

POLITECNICO DI TORINO

Corso di Laurea Magistrale
in Ingegneria Matematica

Tesi di Laurea Magistrale

**Gestione delle scorte di beni deperibili in un sistema
multi-echelon**



Relatore
prof. Paolo Brandimarte

Candidato
Lucio Moro

Anno Accademico 2023-2024

Sommario

Questa tesi studia un problema relativo all'ottimizzazione del processo di gestione degli stock per un prodotto deperibile in un ambiente multi-echelon con due livelli distinti: magazzino e rivenditori. Viene considerato un solo prodotto con tempo di vita costante ed ogni volta che la vita di uno stock termina esso viene scartato. Il sistema presenta tempi di consegna ai rivenditori costanti e non sono presenti costi di trasporto. L'obiettivo è quello di massimizzare il profitto e minimizzare contemporaneamente i costi associati al mantenimento degli stock e la penalità associata alle vendite perse (non è presente backlog degli ordini), considerando sia il tempo di vita del prodotto deperibile che le specificità del sistema multi-echelon.

Per fare ciò è stato implementato un ambiente che simula il sistema tenendo conto di variabili cruciali come la domanda (simulata da due diverse distribuzioni normali) e la politica di acquisto (FIFO, LIFO) oltre a considerare le relazioni tra il magazzino centrale e i rivenditori.

Vengono implementate due policy di ordinazione: Base-stock policy e Constant order policy. Per ottimizzarne i parametri viene utilizzato un metodo euristico chiamato Particle Swarm Optimization. Al termine, le performance delle soluzioni così trovate sono messe a confronto sulla base di differenti scenari di configurazione del sistema.

Indice

Elenco delle tabelle	3
Elenco delle figure	4
1 Letteratura	7
2 Dinamica del sistema	9
2.1 Assunzioni	9
2.2 Stato, azioni e contributo immediato	10
2.3 Politiche implementate	11
2.3.1 Politica di ordinazione costante	11
2.3.2 Politica di ordinazione base-stock	12
2.3.3 Obiettivo	14
3 Implementazione del simulatore	15
3.1 Ottimizzazione	19
4 Esperimenti computazionali e analisi dei risultati	20
5 Conclusioni	31

Elenco delle tabelle

2.1	Notazione	12
4.1	Distribuzioni	21
4.2	Configurazioni	21
4.3	BSP Scenario 1	25
4.4	BSP Scenario 2	26
4.5	BSP Scenario 3	27
4.6	CP Scenario 1	28
4.7	CP Scenario 2	29
4.8	CP Scenario 3	30

Elenco delle figure

1.1	Strutture multi-echelon	8
2.1	Evoluzione giornaliera	13
3.1	UML	18
4.1	Sistema	20
4.2	Profitti divisi per margini	22
4.3	Vendite	23
4.4	Scarti FIFO LIFO	24

Introduzione

Oggi giorno la gestione dell'inventario è ritenuta un'attività fondamentale per il successo in molte aziende. Tra le ragioni abbiamo la necessità di garantire un determinato livello di servizio nei confronti del cliente ma allo stesso tempo vi è il bisogno di contenere i costi (magazzino, ordinazione, trasporto, ecc.). Le due cose però non vanno esattamente nella stessa direzione: l'incertezza della domanda richiede di mantenere un'adeguata quantità di scorte per evitare carenze che portino a delle mancate vendite, ma per fare in modo che questo sia possibile si incorre nel problema dei costi sostenuti dal capitale vincolato nei magazzini, oppure se non si volessero tenere scorte elevate ci sarebbero i continui costi di spedizione da tenere in considerazione. Per trovare il giusto equilibrio tra il livello di servizio e i costi dell'inventario è necessaria una gestione dell'approvvigionamento accurata. Molte aziende tendono ad effettuare questa ottimizzazione solo a livello locale, quindi approcciandosi ad un problema di tipo single-echelon senza pensare a tutta la catena di rifornimento, per motivi di maggiore semplicità. Tuttavia, con l'avanzamento delle tecnologie ci si può confrontare con scenari più complessi dove si punta ad ottimizzare politiche all'interno di un sistema multi-echelon. Studi mostrano come la presenza di un decisore centralizzato, il cui ruolo sia di coordinare le politiche da adottare, permetta il raggiungimento di un risultato migliore.

Entra in gioco una complicazione ulteriore se i prodotti con cui si lavora hanno un tempo di vita limitato. La gestione delle scorte di beni deperibili rappresenta una sfida critica per molte industrie, inclusa l'agroalimentare, la farmaceutica e quella dei prodotti freschi. La freschezza del prodotto è cruciale non solo per garantire la soddisfazione del cliente, ma anche per la sicurezza alimentare e la conformità normativa. Il deterioramento dei beni nel tempo rende necessario un monitoraggio costante e una gestione attenta delle scorte per garantire che i prodotti siano venduti prima che perdano la loro qualità o superino la data di scadenza. Ciò richiede l'implementazione di politiche di riordino e allocazione delle scorte mirate, tenendo conto della freschezza del prodotto e delle esigenze della catena di distribuzione.

Tenere in considerazione la modalità di approvvigionamento dei clienti apporta un ulteriore livello di difficoltà al problema trattato. I clienti che si recano al supermercato per comprare prodotti freschi come yogurt, frutta o della carne, possono decidere di preferire prodotti prossimi alla scadenza, preoccupandosi per l'ambiente o avendo un occhio di riguardo per la lotta allo spreco. In questo caso i beni vengono prelevati con una politica FIFO (First-In-First-Out), al contrario della modalità di acquisto LIFO (Last-In-First-Out), con la quale si preferiscono prodotti con maggior tempo di vita restante, rischiando

di rimanere con merce in scadenza lasciata sugli scaffali. Questi due casi rappresentano gli estremi, ma è possibile considerare anche un mix delle due politiche, o ancora un tipo di scelta che avviene in maniera del tutto casuale.

