	2890,625	93	30	18	9	2540
S_N5_W18_data_set.txt	3421,825	108	30	13	38	3442
C_N5_W6_data_set.txt	2062,5	40	11	1	1	465
C_N5_W8_data_set.txt	3765,625	46	15	3	22	893
C_N5_W10_data_set.txt	5140,625	62	27	3	24	1604
C_N5_W12_data_set.txt	6593,75	80	19	3	2	1641
C_N5_W14_data_set.txt	9609,375	78	27	5	10	2207
C_N5_W16_data_set.txt	6562,5	108	30	3	35	3006
C_N5_W18_data_set.txt	18718,75	104	36	5	24	4244
MC_N5_W6_data_set.txt	1234,375	36	12	1	3	248
MC_N5_W8_data_set.txt	1984,375	62	14	2	11	406
MC_N5_W10_data_set.txt	2640,625	66	18	2	1	618
MC_N5_W12_data_set.txt	4562,5	75	22	4	13	861
MC_N5_W14_data_set.txt	4000	88	27	4	34	1397
MC_N5_W16_data_set.txt	4750	121	19	3	45	1690
MC_N5_W18_data_set.txt	5640,625	107	33	3	14	2353
S_N10_W6_data_set.txt	2750	83	5	70	176	999
S_N10_W8_data_set.txt	5359,375	97	4	94	67	1389
S_N10_W10_data_set.txt	7968,75	128	9	103	55	1865
S_N10_W12_data_set.txt	9312,5	147	6	118	335	3106
S_N10_W14_data_set.txt	10453,125	174	7	146	36	4240
S_N10_W16_data_set.txt	16015,625	197	12	227	156	5058
S_N10_W18_data_set.txt	16796,875	231	18	185	14	6580
C_N10_W6_data_set.txt	47312,5	58	100	108	145	675
C_N10_W8_data_set.txt	62406,25	91	151	172	41	1316
C_N10_W10_data_set.txt	91250	109	180	198	98	2275
C_N10_W12_data_set.txt	95687,5	122	220	253	21	3195
C_N10_W14_data_set.txt	181750	159	242	296	301	3922
C_N10_W16_data_set.txt	170375	176	277	388	410	6942
C_N10_W18_data_set.txt	212671,875	215	318	391	212	6532

Appendice A – Risultati sperimentali

MC_N10_W6_data_set.txt	8468,75	74	129	46	163	923
MC_N10_W8_data_set.txt	16515,625	86	166	55	121	1835
MC_N10_W10_data_set.txt	22562,5	119	200	59	253	2309
MC_N10_W12_data_set.txt	27984,375	131	230	78	75	3287
MC_N10_W14_data_set.txt	27812,5	155	285	99	7	4737
MC_N10_W16_data_set.txt	34015,625	222	318	90	142	5987
MC_N10_W18_data_set.txt	44765,625	228	364	113	89	7830
S_N20_W6_data_set.txt	46000	143	1212	943	290	2258
S_N20_W8_data_set.txt	97062,5	193	1588	1303	1315	3676
S_N20_W10_data_set.txt	90265,625	236	1947	1502	1942	5300
S_N20_W12_data_set.txt	108468,75	277	2281	1798	1301	7439
S_N20_W14_data_set.txt	172750	318	2697	2198	2508	10254
S_N20_W16_data_set.txt	150234,375	371	3108	2454	1320	13020
S_N20_W18_data_set.txt	372031,25	412	3415	2645	1541	15926
C_N20_W6_data_set.txt	5675562,5	124	1074	1000	642	1802
C_N20_W8_data_set.txt	7161640,625	169	1428	1183	1387	3325
C_N20_W10_data_set.txt	7180281,25	210	1730	1513	1701	5197
C_N20_W12_data_set.txt	7145671,875	297	2135	1842	2161	7175
C_N20_W14_data_set.txt	7181359,375	328	2518	2120	1496	9880
C_N20_W16_data_set.txt	7166406,25	354	2854	2544	2813	12917
C_N20_W18_data_set.txt	7152218,75	386	3240	2948	3092	16271
MC_N20_W6_data_set.txt	105234,375	129	147	1075	801	2199
MC_N20_W8_data_set.txt	222796,875	183	188	1356	1481	2980
MC_N20_W10_data_set.txt	301281,25	239	249	1789	1593	5053
MC_N20_W12_data_set.txt	311656,25	262	284	2093	2202	7393
MC_N20_W14_data_set.txt	306515,625	304	351	2498	2281	9702
MC_N20_W16_data_set.txt	537109,375	365	372	2759	3002	12601
MC_N20_W18_data_set.txt	475796,875	392	419	3159	3283	16320
S_N50_W6_data_set.txt	7179437,5	352	15373	14848	16589	6707
S_N50_W8_data_set.txt	6818843,75	456	20444	19944	18053	11866
S_N50_W10_data_set.txt	7184781,25	596	26071	24577	27518	18096
S_N50_W12_data_set.txt	7184000	691	31521	30224	28440	25694
S_N50_W14_data_set.txt	7184234,375	789	36192	34389	26029	34562
S_N50_W16_data_set.txt	7184921,875	886	41786	39413	41928	44532
S_N50_W18_data_set.txt	7187937,5	1023	46710	44074	49413	58489
C_N50_W6_data_set.txt	7188484,375	296	8580	12949	10053	6527
C_N50_W8_data_set.txt	7182203,125	398	11945	18339	18926	11432
C_N50_W10_data_set.txt	7183062,5	536	15409	22935	23217	18869
C_N50_W12_data_set.txt	7181750	622	19012	27591	28512	26672
C_N50_W14_data_set.txt	7184281,25	717	22619	32505	20901	36978
C_N50_W16_data_set.txt	7185718,75	819	26035	37860	39644	45797
C_N50_W18_data_set.txt	7187859,375	965	30041	43139	46180	57759
MC_N50_W6_data_set.txt	7175609,375	349	12527	5994	13828	6498
MC_N50_W8_data_set.txt	7181375	431	17007	8079	19816	11347
MC_N50_W10_data_set.txt	7182265,625	533	21959	10223	23793	17904

## Appendice A – Risultati sperimentali

MC_N50_W12_data_set.txt	7173515,625	666	27116	12393	30585	28117
MC_N50_W14_data_set.txt	7184453,125	795	31505	14649	7101	38009
MC_N50_W16_data_set.txt	7178656,25	890	35635	16731	29177	45138
MC_N50_W18_data_set.txt	7187265,625	1003	40819	19064	43218	62442
S_N80_W6_data_set.txt	7162093,75	494	58004	57904	19292	17490
S_N80_W8_data_set.txt	7008984,375	676	79597	73666	74501	25960
S_N80_W10_data_set.txt	7084953,125	879	97067	95843	89716	41617
S_N80_W12_data_set.txt	7143109,375	1054	118345	114808	65887	61024
S_N80_W14_data_set.txt	7153625	1281	137729	134667	93430	91986
S_N80_W16_data_set.txt	7149625	1350	160778	152862	161741	110417
S_N80_W18_data_set.txt	7197109,375	1533	180037	175934	182271	134390
C_N80_W6_data_set.txt	7172343,75	413	34393	31448	31029	18016
C_N80_W8_data_set.txt	6766968,75	596	75391	72595	66474	31847
C_N80_W10_data_set.txt	7092968,75	767	99239	94532	94831	58333
C_N80_W12_data_set.txt	7192687,5	970	122602	116233	95795	68165
C_N80_W14_data_set.txt	7182734,375	1164	144588	137780	125647	69279
C_N80_W16_data_set.txt	7183656,25	1349	167432	157793	157205	88579
C_N80_W18_data_set.txt	7169562,5	1501	192353	184226	161442	115054
MC_N80_W6_data_set.txt	7183984,375	468	57496	53918	47692	15903
MC_N80_W8_data_set.txt	7181453,125	644	80253	68879	68063	32609
MC_N80_W10_data_set.txt	7154546,875	826	102747	89083	89841	51627
MC_N80_W12_data_set.txt	7165015,625	1054	124760	107497	107683	88490
MC_N80_W14_data_set.txt	7057593,75	1162	147476	128503	113002	77015
MC_N80_W16_data_set.txt	7131875	1315	171542	147309	104115	111391
MC_N80_W18_data_set.txt	7192000	1514	195178	168198	149055	150697
S_N100_W6_data_set.txt	7159312,5	612	111581	104130	87756	20220
S_N100_W8_data_set.txt	7129593,75	821	150042	76916	131708	37999
S_N100_W10_data_set.txt	7169256,25	1030	194951	99676	177290	59474
S_N100_W12_data_set.txt	7156593,75	1303	236161	121572	231512	88195
S_N100_W14_data_set.txt	7164515,625	1519	290244	145603	228148	117382
S_N100_W16_data_set.txt	7172656,25	1702	345080	165774	263894	148518
S_N100_W18_data_set.txt	6961750	1970	364359	187908	335721	188278
C_N100_W6_data_set.txt	7157375	541	103326	93154	41330	17939
C_N100_W8_data_set.txt	6988531,25	743	146055	122957	124483	36768
C_N100_W10_data_set.txt	7158140,625	1018	195864	164855	140513	60646
C_N100_W12_data_set.txt	7156187,5	1232	238599	203078	161535	85787
C_N100_W14_data_set.txt	7129200	1428	285973	244975	216231	116482
C_N100_W16_data_set.txt	7189593,75	1715	330856	291967	303262	151201
C_N100_W18_data_set.txt	7158828,125	1897	381921	324284	178019	184633
MC_N100_W6_data_set.txt	7177546,875	573	102386	99225	95049	19214
MC_N100_W8_data_set.txt	7174015,625	852	147899	117862	137033	37663
MC_N100_W10_data_set.txt	7190421,875	1044	197045	152489	165043	56453
MC_N100_W12_data_set.txt	7141890	1265	239494	188363	224548	82379
MC_N100_W14_data_set.txt	7091562,5	1446	287882	227381	251038	118795
MC_N100_W16_data_set.txt	7193484,375	1697	330206	264352	274875	153838