Capitolo 1

Letteratura

Il termine multi-echelon viene utilizzato per le catene di approvvigionamento in cui un articolo passa attraverso più di una fase (i.e., stabilimento) prima di raggiungere il cliente finale. Lo studio di queste catene è decisamente più complesso di uno sul classico problema single-echelon siccome il tempo di consegna dell'ordine effettuato in una posizione della catena non dipende esclusivamente dal normale tempo che impiegherebbe ad arrivare dal punto di distribuzione ma anche dalla disponibilità che esso ha. La gestione delle scorte di beni deperibili in un sistema multi-echelon rappresenta una sfida complessa e cruciale per molte industrie, inclusa quella agroalimentare, farmaceutica e della logistica di prodotti freschi. L'ottimizzazione di questo processo richiede l'equilibrio tra la freschezza del prodotto e la minimizzazione delle perdite dovute al deterioramento. Di conseguenza, questa area di ricerca ha attirato l'attenzione di molti scienziati. Esistono molti studi sul comportamento e le caratteristiche di tali sistemi, ognuno dei quali sotto una diversa varietà di condizioni. Il problema può essere modellato come un processo decisionale di Markov (MDP), il che rende possibile adottare dei metodi di risoluzione che altrimenti non sarebbe possibile utilizzare.

Tra i primi abbiamo [Clark and Scarf, 1960] i quali affermano che una volta specificati i parametri del modello, la quantità ottimale da ordinare, dovrebbe, almeno in teoria, essere determinabile.

Una delle condizioni può essere la struttura della catena, ad esempio lineare come possiamo trovare in [Lambrecht et al., 1985]. Altri lavori più recenti analizzano l'applicabilità del deep reinforcement learning sulle tre differenti strutture possibili in un sistema multi-echelon, lineare, divergente o generale [Geevers et al., 2023].

Una struttura lineare (1.1 (a)) è quando ogni sottolivello viene rifornito da un solo nodo del livello superiore; divergente (1.1 (b)). quando un nodo può rifornire più livelli sottostanti; generale (1.1 (c)) quando esiste almeno un nodo viene servito da più di un nodo superiore.

Un'altra complicazione può essere aggiunta se si considerano prodotti che con il tempo deteriorano fino ad arrivare ad essere non più commerciabili. Numerosi studi hanno proposto modelli matematici per descrivere il processo di deterioramento, prendendo in considerazione variabili come il tempo, la temperatura e altri fattori ambientali. Ad

esempio, il modello di deterioramento esponenziale è comunemente utilizzato per rappresentare la diminuzione della freschezza del prodotto nel tempo [Abad, 2000]. Con la presa di coscienza da parte di una grande fetta di società di quanto sia importante il tema riguardante lo spreco alimentare e la sostenibilità ambientale, ma soprattutto grazie alle grandi catene di supermercati che si sono rese conto di quanto impattasse sui loro profitti il dover buttare via della merce non più vendibile, possiamo trovare una quantità di lavori maggiore in questo campo nell'ultimo ventennio. [Goyal and Giri, 2001] introducono una classificazione dei beni inventariati, che viene ripresa in molti lavori successivi, la quale divide i prodotti in tre meta-categorie basate su obsolescenza, deteriorazione e non-obsolescenza/deteriorazione. L'obsolescenza si riferisce ad oggetti che perdono valore per, ad esempio, il rapido cambio di tecnologie utilizzate oppure per l'ingresso sul mercato di un'azienda rivale con un prodotto migliore. Altri esempi possono essere i beni legati al mondo della moda o che hanno una stagionalità, che dopo un certo periodo devono essere sventuti oppure in quale modo scartati. Invece la deteriorazione può essere causata dalla cattiva conservazione del prodotto o dal decorrere del tempo.

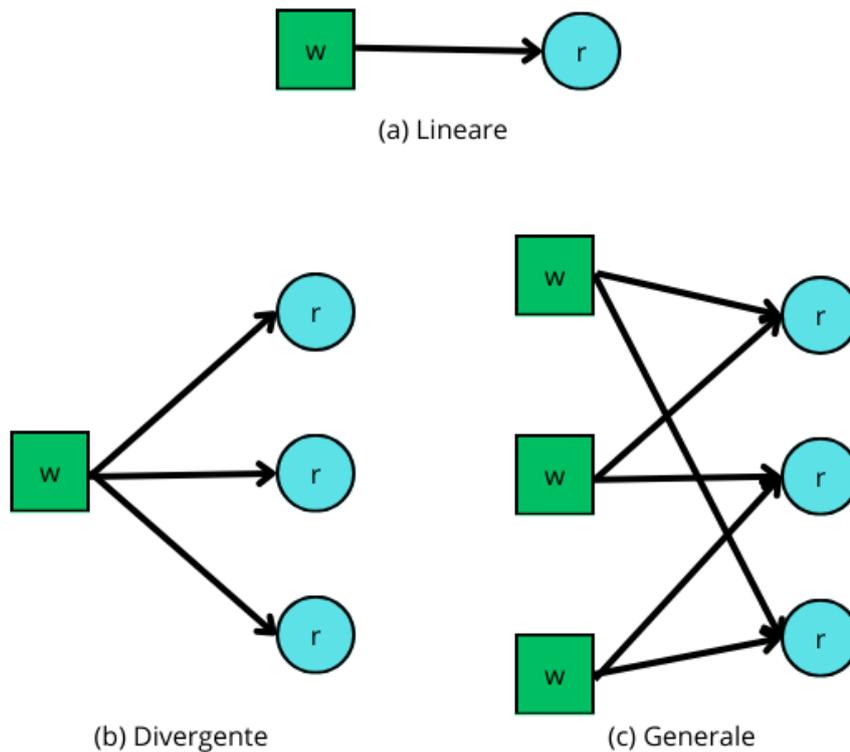


Figura 1.1: Strutture multi-echelon

Capitolo 2

Dinamica del sistema

In questo capitolo viene introdotto il problema di ottimizzazione per la gestione delle scorte di un bene deperibile in forma generale, dove esistono due tipologie di punti: magazzino $w \in W$, rivenditore $r \in R$. Il problema è modellato come un processo Markoviano (MDP), dove tutte le decisioni vengono effettuate da una sola entità centrale. Troviamo inizialmente le assunzioni fatte per la configurazione del sistema e successivamente sono illustrate le due politiche di ordinazione implementate.

2.1 Assunzioni

Il modello sviluppato è basato sulle seguenti assunzioni:

- dato che non vengono presi in considerazione prodotti che interagiscono tra loro, si prende in esame un solo prodotto;
- viene seguita una politica di revisione periodica;
- i tempi di approvvigionamento, misurati in periodi, sono deterministici e indipendenti dal numero di prodotti ordinati;
- le distribuzioni delle domande ai rivenditori sono conosciute
- non c'è backlog degli ordini né per i magazzini né per i rivenditori;
- la domanda non soddisfatta ai rivenditori è soggetta a penalità;
- non c'è limite alla capacità in nessuno dei due livelli;
- il costo di ordinazione è presente solo al livello dei magazzini;
- non sono presenti costi di trasporto;
- il costo per il mantenimento dell'inventario è costante e presente solo nel livello dei rivenditori;
- il prezzo di vendita e la qualità del prodotto rimangono gli stessi per tutti i tempi di vita rimanenti.

2.2 Stato, azioni e contributo immediato

Lo stato è descritto dall'insieme degli inventari di magazzini e rivenditori, più il numero dei prodotti in transito verso i magazzini e quelli in transito dai magazzini ai rivenditori. Le azioni sono gli ordini che sottomettono rivenditori e magazzini alla fine di ogni periodo di tempo t . Un sommario con la spiegazione delle notazioni utilizzate è visibile nella tabella 2.1.

$$s_t = \left[I_{w,t}^F, I_{r,t}^F, T_{w,i'}, T_{w,r,\bar{i}} \right] \quad \forall w \in W, \forall r \in R, \quad i' \in \{1, \dots, \lambda_w\}, \bar{i} \in \{1, \dots, \lambda_{w,r}\}$$

$$a_t = \left[O_{w,t}, O_{r,w,t} \right] \quad \forall w \in W, \forall r \in R$$

Le variabili $I_{w,t}^F, I_{r,t}^F, T_{w,i'}, T_{w,r,\bar{i}}$ rappresentano l'aggregazione dei prodotti lungo i diversi tempi di vita rimanenti $d \in \{1, \dots, SL\}$. All'inizio di ogni periodo di tempo t vengono consegnati gli ordini ed avviene l'aggiornamento degli inventari di magazzini e rivenditori

$$I_{w,t}^I = I_{w,t-1}^F + T_{w,0} \quad \forall w \in W \quad (2.1)$$

$$I_{r,t}^I = I_{r,t-1}^F + \sum_{w \in W} T_{w,r,0} \quad \forall r \in R \quad . \quad (2.2)$$

Dopodiché vengono aggiornati gli ordini che sono in transito facendoli scorrere in avanti di un periodo ed inserendo nell'ultima posizione l'ordine effettuato alla fine del periodo precedente

$$\begin{aligned} T_{w,i'} &= T_{w,i'+1} \quad i' = 1, \dots, \lambda_w - 1; \quad \forall w \in W \\ T_{w,\lambda_w} &= O_{w,t-1} \quad \forall w \in W \\ T_{w,r,\bar{i}} &= T_{w,r,\bar{i}+1} \quad \bar{i} = 1, \dots, \lambda_{w,r} - 1; \quad \forall (r, w) \in (R, W) \end{aligned}$$

Per quanto riguarda gli ordini effettuati dai rivenditori alla fine del periodo di tempo $t-1$, essi vengono soddisfatti se c'è disponibilità, quindi

$$T_{w,r,\lambda_{w,r}} = O_{r,w,t-1} \quad \forall (r, w) \in (R, W) \quad ,$$

altrimenti viene spedito un numero di prodotti proporzionale alla somma totale degli ordini pervenuti al magazzino w .

A questo punto l'inventario di ogni magazzino diventa

$$I_{w,t}^F = I_{w,t}^I - \sum_{r \in R} T_{w,r,\lambda_{w,r}} \quad .$$

L'evento successivo è il soddisfacimento, da parte dei rivenditori, della domanda pervenuta in quel periodo di tempo. Nel caso in cui la disponibilità del prodotto termini, si incombe in una penalità per ogni vendita persa. A questo punto si avrà un livello di magazzino finale che sarà

$$I_{r,t}^F = \max(0, I_{r,t}^I - d_{r,t}) \quad \forall r \in R \quad ,$$

ed un conteggio delle vendite perse pari a

$$LS_{r,t} = \max(0, d_{r,t} - I_{r,t}^I) \quad \forall r \in R \quad .$$

Arriva poi il momento in cui i prodotti scaduti vengono scartati dai magazzini e dai rivenditori. Dopodiché viene osservato lo stato del sistema così che possano essere emessi gli ordini $O_{w,t} \quad \forall w \in W$, $O_{r,w,t} \quad \forall r \in R$.

Al termine del periodo t vengono calcolati i costi sostenuti più la penalità per eventuali vendite mancate

$$c_t(s_t, a_t) = \sum_{w \in W} c \cdot O_{w,t} + \sum_{r \in R} (h_r \cdot I_{r,t}^I + q \cdot LS_{r,t})$$

ed il ricavo ottenuto

$$r_t(s_t, a_t) = \sum_{r \in R} p \cdot (d_{r,t} - LS_{r,t}) \quad .$$

Il profitto al tempo t sarà quindi calcolato come

$$\pi_t(s_t, a_t) = r_t(s_t, a_t) - c_t(s_t, a_t)$$

In figura 2.1 è illustrata la dinamica del sistema durante un periodo di tempo.

2.3 Politiche implementate

Le politiche di ordinamento sono strategie utilizzate dalle aziende per gestire in modo efficiente l'inventario e garantire che i prodotti siano disponibili quando i clienti ne fanno richiesta. Esse determinano quando e quanto riordinare merce per mantenere un adeguato livello di stock senza eccessi o carenze. Nei due livelli considerati in questa tesi le vengono adoperate le stesse politiche, ma con parametri diversi che vengono ottimizzati per ciascuno punto della catena.

2.3.1 Politica di ordinazione costante

Una delle più semplici politiche che si possono adottare è quella che prevede un'ordinazione di un numero di prodotti costante, dato che non prende in considerazione lo stato del sistema. Una questione da considerare è la periodicità di sottomissione di un ordine e si potrebbe far variare anche la quantità in base alla stagionalità della domanda:

$$X^C(S) = S \quad .$$

La periodicità di ordinazione implementata in questo lavoro di tesi è giornaliera invece la stagionalità non è stata presa in considerazione, quindi per ogni periodo di tempo t viene ordinata una quantità pari a S .