MC_N100_W18_data_set.txt	7006671,875	1922	377832	294996	352937	204795
	<b>3993639,28</b>	<b>567,02</b>	<b>64920,78</b>	<b>52295,68</b>	<b>54166,97</b>	<b>34443,24</b>

## 8.4 *Variazione del parametro h*

Sono stati ottenuti i seguenti valori della funzione obiettivo:

dataset	CPLEX	ILS- Random	Rollout FlussoMedio	Rollout Domanda	Rollout Random	Cluster
S_N5_H4_data_set.txt	3856,19	3934,2	3930,6	3856,2	3934,2	3856,2
S_N5_H5_data_set.txt	2618,39	2662,8	2665,2	2618,4	2662,8	2745
S_N5_H6_data_set.txt	2618,39	2618,4	2665,2	2618,4	2662,8	2745
S_N5_H7_data_set.txt	2618,39	2618,4	2665,2	2618,4	2662,8	2745
S_N5_H8_data_set.txt	2618,39	2662,8	2665,2	2618,4	2662,8	2745
S_N5_H9_data_set.txt	2618,39	2618,4	2665,2	2618,4	2662,8	2745
S_N5_H10_data_set.txt	2618,39	2618,4	2665,2	2618,4	2662,8	2745
C_N5_H4_data_set.txt	3881,39	3885,6	3881,4	3881,4	3944,4	3888,6
C_N5_H5_data_set.txt	2660,39	2680,8	2660,4	2660,4	2660,4	2675,4
C_N5_H6_data_set.txt	2660,39	2660,4	2660,4	2660,4	2680,8	2675,4
C_N5_H7_data_set.txt	2660,39	2660,4	2660,4	2660,4	2660,4	2675,4
C_N5_H8_data_set.txt	2660,39	2660,4	2660,4	2660,4	2680,8	2675,4
C_N5_H9_data_set.txt	2660,39	2660,4	2660,4	2660,4	2680,8	2675,4
C_N5_H10_data_set.txt	2660,39	2660,4	2660,4	2660,4	2680,8	2675,4
MC_N5_H4_data_set.txt	3869,39	3871,2	3871,2	3881,4	3871,2	4084,8
MC_N5_H5_data_set.txt	2641,79	2660,4	2644,2	2660,4	2655	2834,4
MC_N5_H6_data_set.txt	2641,79	2644,2	2644,2	2660,4	2644,2	2834,4
MC_N5_H7_data_set.txt	2641,79	2644,2	2644,2	2660,4	2660,4	2834,4
MC_N5_H8_data_set.txt	2641,79	2653,2	2644,2	2660,4	2655	2834,4
MC_N5_H9_data_set.txt	2641,79	2644,2	2644,2	2660,4	2662,2	2834,4
MC_N5_H10_data_set.txt	2641,79	2653,2	2644,2	2660,4	2680,8	2834,4
S_N10_H4_data_set.txt	7745,39	7902,6	7884,6	7902,6	7884,6	7810,8
S_N10_H5_data_set.txt	5200,2	5291,4	5259,6	5259,6	5259,6	5222,4
S_N10_H6_data_set.txt	5200,2	5259,6	5259,6	5259,6	5259,6	5222,4
S_N10_H7_data_set.txt	5200,2	5259,6	5259,6	5259,6	5259,6	5222,4
S_N10_H8_data_set.txt	5200,2	5259,6	5259,6	5259,6	5259,6	5222,4
S_N10_H9_data_set.txt	5200,2	5292	5259,6	5259,6	5259,6	5222,4
S_N10_H10_data_set.txt	5200,2	5259,6	5259,6	5259,6	5259,6	5222,4
C_N10_H4_data_set.txt	7944,59	7972,8	7914	7922,4	8037	7916,4
C_N10_H5_data_set.txt	5362,19	5368,2	5322,6	5338,2	5316,6	5315,4
C_N10_H6_data_set.txt	5362,19	5322,6	5322,6	5338,2	5380,2	5315,4
C_N10_H7_data_set.txt	5362,19	5307	5322,6	5338,2	5325	5315,4
C_N10_H8_data_set.txt	5362,19	5386,8	5322,6	5338,2	5316,6	5315,4
C_N10_H9_data_set.txt	5362,19	5334,6	5322,6	5338,2	5323,8	5315,4

## Appendice A – Risultati sperimentali

C_N10_H10_data_set.txt	5362,19	5400	5322,6	5338,2	5338,2	5315,4
MC_N10_H4_data_set.txt	7785	7944,6	7932,6	7911,6	7902	7942,2
MC_N10_H5_data_set.txt	5254,2	5336,4	5314,8	5360,4	5311,8	5287,2
MC_N10_H6_data_set.txt	5254,2	5325,6	5314,8	5360,4	5308,2	5287,2
MC_N10_H7_data_set.txt	5254,2	5329,2	5314,8	5360,4	5311,8	5287,2
MC_N10_H8_data_set.txt	5254,2	5292,6	5314,8	5360,4	5298	5287,2
MC_N10_H9_data_set.txt	5254,2	5323,8	5314,8	5360,4	5317,2	5287,2
MC_N10_H10_data_set.txt	5254,2	5330,4	5314,8	5360,4	5305,8	5287,2
S_N20_H4_data_set.txt	16616,39	16852,8	16904,4	16725,6	16794	16783,8
S_N20_H5_data_set.txt	16616,39	16839,6	16904,4	16725,6	16723,2	16783,8
S_N20_H6_data_set.txt	16616,39	16860,6	16904,4	16725,6	16771,8	16783,8
S_N20_H7_data_set.txt	16616,39	16792,2	16904,4	16725,6	16702,8	16783,8
S_N20_H8_data_set.txt	16616,39	16861,2	16904,4	16725,6	16765,8	16783,8
S_N20_H9_data_set.txt	16616,39	16866,6	16904,4	16725,6	16759,2	16783,8
S_N20_H10_data_set.txt	16616,39	16939,8	16904,4	16725,6	16695,6	16783,8
C_N20_H4_data_set.txt	16836	16941,6	16822,8	16846,2	16939,8	17029,8
C_N20_H5_data_set.txt	16836	17125,2	16822,8	16846,2	17008,8	17029,8
C_N20_H6_data_set.txt	16848	16986	16822,8	16846,2	16821	17029,8
C_N20_H7_data_set.txt	16836	16929,6	16822,8	16846,2	16846,8	17029,8
C_N20_H8_data_set.txt	16845,6	17071,8	16822,8	16846,2	16880,4	17029,8
C_N20_H9_data_set.txt	16836	16990,2	16822,8	16846,2	16864,8	17029,8
C_N20_H10_data_set.txt	16843,79	17097,6	16822,8	16846,2	16895,4	17029,8
MC_N20_H4_data_set.txt	16669,2	17036,4	16887,6	16926	16828,8	16936,8
MC_N20_H5_data_set.txt	16669,2	16893,6	16887,6	16926	16757,4	16936,8
MC_N20_H6_data_set.txt	16669,2	16855,8	16887,6	16926	16917,6	16936,8
MC_N20_H7_data_set.txt	16669,2	17031	16887,6	16926	16777,8	16936,8
MC_N20_H8_data_set.txt	16669,2	16858,2	16887,6	16926	16788	16936,8
MC_N20_H9_data_set.txt	16669,2	16975,2	16887,6	16926	16780,8	16936,8
MC_N20_H10_data_set.txt	16669,2	16880,4	16887,6	16926	16741,2	16936,8
S_N50_H4_data_set.txt	45881,4	46812,6	46264,8	46696,8	46290,6	47068,2
S_N50_H5_data_set.txt	33509,4	33981,6	33870	34213,8	33727,2	34475,4
S_N50_H6_data_set.txt	33509,4	34122	33870	34213,8	33907,2	34475,4
S_N50_H7_data_set.txt	33509,4	33960,6	33870	34213,8	33833,4	34475,4
S_N50_H8_data_set.txt	33531,6	34405,2	33870	34213,8	33753,6	34475,4
S_N50_H9_data_set.txt	33520,79	34134	33870	34213,8	33821,4	34475,4
S_N50_H10_data_set.txt	33523,2	33873	33870	34213,8	33914,4	34475,4
C_N50_H4_data_set.txt	46468,79	47031,6	46877,4	47284,2	46741,2	47583,6
C_N50_H5_data_set.txt	34096,79	34306,8	34003,2	34359	34059,6	34581,6
C_N50_H6_data_set.txt	34238,4	34353	34003,2	34359	34131,6	34581,6
C_N50_H7_data_set.txt	34203	34342,8	34003,2	34359	34126,2	34581,6
C_N50_H8_data_set.txt	34286,4	34267,2	34003,2	34359	34219,8	34581,6
C_N50_H9_data_set.txt	34237,2	34489,8	34003,2	34359	34029	34581,6
C_N50_H10_data_set.txt	34857	34410,6	34003,2	34359	34222,8	34581,6
MC_N50_H4_data_set.txt	45994,2	47058	46485,6	47056,2	46338,6	47422,8
MC_N50_H5_data_set.txt	33717	34026	33973,8	34404,6	33772,8	34402,8

Appendice A – Risultati sperimentali

MC_N50_H6_data_set.txt	33654	34079,4	33973,8	34404,6	33979,8	34402,8
MC_N50_H7_data_set.txt	33705,6	34347,6	33973,8	34404,6	33927,6	34402,8
MC_N50_H8_data_set.txt	34342,2	34159,2	33973,8	34404,6	33913,2	34402,8
MC_N50_H9_data_set.txt	33865,2	34316,4	33973,8	34404,6	33949,8	34402,8
MC_N50_H10_data_set.txt	34426,2	34401	33973,8	34404,6	34090,8	34402,8
S_N80_H4_data_set.txt	78247,8	79484,4	78671,4	80252,4	79092	80569,8
S_N80_H5_data_set.txt	57311,4	57840,6	57379,2	58361,4	57569,4	58162,8
S_N80_H6_data_set.txt	58408,2	57744,6	57226,8	57741	57262,8	58041,6
S_N80_H7_data_set.txt	57531	57886,8	57226,8	57741	57444,6	58041,6
S_N80_H8_data_set.txt	60451,19	57918,6	57226,8	57741	57288,6	58041,6
S_N80_H9_data_set.txt	57623,39	57876,6	57226,8	57741	57348,6	58041,6
S_N80_H10_data_set.txt	57969	57645	57226,8	57741	57447,6	58041,6
C_N80_H4_data_set.txt	79641,59	80206,2	79045,2	82023,6	79694,4	81041,4
C_N80_H5_data_set.txt	58677	58054,8	57624,6	59691,6	57664,2	58312,8
C_N80_H6_data_set.txt	59667,6	58158,6	57568,2	58030,8	57499,8	58544,4
C_N80_H7_data_set.txt	59894,39	57862,8	57568,2	58030,8	57777	58544,4
C_N80_H8_data_set.txt	78056,17	58145,4	57568,2	58030,8	57442,8	58544,4
C_N80_H9_data_set.txt	59347,2	58164	57568,2	58030,8	57526,2	58544,4
C_N80_H10_data_set.txt	58512,47	57840	57568,2	58030,8	57793,2	58544,4
MC_N80_H4_data_set.txt	78642	79932	78902,4	80395,8	78986,4	79866
MC_N80_H5_data_set.txt	57102,6	57663,6	57343,8	58831,2	57342,6	58524
MC_N80_H6_data_set.txt	58847,39	57794,4	57370,2	58288,2	57254,4	58524
MC_N80_H7_data_set.txt	58867,79	57789,6	57370,2	58288,2	57463,8	58524
MC_N80_H8_data_set.txt	57357	57831	57370,2	58288,2	57494,4	58524
MC_N80_H9_data_set.txt	58739,4	57834,6	57370,2	58288,2	57330,6	58524
MC_N80_H10_data_set.txt	57801	57854,4	57370,2	58288,2	57109,2	58524
S_N100_H4_data_set.txt	100862,4	101749,2	101645,4	102954	101249,4	102964,2
S_N100_H5_data_set.txt	72474	72925,8	72791,4	73162,2	72622,2	73765,8
S_N100_H6_data_set.txt	83960,83	72850,2	72787,2	74783,4	72280,2	74206,8
S_N100_H7_data_set.txt	152444,16	72817,8	72499,8	74385,6	72666,6	74361,6
S_N100_H8_data_set.txt	73798,8	72932,4	72499,8	74385,6	72383,4	74361,6
S_N100_H9_data_set.txt	133542,46	72707,4	72499,8	74385,6	72382,8	74361,6
S_N100_H10_data_set.txt	114774,4	73065,6	72499,8	74385,6	72809,4	74361,6
C_N100_H4_data_set.txt	102031,8	102600	101746,2	103117,8	101592,6	104005,8
C_N100_H5_data_set.txt	74121,59	73231,2	72718,8	72834,6	72600	74219,4
C_N100_H6_data_set.txt		73231,2	72855	74452,8	72564,6	74316
C_N100_H7_data_set.txt		73039,2	72796,2	74702,4	72402	74383,8
C_N100_H8_data_set.txt		73195,8	72796,2	74702,4	72816,6	74383,8
C_N100_H9_data_set.txt		73152	72796,2	74702,4	72536,4	74383,8
C_N100_H10_data_set.txt		73024,8	72796,2	74702,4	72522	74383,8
MC_N100_H4_data_set.txt	120662,59	102178,2	101339,4	103197	100963,8	104661,6
MC_N100_H5_data_set.txt	73317,59	73187,4	72526,2	73737	72580,8	74167,2
MC_N100_H6_data_set.txt		73015,8	72502,2	74608,2	72573,6	73959
MC_N100_H7_data_set.txt		72846,6	72658,2	73368	72598,2	73958,4
MC_N100_H8_data_set.txt		73056,6	72664,2	73156,8	72854,4	73958,4

## Appendice A – Risultati sperimentali

MC_N100_H9_data_set.txt		72854,4	72664,2	73156,8	72429,6	73958,4
MC_N100_H10_data_set.txt		73119	72664,2	73156,8	72532,2	73958,4
	<b>31794,39</b>	<b>33283,68</b>	<b>33064,82</b>	<b>33516,03</b>	<b>33052,12</b>	<b>33628,35</b>

Per quanto riguarda i tempi di elaborazione sono stati rilevati i seguenti valori:

dataset	CPLEX	ILS-Rollout	Rollout FlussoMedio	Rollout Domanda	Rollout Random	Cluster
S_N5_H4_data_set.txt	875	35	13	5	14	503
S_N5_H5_data_set.txt	1031,25	43	3	6	20	707
S_N5_H6_data_set.txt	1109,375	53	4	10	8	976
S_N5_H7_data_set.txt	1328,125	62	7	7	13	951
S_N5_H8_data_set.txt	1609,375	74	7	10	24	1067
S_N5_H9_data_set.txt	1781,25	78	5	9	4	1278
S_N5_H10_data_set.txt	2000	88	8	12	18	2477
C_N5_H4_data_set.txt	2062,5	36	11	1	12	408
C_N5_H5_data_set.txt	3093,75	47	12	1	1	610
C_N5_H6_data_set.txt	4062,5	60	20	2	5	924
C_N5_H7_data_set.txt	5937,5	60	15	2	6	2162
C_N5_H8_data_set.txt	6687,5	68	16	2	20	1726
C_N5_H9_data_set.txt	7031,25	90	31	4	27	1768
C_N5_H10_data_set.txt	10062,5	84	23	3	46	3115
MC_N5_H4_data_set.txt	1234,375	31	7	1	6	263
MC_N5_H5_data_set.txt	2328,125	44	11	2	13	398
MC_N5_H6_data_set.txt	3343,75	54	10	2	2	687
MC_N5_H7_data_set.txt	3984,375	71	14	2	2	1222
MC_N5_H8_data_set.txt	4375	72	13	2	27	1312
MC_N5_H9_data_set.txt	5296,875	77	22	4	19	1265
MC_N5_H10_data_set.txt	4290,625	99	16	3	13	1590
S_N10_H4_data_set.txt	2750	74	3	77	109	1264
S_N10_H5_data_set.txt	3812,5	102	96	205	48	1907
S_N10_H6_data_set.txt	3515,625	137	130	232	63	2237
S_N10_H7_data_set.txt	4343,75	176	148	281	55	3033
S_N10_H8_data_set.txt	5390,625	168	164	309	42	3731
S_N10_H9_data_set.txt	7500	167	179	332	83	4794
S_N10_H10_data_set.txt	8343,75	199	201	366	166	5172
C_N10_H4_data_set.txt	47312,5	69	95	111	14	728
C_N10_H5_data_set.txt	39796,875	76	75	95	93	1310
C_N10_H6_data_set.txt	50640,625	96	96	113	50	2790
C_N10_H7_data_set.txt	58781,25	111	107	128	136	2013
C_N10_H8_data_set.txt	54531,25	129	126	145	71	3771
C_N10_H9_data_set.txt	69968,75	156	140	173	204	4403
C_N10_H10_data_set.txt	84390,625	180	155	181	337	5346

## Appendice A – Risultati sperimentali

MC_N10_H4_data_set.txt	8468,75	71	115	32	35	825
MC_N10_H5_data_set.txt	13171,875	87	172	76	25	1507
MC_N10_H6_data_set.txt	24765,625	97	215	109	197	2770
MC_N10_H7_data_set.txt	24812,5	117	246	115	164	2061
MC_N10_H8_data_set.txt	23218,75	138	287	135	20	3362
MC_N10_H9_data_set.txt	26125	180	310	126	271	4485
MC_N10_H10_data_set.txt	32218,75	171	356	149	59	4839
S_N20_H4_data_set.txt	46000	146	1178	995	1345	2036
S_N20_H5_data_set.txt	66500	175	1460	1144	272	4051
S_N20_H6_data_set.txt	142109,375	215	1771	1419	1573	4632
S_N20_H7_data_set.txt	103421,875	259	2041	1564	1545	6175
S_N20_H8_data_set.txt	106890,625	285	2343	1802	513	8811
S_N20_H9_data_set.txt	161375	320	2719	1982	2373	9541
S_N20_H10_data_set.txt	153156,25	385	3110	2275	2839	12056
C_N20_H4_data_set.txt	5675562,5	118	977	932	816	2492
C_N20_H5_data_set.txt	7106015,625	163	1237	1054	1103	2999
C_N20_H6_data_set.txt	7162250	187	1460	1324	1459	3862
C_N20_H7_data_set.txt	7180265,625	217	1717	1450	1278	5628
C_N20_H8_data_set.txt	7136531,25	260	1974	1652	1809	7417
C_N20_H9_data_set.txt	7158078,125	296	2228	1887	1752	8603
C_N20_H10_data_set.txt	7143625	333	2535	2020	2081	11124
MC_N20_H4_data_set.txt	105234,375	132	124	1053	948	1717
MC_N20_H5_data_set.txt	187750	158	161	1285	774	4102
MC_N20_H6_data_set.txt	257890,625	201	200	1449	589	4497
MC_N20_H7_data_set.txt	374015,625	247	246	1714	1263	5954
MC_N20_H8_data_set.txt	443734,375	275	280	1928	1905	7941
MC_N20_H9_data_set.txt	404062,5	321	299	2264	1817	9264
MC_N20_H10_data_set.txt	494921,875	354	353	2431	2469	11789
S_N50_H4_data_set.txt	7179430	360	14887	14398	14736	7686
S_N50_H5_data_set.txt	7180730	453	10450	15332	20438	14841
S_N50_H6_data_set.txt	7178230	601	12918	18493	25307	17747
S_N50_H7_data_set.txt	7178750	658	15197	21999	30217	25494
S_N50_H8_data_set.txt	7003870	773	17743	25184	14325	26563
S_N50_H9_data_set.txt	7181350	868	20693	28248	17124	31818
S_N50_H10_data_set.txt	7181570	1072	22997	31691	41277	40920
C_N50_H4_data_set.txt	7188480	285	8638	12717	9967	6486
C_N50_H5_data_set.txt	7180170	368	18091	14584	16970	8081
C_N50_H6_data_set.txt	7180430	445	22001	17498	19482	11827
C_N50_H7_data_set.txt	6887400	532	26428	20774	20869	16285
C_N50_H8_data_set.txt	7182500	622	30615	23347	20601	19542
C_N50_H9_data_set.txt	7181370	745	35101	26433	20633	24188
C_N50_H10_data_set.txt	7181320	825	39457	29505	23584	29620
MC_N50_H4_data_set.txt	7175600	313	12536	5998	11896	8298
MC_N50_H5_data_set.txt	7180510	419	17527	10677	18030	11154
MC_N50_H6_data_set.txt	7180840	502	21435	12729	5755	15479

## Appendice A – Risultati sperimentali

MC_N50_H7_data_set.txt	6652040	589	25932	15179	26546	21362
MC_N50_H8_data_set.txt	7184230	711	29948	17628	28777	26807
MC_N50_H9_data_set.txt	7181430	806	34319	20281	23027	34234
MC_N50_H10_data_set.txt	7178280	952	38897	21556	38830	42823
S_N80_H4_data_set.txt	7162090	499	57587	57129	10700	14525
S_N80_H5_data_set.txt	7182590	665	63221	76639	52619	19573
S_N80_H6_data_set.txt	7185420	881	101873	81602	59029	28524
S_N80_H7_data_set.txt	6958930	1008	122379	98695	111355	40730
S_N80_H8_data_set.txt	7185090	1180	144523	113229	125517	51971
S_N80_H9_data_set.txt	7185210	1338	166707	129231	127387	64089
S_N80_H10_data_set.txt	7183510	1526	190762	147021	137167	79711
C_N80_H4_data_set.txt	7172340	426	34245	31581	48398	11542
C_N80_H5_data_set.txt	7182230	562	71813	65793	65072	17589
C_N80_H6_data_set.txt	7183920	705	69411	81316	34854	24630
C_N80_H7_data_set.txt	7184120	831	82244	96675	91060	35253
C_N80_H8_data_set.txt	7185180	979	97528	114375	71519	49001
C_N80_H9_data_set.txt	7185450	1131	110440	128311	112031	61664
C_N80_H10_data_set.txt	7185570	1327	125877	148950	56369	73996
MC_N80_H4_data_set.txt	7183980	463	57857	55528	51091	12551
MC_N80_H5_data_set.txt	7182960	609	67113	35218	12356	20221
MC_N80_H6_data_set.txt	7182890	754	91353	85206	87223	29874
MC_N80_H7_data_set.txt	7184810	920	110658	101882	29023	39740
MC_N80_H8_data_set.txt	7180250	1059	129877	117925	122301	53309
MC_N80_H9_data_set.txt	7184600	1230	147622	134851	112996	66543
MC_N80_H10_data_set.txt	7187230	1422	170235	150723	135698	81952
S_N100_H4_data_set.txt	7159310	577	111636	102877	104592	15574
S_N100_H5_data_set.txt	7177060	818	117316	135963	118981	26852
S_N100_H6_data_set.txt	6974030	1051	165137	172539	178362	41047
S_N100_H7_data_set.txt	6920480	1221	198289	200778	216407	53183
S_N100_H8_data_set.txt	7160730	1505	237881	238169	216775	66840
S_N100_H9_data_set.txt	7126070	1673	271786	271600	281795	85844
S_N100_H10_data_set.txt	7085400	1931	308472	302885	200948	106339
C_N100_H4_data_set.txt	7157370	531	102432	93542	81765	15864
C_N100_H5_data_set.txt	7148060	805	138735	125817	111848	25845
C_N100_H6_data_set.txt	0	878	147885	130039	157961	37446
C_N100_H7_data_set.txt	0	1214	211616	158180	193097	50144
C_N100_H8_data_set.txt	0	1298	249480	184375	224790	66623
C_N100_H9_data_set.txt	0	1499	295196	213661	245348	83109
C_N100_H10_data_set.txt	0	1735	336978	241120	254890	103473
MC_N100_H4_data_set.txt	7177540	562	102003	98454	81026	15501
MC_N100_H5_data_set.txt	7190120	773	141443	118184	129246	23576
MC_N100_H6_data_set.txt	0	972	181019	44977	115133	38056
MC_N100_H7_data_set.txt	0	1140	192264	203343	152828	52210
MC_N100_H8_data_set.txt	0	1340	228844	232549	201585	68491
MC_N100_H9_data_set.txt	0	1546	270432	266927	262507	83322