Tabella 2.1: Notazione

Indici	
w	Magazzino $w \in W$
r	Rivenditore $r \in R$
t	Periodo di tempo $t = 1, \dots, T$
Parametri	
λ_w	Tempo di consegna dal fornitore al magazzino w
$\lambda_{w,r}$	Tempo di consegna dal magazzino w al rivenditore r
h_r	Costo di inventario per i rivenditori
q	Penalità per la vendita persa
c	Costo del singolo prodotto
p	Prezzo di vendita del singolo prodotto
SL	Shelf-life, tempo di vita del prodotto
Variabili	
$I_{w,t}^I$	Livello di inventario del magazzino w all'inizio del periodo t
$I_{r,t}^I$	Livello di inventario del rivenditore r all'inizio del periodo t
$I_{w,t}^F$	Livello di inventario del magazzino w alla fine del periodo t
$I_{r,t}^F$	Livello di inventario del rivenditore r alla fine del periodo t
$T_{w,i'}$	Numero di prodotti in transito verso il magazzino in arrivo tra i' periodi di tempo
$T_{w,r,\bar{i}}$	Numero di prodotti in transito verso il rivenditore r dal magazzino w in arrivo tra \bar{i} periodi di tempo
$O_{w,t}$	Numero di prodotti ordinati dal magazzino w al tempo t
$O_{r,w,t}$	Numero di prodotti ordinati dal rivenditore r al magazzino w
$LS_{r,t}$	Numero di vendite perse dal rivenditore r al tempo t
$d_{r,t}$	Domanda pervenuta al rivenditore r al tempo t

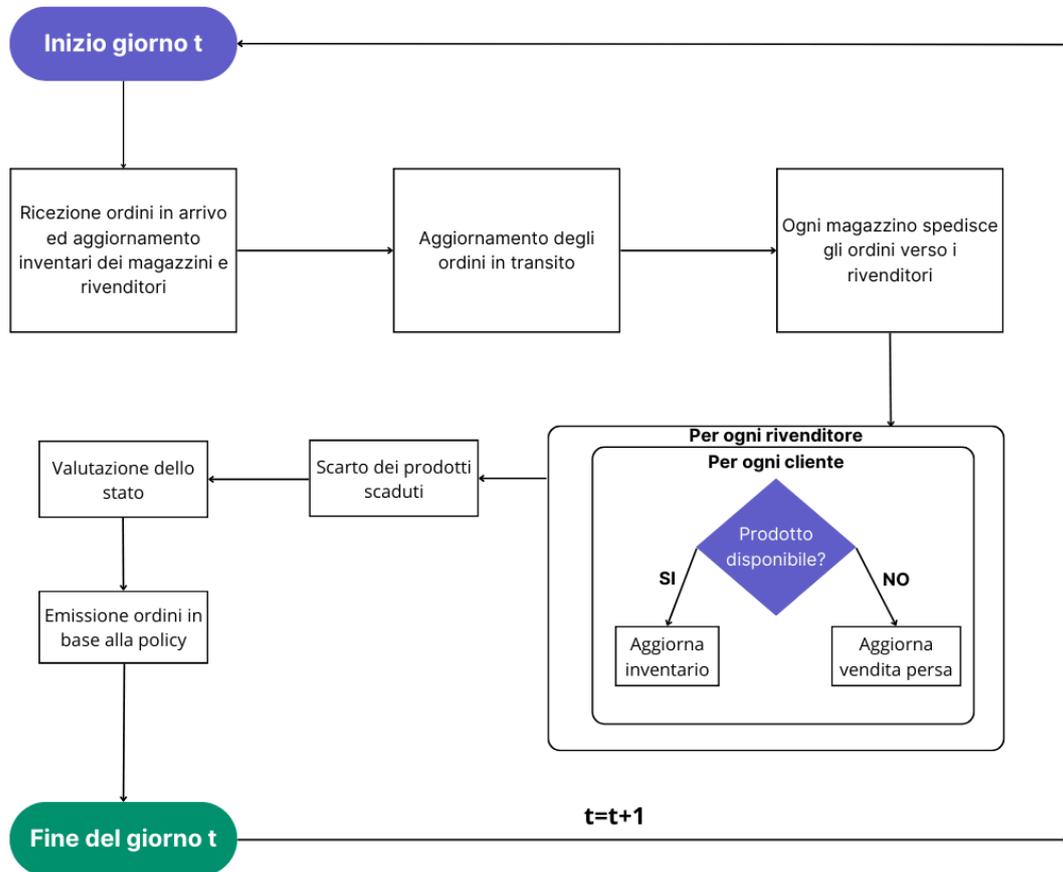
2.3.2 Politica di ordinazione base-stock

Una delle classiche euristiche utilizzate per questa tipologia di problemi è la politica di ordinazione base-stock [Gioia et al., 2023]. Qui la quantità da ordinare alla fine di ogni periodo di tempo t dipende dallo stato dell'inventario che si sta osservando, che comprende anche gli ordini in transito verso il nodo che stiamo considerando, e da un parametro S , il quale indica il livello di magazzino minimo desiderato:

$$X^B(s_t|S_w) = \max \left(0, S_w - \left(\sum_{i=1}^{\lambda_w} T_{w,i} + I_w^F \right) \right) .$$

Qui è stato preso d'esempio il magazzino w , ma lo stesso ragionamento viene applicato anche per i rivenditori.

Figura 2.1: Evoluzione giornaliera



2.3.3 Obiettivo

Dopo aver definito le politiche che verranno utilizzate resta solo capire qual è l'obiettivo da raggiungere. Dato che le policy dipendono dal parametro S , puntiamo a massimizzare il profitto giornaliero medio per un determinato orizzonte di tempo fissato, dove andiamo a simulare la politica di rifornimento, qualunque essa sia:

$$\max_S \mathbb{E} \left\{ \frac{1}{T} \sum_{t=1}^T \pi_t(s_t, X^B(s_t|S)) \right\}$$

oppure

$$\max_S \mathbb{E} \left\{ \frac{1}{T} \sum_{t=1}^T \pi_t(s_t, X^C(S)) \right\}.$$

In questo caso, a differenza della classica formulazione che troviamo nella programmazione dinamica, non c'è il fattore di sconto siccome viene considerato il profitto medio.

Capitolo 3

Implementazione del simulatore

L'implementazione di un simulatore per la gestione delle scorte multi-echelon di prodotti deperibili con deterioramento deterministico è essenziale per sviluppare e ottimizzare strategie di gestione delle scorte che minimizzino le perdite dovute al deterioramento del prodotto, assicurando al contempo una disponibilità ottimale del prodotto lungo la catena di distribuzione. Questo approccio fornisce una base solida per prendere decisioni informate e migliorare l'efficienza della gestione delle scorte nel settore dei prodotti deperibili. Ho costruito, avvalendomi del software MATLAB, un framework in grado di riprodurre le azioni che si susseguono all'interno del sistema giornalmente, modellando l'entità coinvolte attraverso la programmazione ad oggetti. Anche se il problema da me trattato contiene un solo magazzino con due rivenditori, il simulatore è stato implementato con la possibilità di essere esteso a casi più complessi che mantengano però un tipo di struttura al massimo divergente.

Magazzino

L'oggetto di tipo **Magazzino** rappresenta il secondo livello all'interno della catena di approvvigionamento, dato che è quello dal quale partiranno i rifornimenti al livello uno (quello che deve soddisfare la domanda della clientela). Al suo interno troviamo le variabili:

- *inventario*: vettore contenente la quantità di prodotti suddivisi in giorni di vita rimanenti;
- *inTransito*: vettore con gli ordini in arrivo verso il magazzino a distanze di tempo differenti (non vi è distinzione per la durata del prodotto dato che arrivano al magazzino solo prodotti con il massimo della freschezza);
- *ordini*: matrice contenente gli ordini ricevuti al termine del periodo di tempo precedente con associati i tempi di consegna ai rivenditori;
- *leadTime*: scalare che indica i giorni necessari ad un ordine per arrivare al magazzino.

I metodi che invece permettono al magazzino di poter espletare le proprie funzioni sono i seguenti:

- *aggVita()* è adibito all'aggiornamento della vita dei prodotti all'interno dell'*inventario* e restituisce il numero di quelli che sono arrivati al termine della loro fruibilità;
- *emettiOrdine(ordine)* è il metodo tramite il quale il magazzino sottomette l'ordine alla fine del periodo t , il quale arriverà dopo *leadTime* istanti di tempo;
- *riceviOrdini(ordini)* prende in carico il nuovo ordine effettuato dai rivenditori;
- *evadiOrdini()* ad ogni istante di tempo t spedisce gli ordini ricevuti al termine del periodo $t - 1$ in base alla disponibilità presente in quel momento nell'inventario: se ci sono abbastanza prodotti per tutti gli ordini ricevuti, essi vengono estratti in modalità FIFO e spediti ai rivenditori, altrimenti gli ordini vengono ridimensionati in proporzionalmente al totale degli ordini ricevuti.

Rivenditore

A rappresentare il primo livello c'è l'oggetto di tipo **Rivenditore** che è descritto da:

- *inventario*: vettore contenente la quantità di prodotti suddivisi per giorni di vita rimanenti;
- *costoMagazzino*: scalare che indica le spese associate allo stoccaggio e alla gestione delle scorte;
- *leadTime*: scalare che denota i periodi di tempo necessari per l'arrivo degli ordini dal magazzino.

Invece le funzioni all'interno dell'oggetto sono:

- *aggInv(inEntrata)* permette al rivenditore di aggiungere al proprio inventario l'ordine in ingresso;
- *aggVita()* come nell'oggetto **Magazzino**, aggiorna la vita dei prodotti sugli scaffali e restituisce la quantità di quelli scartati;
- *vendita(flag, domanda)* è il metodo che svolge l'azione principale di un rivenditore, soddisfare la domanda. Riceve in input un *flag* che indica la modalità di acquisto dei clienti (FIFO o LIFO) e la domanda pervenuta in quel periodo di tempo. Esso restituisce due valori *venduto* e *perso*, che rappresentano rispettivamente la domanda soddisfatta e il numero di vendite mancate.

Trasporto

Per poter tenere traccia degli ordini in transito dal magazzino al rivenditore ho deciso di creare l'oggetto di tipo **Trasporto**. In esso c'è una variabile:

- *nastro*: una matrice che contiene un numero di vettori pari al *leadTime* del **Rivenditore** per il quale è stato istanziato l'oggetto **Trasporto**, ed ognuno di essi ha la stessa funzione delle variabili *inventario* presenti negli oggetti **Magazzino** e **Rivenditore**.

Per quanto riguarda i metodi utili alla gestione del trasporto abbiamo:

- *carico(nuovoOrdine)* shifta di una cella in avanti i vettori colonna (diminuisce di un giorno il numero di periodi mancanti per la consegna di ogni ordine) ed aggiunge in ultima posizione l'ultimo stock di prodotti spedito dal magazzino verso il rivenditore;
- *aggVita* svolge lo stesso compito dell'omonimo metodo citato nei due oggetti precedenti, ma su tutta la matrice *nastro*;
- *consegna* fornisce l'ordine a distanza minore dal rivenditore, cioè quello che viene consegnato all'inizio di ogni periodo di tempo *t*.

ScenarioRandom

Per simulare la domanda ai vari rivenditori ho creato un oggetto che prende in input vari parametri come il valore atteso e la deviazione standard della domanda, ed un vettore contenente gli indici di stagionalità per i diversi giorni della settimana. In base al giorno della settimana, avremo che il numero di clienti atteso dipenderà dal valore atteso della distribuzione, μ , e dall'indice corrispondente, v_k , nel seguente modo

$$\mu_k = v_k \mu \quad k \in \{0, 1, \dots, 6\} \quad .$$

Questo oggetto è in grado di generare, ogni volta che viene chiamata la sua funzione *creaScenario(orizzonteTemp)*, un campione di domanda, dalla distribuzione definita al suo interno, per un numero di periodi pari a *orizzonteTemp*.

Simulazione

Dopo aver definito tutti i protagonisti presenti nel nostro sistema, ho creato un programma che simula tutto ciò che accade nella dinamica illustrata all'interno del capitolo 2.

Al suo interno si possono modificare i parametri che riguardano il prodotto (costo, prezzo di vendita, penalità per la vendita mancata e la sua shelf-life) e quelli riguardanti la simulazione (flag per la modalità di acquisto, lead time dei rivenditori, flag per la policy di riordino e orizzonte temporale). Dopo aver deciso la configurazione si passa all'inizializzazione degli inventari di magazzino e rivenditori, insieme alla creazione dei "nastri di trasporto".