MC_N100_H10_data_set.txt	0	1868	306298	302931	284186	102707
	<b>3420316,23</b>	<b>527,18</b>	<b>55844,24</b>	<b>50255,91</b>	<b>47216,60</b>	<b>21450,29</b>

## 8.5 *Variazione del parametro m*

Sono stati ottenuti i seguenti valori della funzione obiettivo:

dataset	CPLEX	ILS-Random	Rollout FlussoMedio	Rollout Domanda	Rollout Random	Cluster
S_N5_M4_data_set.txt	3856,2	3930,6	3930,6	3856,2	3934,2	3856,2
S_N5_M5_data_set.txt	4058,4	4058,4	4137,6	4058,4	4136,4	4237,8
S_N5_M6_data_set.txt	4260,6	4260,6	4339,8	4260,6	4338,6	4260,6
S_N5_M7_data_set.txt	4462,8	4540,8	4542	4462,8	4540,8	4645,8
S_N5_M8_data_set.txt	4665	4743	4744,2	4665	4665	4665
S_N5_M9_data_set.txt	4867,2	4945,2	4946,4	4867,2	4945,2	4867,2
S_N5_M10_data_set.txt	5069,4	5069,4	5148,6	5069,4	5147,4	5257,8
C_N5_M4_data_set.txt	3881,4	3881,4	3881,4	3881,4	3881,4	3888,6
C_N5_M5_data_set.txt	4071,6	4071,6	4071,6	4072,2	4071,6	4079,4
C_N5_M6_data_set.txt	4273,8	4273,8	4273,8	4274,4	4273,8	4281,6
C_N5_M7_data_set.txt	4476	4476	4476	4476	4476	4483,8
C_N5_M8_data_set.txt	4678,2	4678,2	4678,2	4678,2	4678,2	4686
C_N5_M9_data_set.txt	4880,4	4880,4	4880,4	4880,4	4880,4	4888,2
C_N5_M10_data_set.txt	5082,6	5082,6	5082,6	5082,6	5082,6	5090,4
MC_N5_M4_data_set.txt	3869,4	3876	3871,2	3881,4	3934,8	4084,8
MC_N5_M5_data_set.txt	4071,6	4073,4	4073,4	4073,4	4073,4	4298,4
MC_N5_M6_data_set.txt	4273,8	4273,8	4275,6	4275,6	4275,6	4512
MC_N5_M7_data_set.txt	4476	4477,8	4477,8	4477,8	4477,8	4725,6
MC_N5_M8_data_set.txt	4678,2	4678,2	4680	4678,2	4680	4939,2
MC_N5_M9_data_set.txt	4880,4	4880,4	4882,2	4880,4	4882,2	5152,8
MC_N5_M10_data_set.txt	5082,6	5082,6	5084,4	5082,6	5084,4	5366,4
S_N10_M4_data_set.txt	7745,4	7884,6	7884,6	7902,6	7884,6	7810,8
S_N10_M5_data_set.txt	8078,4	8193,6	8193,6	8193,6	8262,6	8122,8
S_N10_M6_data_set.txt	8452,8	8580	8511	8580	8580	8503,2
S_N10_M7_data_set.txt	8833,2	8891,4	8899,8	8891,4	8891,4	8883,6
S_N10_M8_data_set.txt	9213,6	9280,2	9280,2	9296,4	9271,8	9264
S_N10_M9_data_set.txt	9594	9660,6	9652,2	9676,8	9652,2	9644,4
S_N10_M10_data_set.txt	9974,39	10029	9987,6	10005	9987,6	10024,8
C_N10_M4_data_set.txt	7944,6	8074,2	7914	7922,4	7975,8	7916,4
C_N10_M5_data_set.txt	8186,39	8296,8	8236,8	8240,4	8243,4	8257,2
C_N10_M6_data_set.txt	8509,79	8620,2	8555,4	8611,2	8541	8626,2
C_N10_M7_data_set.txt	8869,19	9010,8	8914,8	8872,2	8970	8922,6
C_N10_M8_data_set.txt	9249,6	9313,2	9282	9286,8	9309	9290,4
C_N10_M9_data_set.txt	9630	9678,6	9660,6	9657,6	9647,4	9670,8

## Appendice A – Risultati sperimentali

C_N10_M10_data_set.txt	10010,39	10080,6	10029	10029,6	10023,6	10051,2
MC_N10_M4_data_set.txt	7784,99	7978,8	7932,6	7911,6	7915,8	7942,2
MC_N10_M5_data_set.txt	8125,19	8275,2	8286	8229	8280	8158,2
MC_N10_M6_data_set.txt	8465,4	8611,2	8601,6	8591,4	8603,4	8500,8
MC_N10_M7_data_set.txt	8845,8	8944,2	8934,6	8906,4	8906,4	8884,8
MC_N10_M8_data_set.txt	9226,19	9264	9282	9255	9252,6	9269,4
MC_N10_M9_data_set.txt	9606,59	9739,8	9649,8	9649,8	9657,6	9654
MC_N10_M10_data_set.txt	9987	10013,4	10029	10016,4	10032	10038,6
S_N20_M4_data_set.txt	16616,4	16853,4	16904,4	16725,6	16768,8	16783,8
S_N20_M5_data_set.txt	17236,2	17548,8	17356,8	17421,6	17398,2	17511,6
S_N20_M6_data_set.txt	17904	18367,8	18281,4	18291,6	18063	17984,4
S_N20_M7_data_set.txt	18612,6	19123,2	19018,8	18982,8	18865,2	18828
S_N20_M8_data_set.txt	19330,2	19788	19710,6	19776	19731	19596,6
S_N20_M9_data_set.txt	20075,4	20461,2	20445,6	20475	20300,4	20260,2
S_N20_M10_data_set.txt	20835	21106,2	21087	21096,6	21037,2	21075,6
C_N20_M4_data_set.txt	16835,99	17068,8	16822,8	16846,2	16872	17029,8
C_N20_M5_data_set.txt	17347,2	17626,8	17514	17582,4	17496	17635,8
C_N20_M6_data_set.txt	18016,2	18466,2	18277,2	18358,8	18190,8	18187,8
C_N20_M7_data_set.txt	18719,4	19169,4	18999	19006,8	18982,8	18959,4
C_N20_M8_data_set.txt	19425,6	19903,2	19648,2	19747,2	19732,2	19615,8
C_N20_M9_data_set.txt	20075,4	20533,2	20517	20540,4	20533,8	20241
C_N20_M10_data_set.txt	20884,2	21345	21236,4	21261	21322,8	20935,8
MC_N20_M4_data_set.txt	16669,2	16810,8	16887,6	16926	16733,4	16936,8
MC_N20_M5_data_set.txt	17288,99	17593,2	17565	17587,2	17449,2	17455,2
MC_N20_M6_data_set.txt	17926,2	18393	18300	18295,8	18118,2	18187,8
MC_N20_M7_data_set.txt	18634,19	19090,2	18964,8	19006,2	18904,8	18828
MC_N20_M8_data_set.txt	19336,8	19822,2	19682,4	19762,8	19713,6	19513,8
MC_N20_M9_data_set.txt	20086,79	20604	20406,6	20525,4	20290,2	20188,8
MC_N20_M10_data_set.txt	20842,8	21192,6	21114,6	21142,2	21025,8	20958
S_N50_M4_data_set.txt	45881,4	46768,8	46264,8	46696,8	46181,4	47068,2
S_N50_M5_data_set.txt	46918,79	47941,2	47568	48064,8	47464,2	48324
S_N50_M6_data_set.txt	48445,79	49528,2	49162,8	49395	49174,2	49098
S_N50_M7_data_set.txt	50017,2	51222	50938,2	51556,8	50800,8	50839,2
S_N50_M8_data_set.txt	51648,6	53316	52427,4	53391,6	52749	52604,4
S_N50_M9_data_set.txt	53315,4	54694,2	54297,6	54525	54548,4	54109,8
S_N50_M10_data_set.txt	55098,6	56555,4	56274	56798,4	56392,2	55860
C_N50_M4_data_set.txt	46468,79	46927,2	46877,4	47284,2	46569,6	47583,6
C_N50_M5_data_set.txt	47336,4	47815,8	47700	48133,8	47779,2	48296,4
C_N50_M6_data_set.txt	48666,6	49615,2	49149	49380	49193,4	49449
C_N50_M7_data_set.txt	50259,6	51379,2	50958	51101,4	50897,4	50956,2
C_N50_M8_data_set.txt	51777,6	53192,4	52756,2	52931,4	52685,4	52677
C_N50_M9_data_set.txt	53577,6	55042,8	54529,2	54967,2	54507,6	54181,2
C_N50_M10_data_set.txt	55209	56902,2	56487,6	56425,8	56237,4	55886,4
MC_N50_M4_data_set.txt	45994,2	46679,4	46485,6	47056,2	46637,4	47422,8
MC_N50_M5_data_set.txt	47116,79	48189,6	47446,8	48609	47559,6	47780,4