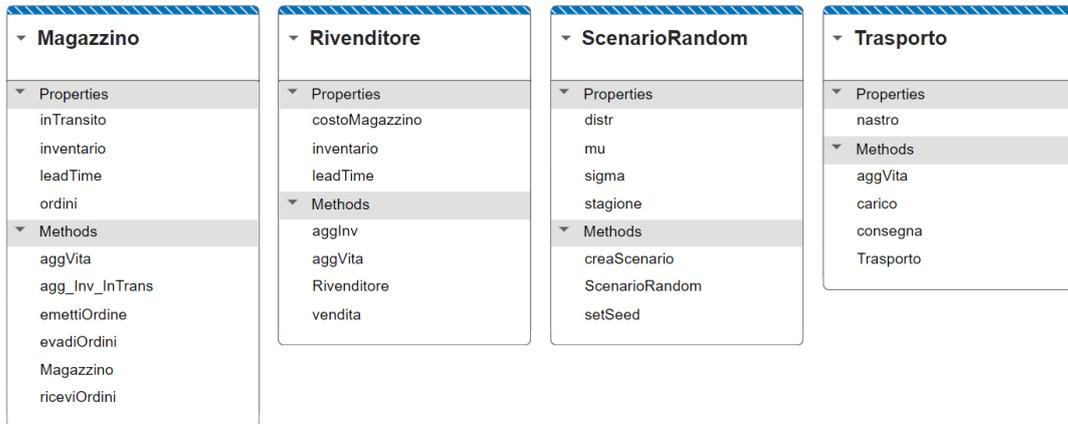


Figura 3.1: UML

Oltre a queste variabili c'è bisogno di indici essenziali per la valutazione della performance delle policy:

- *ordinato*: somma di tutti gli ordini emessi dal magazzino;
- *vendite*: matrice che tiene traccia di tutte le vendite effettuate in ogni periodo di tempo da ogni rivenditore;
- *holding*: matrice in cui viene salvato l'inventario all'inizio di ogni periodo di tempo per ogni rivenditore;
- *lostSale*: matrice contenente il numero di vendite perse in ogni periodo di tempo da ogni rivenditore;
- *scartoW* e *scarto_r*: tengono traccia del numero dei prodotti scartati alla fine di ogni intervallo di tempo dal magazzino e dai rivenditori.

Dopo tutta la parte di preparazione, inizia la simulazione che va avanti per ogni periodo, dal giorno 1 fino alla fine dell'orizzonte temporale, ed al termine di essa avviene il calcolo del profitto totale che sarà la misura di performance.

Il codice è disponibile al link : <https://github.com/LucioMoro/Tesi.git>

3.1 Ottimizzazione

L'ottimizzazione dei parametri appartenenti alle policy scelte, è stata fatta utilizzando la funzione MATLAB *particleswarm()*, facente parte del toolbox di ottimizzazione globale, basata sull'euristica "Particle Swarm Optimization". La suddetta è un'euristica di ottimizzazione basata sul comportamento sociale degli esseri viventi, come all'interno degli sciame. Le applicazioni di questo algoritmo sono svariate e molti modelli ibridi sono stati proposti [Gad, 2022].

Una soluzione candidata è rappresentata come una particella nello spazio delle soluzioni e l'algoritmo prova in modo iterativo a migliorarla, basandosi su una determinata misura di qualità. Queste particelle si muovono nello spazio in base a due componenti principali che sono "velocità" e "posizione". La velocità è influenzata dalla sua esperienza personale e dall'esperienza del vicinato quindi, quando una soluzione migliore viene trovata, lo sciame viene fatto muovere nella direzione della best solution. L'algoritmo termina in due casi: quando il numero massimo di iterazioni viene raggiunto oppure quando converge ad una soluzione. Utilizzare questo metodo ha un vantaggio molto grande, cioè la necessità di poche, o nulle, assunzioni sul problema, ma non garantisce che una soluzione ottima venga mai trovata.

Per poter utilizzare la funzione *particleswarm()* ho incapsulato il simulatore all'interno di una funzione che prende in ingresso i parametri delle policy e restituisce l'opposto della misura di performance, quindi girata come una funzione di costo, dato che l'obiettivo di *particleswarm()* è minimizzare. L'altro dato minimo richiesto dalla funzione è il numero di variabili da ottimizzare, ed in aggiunta ho sfruttato la possibilità di inserire un limite inferiore, dato che il livello di magazzino non può essere negativo, ed un limite superiore, per ridurre lo spazio da esplorare, per la soluzione.

Capitolo 4

Esperimenti computazionali e analisi dei risultati

Passiamo ora alla descrizione della fase di sperimentazione. Come già detto il sistema preso in considerazione si compone di un magazzino che ha il compito di rifornire due rivenditori differenti. C'è un solo una tipologia di prodotto che si muove all'interno della catena il quale ha una vita di scaffale costante misurata in periodi (o giorni). Non esiste backlog né al magazzino né ai rivenditori. Il costo di magazzino è impostato fisso ad 1 per entrambi i rivenditori mentre nel magazzino viene trascurata la spesa di gestione. Non vi sono spese di trasporto della merce ma viene solo sostenuto un costo di acquisto del prodotto a livello del magazzino. La vita del prodotto inizia ad essere calcolata dal momento in cui arriva al centro di smistamento.

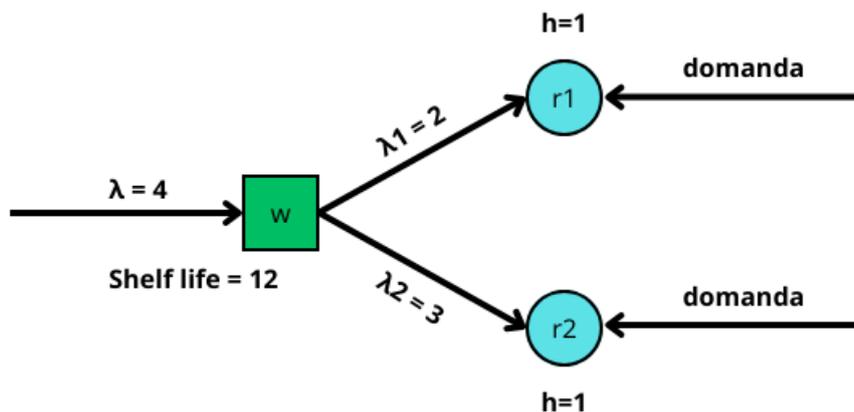


Figura 4.1: Sistema

L'orizzonte temporale su cui ho concentrato la sperimentazione è un anno (52 settimane). Il comportamento dei clienti è uguale in entrambi i rivenditori, quindi o entrambi LIFO o entrambi FIFO.