## Appendice A – Risultati sperimentali

MC_N50_M6_data_set.txt	48496,2	49771,2	49071	49435,8	49140,6	49638,6
MC_N50_M7_data_set.txt	50102,4	51363,6	50793	51199,8	50833,2	51102
MC_N50_M8_data_set.txt	51668,2	53022,6	52609,2	53302,2	52654,8	52646,4
MC_N50_M9_data_set.txt	53344,2	54852	54493,8	55013,4	54525	53770,2
MC_N50_M10_data_set.txt	55082,4	56692,2	56181,6	56465,4	56259,6	55717,2
S_N80_M4_data_set.txt	78247,8	80260,8	78671,4	80252,4	79381,2	80569,8
S_N80_M5_data_set.txt	79789,8	80796	80178,6	81644,4	80112,6	80887,2
S_N80_M6_data_set.txt	81360,59	83209,2	82279,8	84532,2	82423,8	82999,8
S_N80_M7_data_set.txt	83708,4	85762,2	84657,6	86669,4	84829,2	84682,2
S_N80_M8_data_set.txt	86761,8	87675	87126,6	88708,8	86914,8	87238,2
S_N80_M9_data_set.txt	88560,59	90832,2	90244,8	92103	90093	89808
S_N80_M10_data_set.txt	91185	93103,8	92559,6	94711,8	92677,2	92761,8
C_N80_M4_data_set.txt	79641,59	79762,8	79045,2	82023,6	79669,8	81041,4
C_N80_M5_data_set.txt	80389,8	81420,6	80239,8	81805,2	80205	80756,4
C_N80_M6_data_set.txt	84139,2	83322	82478,4	84498,6	82123,2	82600,8
C_N80_M7_data_set.txt	84376,8	85859,4	84928,8	86676	85237,2	84706,2
C_N80_M8_data_set.txt	87199,2	88500,6	87576,6	88510,8	87340,2	86904
C_N80_M9_data_set.txt	90064,79	90976,8	90141,6	91186,8	90050,4	89164,8
C_N80_M10_data_set.txt	91805,4	93476,4	92770,8	93862,8	92722,8	92384,4
MC_N80_M4_data_set.txt	78642	80029,8	78902,4	80395,8	78970,8	79866
MC_N80_M5_data_set.txt	80257,79	80919,6	80376,6	81520,8	80022	81144,6
MC_N80_M6_data_set.txt	84535,8	82866,6	82295,4	83474,4	82270,8	82666,8
MC_N80_M7_data_set.txt	84144,6	85599	84801,6	85273,2	85139,4	85060,2
MC_N80_M8_data_set.txt	86536,8	87864	87581,4	88545	87354,6	87343,8
MC_N80_M9_data_set.txt	88544,4	90954,6	90101,4	91096,8	89905,8	89700
MC_N80_M10_data_set.txt	91276,79	93794,4	93208,8	94108,8	92897,4	92593,2
S_N100_M4_data_set.txt	100862,4	101907	101645,4	102954	100984,8	102964,2
S_N100_M5_data_set.txt	101347,2	103470	102421,2	105327	103169,4	104308,2
S_N100_M6_data_set.txt	104736	105606	104270,4	105515,4	103821,6	105715,8
S_N100_M7_data_set.txt	106151,4	107727	106993,2	108112,2	107147,4	107690,4
S_N100_M8_data_set.txt	119231,8	111590,4	110406	112825,2	110308,2	110794,2
S_N100_M9_data_set.txt	142048,48	114627	113440,2	114864	113560,2	113154
S_N100_M10_data_set.txt	254355,48	117964,2	117006	118187,4	117103,2	116516,4
C_N100_M4_data_set.txt	102031,8	102200,4	101746,2	103117,8	102096	104005,8
C_N100_M5_data_set.txt	0	103170	102458,4	105314,4	102117,6	104631,6
C_N100_M6_data_set.txt	0	105064,2	104470,2	106159,8	104661,6	105531,6
C_N100_M7_data_set.txt	0	107629,2	106991,4	107949,6	106813,2	107692,2
C_N100_M8_data_set.txt	0	111255	110314,8	111748,8	110042,4	110644,2
C_N100_M9_data_set.txt	0	114757,8	114393,6	114520,8	113913	113206,8
C_N100_M10_data_set.txt	0	118052,4	117791,4	119067,6	117513,6	116665,8
MC_N100_M4_data_set.txt	120662,59	102106,8	101339,4	103197	101169,6	104661,6
MC_N100_M5_data_set.txt	133188,55	102677,4	102868,8	105104,4	102823,8	103812
MC_N100_M6_data_set.txt	126557,82	104703	104063,4	105909,6	104776,2	105354
MC_N100_M7_data_set.txt	285687,71	107662,8	107436,6	109703,4	107017,2	107769
MC_N100_M8_data_set.txt	0	110809,2	110736,6	111561	110354,4	110426,4

## Appendice A – Risultati sperimentali

MC_N100_M9_data_set.txt	0	114738	113686,2	114874,8	113336,4	113540,4
MC_N100_M10_data_set.txt	0	118413	117007,8	118376,4	117164,4	117020,4
	<b>41328,60</b>	<b>46492,93</b>	<b>46151,28</b>	<b>46736,05</b>	<b>46134,15</b>	<b>46320,74</b>

Per quanto riguarda i tempi di elaborazione sono stati rilevati i seguenti valori:

dataset	CPLEX	ILS-Rollout	Rollout FlussoMedio	Rollout Domanda	Rollout Random	Cluster
S_N5_M4_data_set.txt	875	34	8	6	1	451
S_N5_M5_data_set.txt	1078,125	37	3	19	19	585
S_N5_M6_data_set.txt	1468,75	48	3	8	9	1013
S_N5_M7_data_set.txt	1625	67	3	8	17	815
S_N5_M8_data_set.txt	1953,125	60	7	11	4	1453
S_N5_M9_data_set.txt	2296,875	65	4	10	9	1754
S_N5_M10_data_set.txt	2671,875	90	8	11	23	1532
C_N5_M4_data_set.txt	2062,5	30	9	2	17	399
C_N5_M5_data_set.txt	2468,75	43	10	18	8	601
C_N5_M6_data_set.txt	4421,875	59	11	12	16	863
C_N5_M7_data_set.txt	3453,125	63	14	16	19	1199
C_N5_M8_data_set.txt	4593,75	72	19	25	7	1618
C_N5_M9_data_set.txt	6390,625	78	19	20	12	2028
C_N5_M10_data_set.txt	6046,875	97	40	40	12	2521
MC_N5_M4_data_set.txt	1234,375	33	7	1	5	305
MC_N5_M5_data_set.txt	1859,375	47	12	4	6	383
MC_N5_M6_data_set.txt	4187,5	66	32	2	14	484
MC_N5_M7_data_set.txt	3437,5	62	27	5	20	666
MC_N5_M8_data_set.txt	3343,75	63	15	4	10	904
MC_N5_M9_data_set.txt	3890,625	84	20	4	2	1108
MC_N5_M10_data_set.txt	4375	80	26	8	4	1302
S_N10_M4_data_set.txt	2750	81	3	58	128	768
S_N10_M5_data_set.txt	5328,125	100	216	210	29	1084
S_N10_M6_data_set.txt	7203,125	99	201	140	50	1520
S_N10_M7_data_set.txt	5296,875	138	259	194	252	2018
S_N10_M8_data_set.txt	5937,5	128	6	173	249	2683
S_N10_M9_data_set.txt	7031,25	155	195	230	184	3455
S_N10_M10_data_set.txt	8578,125	176	331	170	33	4258
C_N10_M4_data_set.txt	47312,5	73	91	114	144	699
C_N10_M5_data_set.txt	37921,875	80	82	183	96	1137
C_N10_M6_data_set.txt	24500	95	216	215	101	1544
C_N10_M7_data_set.txt	15468,75	110	92	82	53	2061
C_N10_M8_data_set.txt	23187,5	119	221	222	80	2658
C_N10_M9_data_set.txt	19390,625	140	18	291	309	3445
C_N10_M10_data_set.txt	23750	158	223	120	236	4354
MC_N10_M4_data_set.txt	8468,75	77	110	43	128	827