Le due distribuzioni da cui ho generato gli scenari di domanda sono illustrate nella tabella 4.1 e sono state prese da [Gioia et al., 2023].

Rivenditore	Distribuzione	μ	σ	Stagionalità
1	Normale	100	15	[90, 100, 100, 100, 130, 200, 200]
2	Normale	200	20	[90, 100, 100, 100, 130, 200, 200]

Tabella 4.1: Distribuzioni

Ho deciso di studiare varie configurazioni del problema andando a costruire un piano sperimentale riportato nella tabella 4.2. I fattori presi in considerazione sono il margine di guadagno su ogni prodotto, la penalità per ogni vendita mancata, i lead time ai due rivenditori e la modalità di acquisto dei clienti.

N° Problema	Shelf life	Margine	Penalità	λ_1	λ_2	Acquisto
1	12	7	2	2	3	FIFO
2	12	7	2	2	3	LIFO
3	12	7	2	5	6	FIFO
4	12	7	2	5	6	LIFO
5	12	7	8	2	3	FIFO
6	12	7	8	2	3	LIFO
7	12	7	8	5	6	FIFO
8	12	7	8	5	6	LIFO
9	12	3	2	2	3	FIFO
10	12	3	2	2	3	LIFO
11	12	3	2	5	6	FIFO
12	12	3	2	5	6	LIFO
13	12	3	8	2	3	FIFO
14	12	3	8	2	3	LIFO
15	12	3	8	5	6	FIFO
16	12	3	8	5	6	LIFO
17	12	7	2	8	9	FIFO
18	12	7	2	8	9	LIFO
19	12	7	8	8	9	FIFO
20	12	7	8	8	9	LIFO
21	12	3	2	8	9	FIFO
22	12	3	2	8	9	LIFO
23	12	3	8	8	9	FIFO
24	12	3	8	8	9	LIFO

Tabella 4.2: Configurazioni

L'ottimizzazione è stata ripetuta, per entrambe le policy, su tre scenari generati dalle distribuzioni in 4.1, ed i risultati ottenuti, con el soluzioni che li hanno generati, sono riportati in 4.3, 4.4, 4.5, 4.6, 4.7 e 4.8.

Tra i risultati ottenuti c'è da segnalare il fatto che in nessuna simulazione sono presenti scarti all'interno del magazzino, quindi le soluzioni trovate tendono a far rimanere le scorte il meno possibile all'interno del magazzino centrale.

Prendendo in considerazione i problemi divisi per margine di profitto, possiamo vedere come il caso FIFO sia sempre il più favorevole in termini di profitto, inoltre, dalle figure 4.2b e 4.2d, vediamo che quando il margine è basso e lead time e penalità sono alti, la soluzione migliore ci porta comunque ad un profitto negativo (perdite).

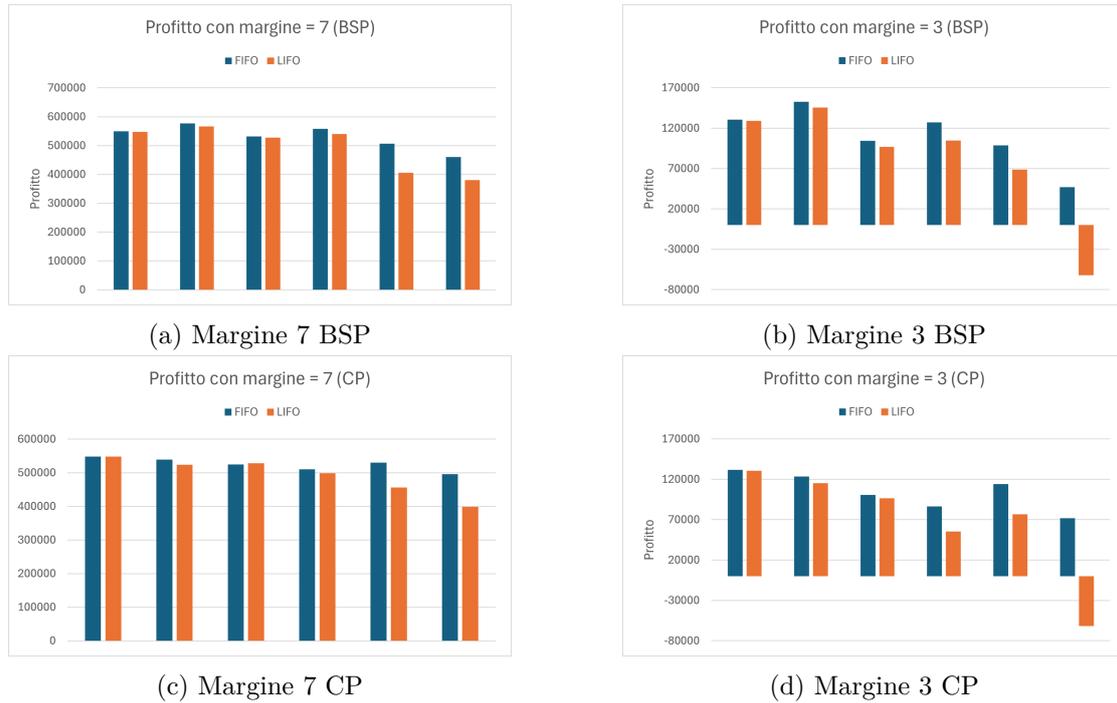
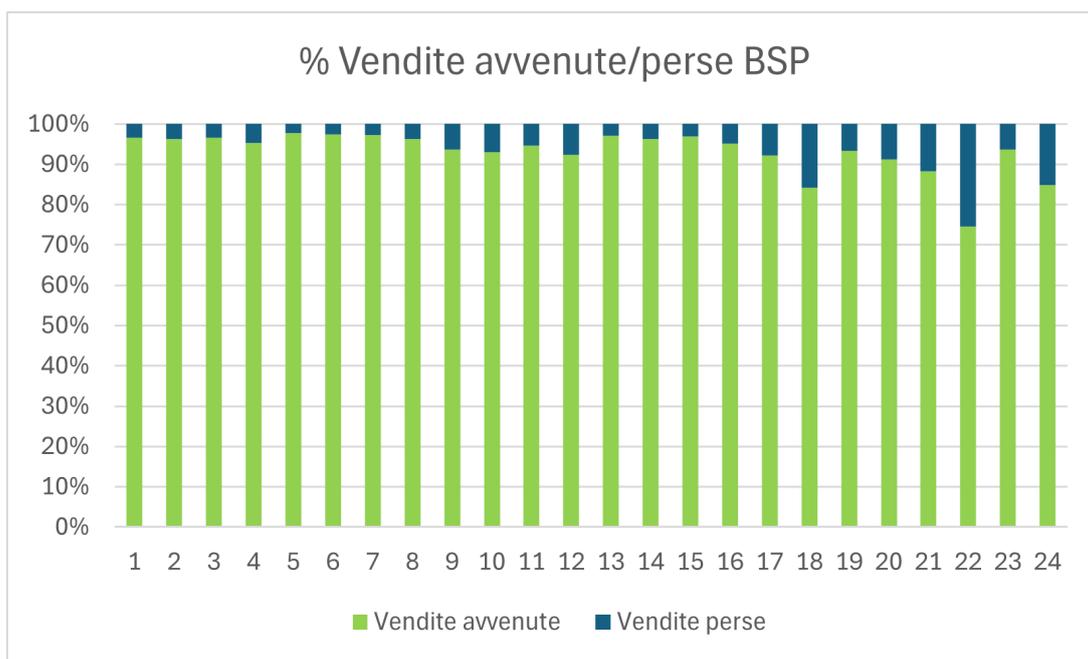