Appendice A – Risultati sperimentali

MC_N10_M5_data_set.txt	17640,625	88	96	138	164	1123
MC_N10_M6_data_set.txt	10968,75	103	62	174	223	1597
MC_N10_M7_data_set.txt	8625	115	250	111	226	2100
MC_N10_M8_data_set.txt	9906,25	131	13	156	276	2725
MC_N10_M9_data_set.txt	12265,625	146	261	158	44	3466
MC_N10_M10_data_set.txt	14140,625	160	329	206	238	4557
S_N20_M4_data_set.txt	46000	146	1152	934	976	1773
S_N20_M5_data_set.txt	86937,5	176	1273	1276	1129	2603
S_N20_M6_data_set.txt	108781,25	206	1711	1201	1591	3575
S_N20_M7_data_set.txt	380765,625	240	1184	1124	1081	4441
S_N20_M8_data_set.txt	268062,5	268	2223	1919	1592	6160
S_N20_M9_data_set.txt	326796,875	301	2189	2063	1539	7757
S_N20_M10_data_set.txt	344812,5	329	2378	1223	1896	9021
C_N20_M4_data_set.txt	5675562,5	130	981	915	942	1639
C_N20_M5_data_set.txt	1081000	165	1306	888	871	2593
C_N20_M6_data_set.txt	940937,5	184	1124	960	1223	3642
C_N20_M7_data_set.txt	2397359,375	228	520	1206	1213	4529
C_N20_M8_data_set.txt	1567562,5	243	1966	1813	1577	6170
C_N20_M9_data_set.txt	326796,875	268	1879	1224	2077	7965
C_N20_M10_data_set.txt	550953,125	309	1104	1711	2196	10200
MC_N20_M4_data_set.txt	105234,375	135	131	981	502	1690
MC_N20_M5_data_set.txt	197437,5	169	173	1159	1013	2739
MC_N20_M6_data_set.txt	159687,5	193	214	853	1161	4054
MC_N20_M7_data_set.txt	513578,125	219	792	884	1041	4792
MC_N20_M8_data_set.txt	304625	251	1912	1659	1999	6291
MC_N20_M9_data_set.txt	333234,375	289	1059	616	1247	8522
MC_N20_M10_data_set.txt	268218,75	322	2484	1407	2544	10505
S_N50_M4_data_set.txt	7179437,5	357	14939	14562	15967	6282
S_N50_M5_data_set.txt	7183296,875	435	20661	13781	20051	8276
S_N50_M6_data_set.txt	7111734,375	530	21591	14994	25007	12613
S_N50_M7_data_set.txt	7184578,125	604	28950	25925	25989	16662
S_N50_M8_data_set.txt	6853718,75	684	31167	12106	29103	19666
S_N50_M9_data_set.txt	7187296,875	775	34905	31738	32106	24300
S_N50_M10_data_set.txt	7185234,375	840	30256	34580	1908	30439
C_N50_M4_data_set.txt	7188484,375	284	8666	12598	13708	6251
C_N50_M5_data_set.txt	7183312,5	378	16293	17058	17328	8488
C_N50_M6_data_set.txt	6740750	451	21503	19580	19815	12106
C_N50_M7_data_set.txt	7185531,25	534	24890	17102	24886	16995
C_N50_M8_data_set.txt	7187140,625	613	28501	12289	26088	20234
C_N50_M9_data_set.txt	7188437,5	701	9501	11494	30025	28110
C_N50_M10_data_set.txt	7132640,625	787	4975	22779	28465	30564
MC_N50_M4_data_set.txt	7175609,375	315	12591	5939	13267	5776
MC_N50_M5_data_set.txt	7183281,25	432	20330	14061	16559	8332
MC_N50_M6_data_set.txt	7059828,125	496	22574	15870	19294	12066
MC_N50_M7_data_set.txt	7186187,5	576	27431	21096	15456	16519

## Appendice A – Risultati sperimentali

MC_N50_M8_data_set.txt	7185984,375	652	26124	22614	21344	20557
MC_N50_M9_data_set.txt	7176515,625	835	34063	26096	21835	27044
MC_N50_M10_data_set.txt	7185859,375	796	30168	27199	34330	31071
S_N80_M4_data_set.txt	7162093,75	508	58623	57646	49723	12050
S_N80_M5_data_set.txt	7099718,75	663	79515	80612	73191	17952
S_N80_M6_data_set.txt	7006125	784	87653	95759	83929	26696
S_N80_M7_data_set.txt	7119562,5	973	82078	103438	96144	36433
S_N80_M8_data_set.txt	7000906,25	1085	128427	116423	117966	44447
S_N80_M9_data_set.txt	7183546,875	1260	128357	53163	119776	54051
S_N80_M10_data_set.txt	7154671,875	1404	148411	133044	133947	69816
C_N80_M4_data_set.txt	7172343,75	419	34631	31948	48657	11795
C_N80_M5_data_set.txt	7189000	590	65116	63817	17356	18226
C_N80_M6_data_set.txt	7135718,75	708	46005	78063	80313	27399
C_N80_M7_data_set.txt	7139828,125	900	105058	98148	95558	34454
C_N80_M8_data_set.txt	7155093,75	1052	92120	100855	52540	49159
C_N80_M9_data_set.txt	7174437,5	1126	120047	110152	122538	57961
C_N80_M10_data_set.txt	7128343,75	1322	139786	133668	124981	70594
MC_N80_M4_data_set.txt	7183984,375	491	57516	54916	52968	11053
MC_N80_M5_data_set.txt	7023843,75	653	76720	74786	47966	19694
MC_N80_M6_data_set.txt	7165750	770	83005	66549	79361	26428
MC_N80_M7_data_set.txt	7172375	980	88733	94585	91285	35348
MC_N80_M8_data_set.txt	7176406,25	1087	100073	92301	92488	45552
MC_N80_M9_data_set.txt	7130781,25	1454	134093	110053	126451	61098
MC_N80_M10_data_set.txt	6987390,625	1703	52987	119031	130661	66060
S_N100_M4_data_set.txt	7159312,5	658	110809	103670	68225	16452
S_N100_M5_data_set.txt	7162656,25	861	148692	117241	137835	26346
S_N100_M6_data_set.txt	6952000	1027	186611	145516	140768	40854
S_N100_M7_data_set.txt	7151750	1338	214200	190622	138615	52003
S_N100_M8_data_set.txt	7120125	1494	215966	238990	168953	63803
S_N100_M9_data_set.txt	7177875	1587	261388	197363	215353	79967
S_N100_M10_data_set.txt	7199265,625	1779	293132	289832	271316	98222
C_N100_M4_data_set.txt	7157375	532	103416	93471	79911	15065
C_N100_M5_data_set.txt	7065750	734	137209	108682	125292	25625
C_N100_M6_data_set.txt	7167281,25	954	171526	152906	161397	39602
C_N100_M7_data_set.txt	7179281,25	1186	203052	194638	188075	54201
C_N100_M8_data_set.txt	7052734,375	1285	198627	221267	231661	67061
C_N100_M9_data_set.txt	7121078,125	1465	19878	188903	242738	83333
C_N100_M10_data_set.txt	7194203,125	1667	289309	249665	269316	100718
MC_N100_M4_data_set.txt	7177546,875	570	101881	100056	30118	16640
MC_N100_M5_data_set.txt	7177671,875	766	140335	133988	138248	24313
MC_N100_M6_data_set.txt	7173281,25	948	175569	151106	162052	37580
MC_N100_M7_data_set.txt	7116968,75	1154	205434	192008	183351	54258
MC_N100_M8_data_set.txt	7179450	1348	219813	221223	186473	68461
MC_N100_M9_data_set.txt	7165250	1576	252534	217930	236867	88569
MC_N100_M10_data_set.txt	7172125	1733	256304	276430	257892	111095

## Appendice A – Risultati sperimentali

---

	<b>3697567,56</b>	<b>517,38</b>	<b>50127,34</b>	<b>48234,63</b>	<b>47283,12</b>	<b>19869,12</b>
--	-------------------	---------------	-----------------	-----------------	-----------------	-----------------

## 9 Bibliografia

1. **Ballou Ronald H.**, (1992). *Business Logistics Management. Planning, Organizing and Controlling the Supply Chain*. Prentice-Hall International, 4th.
2. **Bassan Y., Roll Y. and Rosenblatt M.J.**, (1980). *Internal layout design of a warehouse*. AIIE Transactions, 12 (4) pp. 317-322.
3. **Caron F., Marchet G. and Perego A.**, (1998). *Routing policies and COI-based storage policies in picker-to-part systems*. International journal of Production Research.
4. **Christofides N. and Colloff I.**, (1972). *The rearrangement of items in a warehouse*. Operation Research. 21 pp. 577-589.
5. **De Koster R., Le-Duc T. and Roodbergen K. J.**, (2007). *Design and control of warehouse order picking: a literature review*. European Journal of Operational Research.
6. **Drury J.**, (1988). *Towards more efficient order picking*. The Institute of Materials Management, Cranfield, U.K.
7. **Eberhart R.C. and Kennedy J.**, (1995). *A new optimizer using particle swarm theory*. In Proceedings of 6th Symposium Micro Machine and Human Science, Nagoya, pp. 39-43.
8. **Frazelle E.H.**, (2002). *World-Class Warehousing and Material Handling*. New York.
9. **Ghiani G. and Musmanno R.**, (2000). *Modelli e metodi per l'organizzazione dei sistemi logistici*. Pitagora Editrice Bologna.
10. **Gibson D. R. and Sharp G. P.**, (1992). *Order batching procedures*. European Journal of Operational Research. 58 (1), pp. 57-67.
11. **Goetschalckx M.**, (1998). *A review of unit load storage policies in warehouse operations*. In Proceedings of EURO XVI Conference, Brussels, July 12–15.
12. **Goetschalckx M. and Ashayeri J.**, (1989). *Classification and design of order picking systems*. Logistics World, pp. 99-106.
13. **Goetschalckx M. and Ratliff H.D.**, (1990). *Shared storage policies based on the duration stay of unit loads*. Management Science, 36 (9), pp. 1120-1132.
14. **Graves S. C., Hausman W. H. and Schwarz L. B.**, (1977). *Storage-retrieval interleaving in automatic warehousing systems*. Management Science, 23, pp. 935-945.
15. **Gray A. E., Karmarkar U. S. and Seidmann A.**, (1992). *Desing and operation of an order-consolidation warehouse: models and applications*. European Journal of Operational Research, 58, pp. 14-36.

16. **Gu J., Goetschalckx M. and McGinnis L. F.**, (2010). *Research on warehouse design and performance evaluation: A comprehensive review*. European Journal of Operational Research, 203, pp. 539-549.
17. **Gu J., Goetschalckx M. and McGinnis L. F.**, (2007). *Research on warehouse operation: A comprehensive review*. European Journal of Operational Research, 177, pp. 1-21.
18. **Harmatuck D.J.**, (1976). *A comparison of two approaches to stock location*. The Logistics and Transportation review, 12 (4), pp. 282-284.
19. **Hausman W. H., Schwarz L. B. and Graves S. C.**, (1976). *Optimal storage assignment in automatic warehousing systems*. Management Science, 22 (6), pp. 629-638.
20. **Heragu S.S., Du L., Mantel R. J., Schuur P. C.**, (2005). *Mathematical model for warehouse design and product allocation*. International Journal of Production Research, 43 (2), pp. 327-338.
21. **Heskett J. L.**, (1963). *Cube-per-order index - a key to warehouse stock location*. Transportation and Distribution Management, 3, pp. 27-31.
22. **Heskett J. L.**, (1964). *Putting the Cube-per-order index to work in warehouse layout*. Transportation and Distribution Management, 4, pp. 23-30.
23. **Hwang H., Yong H. O. and Cha C. N.**, (2003). *A stock location rule for a low level picker-to-part system*. Engineering Optimization.
24. **Jaikumar R. and Solomon M.M.**, (1990). *Dynamic operational policies in an automated warehouse*. IIE Transactions, 22 (4), pp. 370-376.
25. **Jarvis J. M. and McDowell E. D.**, (1991). *Optimal product layout in an order picking warehouse*. IIE Transactions, 23 (1), pp. 93-102.
26. **Kallina C. and Lynn J.**, (1976). *Application of the cube-per-order index rule for stock location in a distribution warehouse*. Interfaces, 7 (1), pp. 37-46.
27. **Larson T. N., March H. and Kusiak A.**, (1997). *A heuristic approach to warehouse layout with class-based storage*. IIE Transactions, 29, pp. 337-348.
28. **Le-Duc T. and De Koster R.**, (2005c). *Travel distance estimation and storage zone optimization in a 2-block class-based storage strategy warehouse*. International Journal of Production Research, pp. 3561-3581.
29. **Le-Duc T.**, (2005). *Design and control of efficient order picking processes*. PhD thesis, RSM Erasmus University.
30. **Mallette A. J. and Francis R. L.**, (1972). *A generalizad assignment approach to optimal facility layout*. IIE Transactions.
31. **Malmborg C.J. and Bhaskaran K.**, (1990). *A revised proof of optimality for cube-per-order index rule for stored item location*. Applied Mathematical Modelling.

32. **Muppani (Muppant) and Adil G.K.**, (2005). *Storage class formation considering space and handling cost for warehouse planning*. In IX Annual International Conference of the Society of Operations Management.
33. **Muralidharan B., Linn R. J. and Pandit R.**, (1995). *Shuffling heuristics for the storage location assignment in a AS/RS*. International Journal of Production Research, 33 (6), pp. 1661-1672.
34. **Önüt Semih, Tuzkaya Umut R. and Doğaç Bilgehan**, (2008). *A particle swarm optimization algorithm for the multiple-level warehouse layout design problem*. Computer & industrial Engineering, 54, pp. 783-799.
35. **Petersen C. G. and Aase G.**, (2004). *A comparison of picking, storage, and routing policies in manual order picking*. International Journal of Production Economics.
36. **Petersen C. G. and Schmenner R.W.**, (1999). *An evaluation of routing and volume-based storage policies in an order picking operation*. Decision Science.
37. **Petersen C. G.**, (1999). *The impact of routing and storage policies in warehouse efficiency*. International Journal of Operations and Production Management.
38. **Petersen C.G.**, (2000). *An evaluation of order picking policies for mail order companies*. Production and Operations Management, 9(4), pp. 319-335.
39. **Petersen C.G., Aase G. and Heiser D.R.**, (2004). *Improving order-picking performance through the implementation of class-based storage*. International Journal of Physical Distribution & Logistics Management, 34(7), pp. 534-544.
40. **Reza Zanjirani Farahani and Masoud Hekmatfar**, (2009). *Facility Location: Concepts, Models, Algorithms and Case Studies*.
41. **Roll Y. and Rosenblatt M.J.**, (1987). *Shifting in warehouse*. Material Flow 4, pp. 147-157.
42. **Sadiq M., Landers T.L. and Taylor G.D.**, (1996). *An assignment algorithm for dynamic picking systems*. IIE Transactions, 28, pp. 607-616.
43. **Sanei O., Nasiri V., Marjani M.R., Moattar Hussein S.M.**, (2011). *A heuristic algorithm for the warehouse space assignment problem considering operational constraints: with application in a case study*. In Proceedings of the 2011 International Conference on Industrial Engineering and Operations Management, Kuala Lumpur, Malaysia, January 22-24.
44. **Shwarz L. B., Graves S. C. and Hausman W. H.**, (1978). *Scheduling policies for automatic warehousing systems: simulation results*. IIE Transactions, 10 (3), pp. 260-270.
45. **Tompkins J. A., White J. A., Bozer Y. A., Frazelle E. H., Tanchoco J. M. A., Trevino J.**, (2003). *Facilities planning*.
46. **Van den Berg J.P.**, (1999). *A literature survey on planning and control of warehousing systems*. IIE Transactions, 31, pp. 751-762.

47. **Van den Berg J.P.**, (1996). *Class-based storage allocation in a single command warehouse with space requirement constraints*. International Journal of Industrial Engineering, 3(1), pp. 21-28.
48. **Van der Berg J. P. and Zijm W. H. M.**, (1999). *Models for warehouse management: Classification and examples*. International journal of Production Economics, 59, pp. 519-528.
49. **Viale D.**, (1997). *La gestione del magazzino*. Franco Angeli.
50. **Vignati Gianfranco**, (2002). *Manuale di logistica. Magazzino-Antinfortunistica-Informatica-Aspetti legali-Assicurazione*. Ulrico Hoepli Milano, 1.
51. **Wilson H. G.**, (1977). *Order quantity, product popularity, and the location of stock in a warehouse*. AIIE Transactions, 9 (3), pp. 230-237.