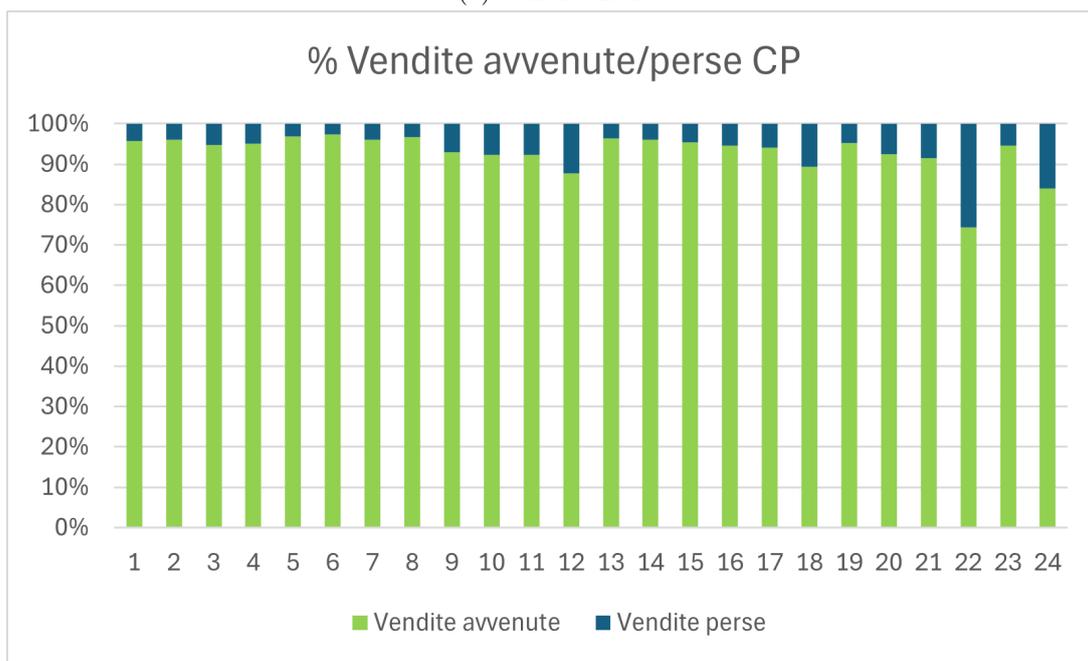
Figura 4.2: Profitti divisi per margini

Andando ad analizzare il volume delle vendite con le due diverse policy, ho visto che per la maggior parte dei problemi la percentuale di vendite perse resta al di sotto del 10% sia con la BSP che con la CP. Policy che si comportano allo stesso modo per le configurazioni 18, 22 e 24, dove, però, entrambe ci portano a superare la soglia predetta. Nello specifico, quando la penalità è bassa, il lead time è alto e i prodotti vengono acquistati in modalità LIFO, il sistema tende a preferire lo sconto di qualche cliente piuttosto che dover scartare prodotti scaduti, registrando un 25% di vendite perse (figura 4.3).

Un aspetto interessante da notare nei problemi risolti con la policy costante, è che la quantità ottimale di ordinazione per il magazzino è mediamente 289 ± 18 , mentre i valori di S per i due rivenditori hanno sempre un rapporto all'incirca 1 : 2: sicuramente questo deriva dalla modalità con cui il magazzino evade gli ordini quando non ci sono abbastanza



(a) Vendite BSP

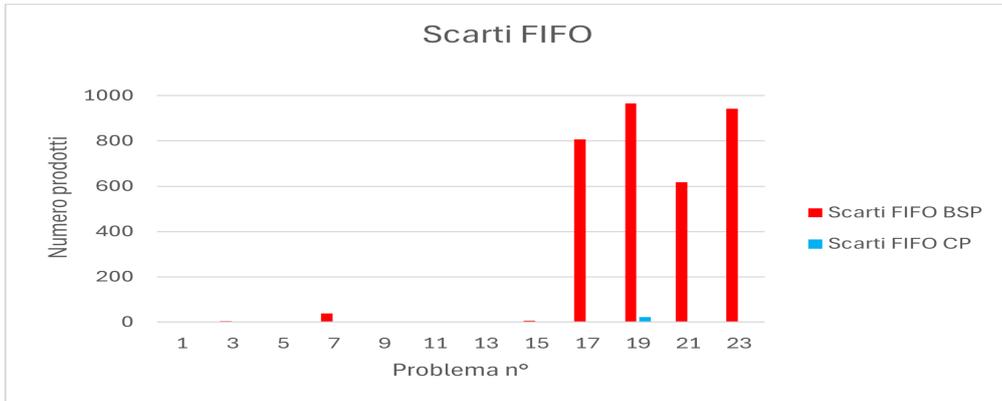


(b) Vendite CP

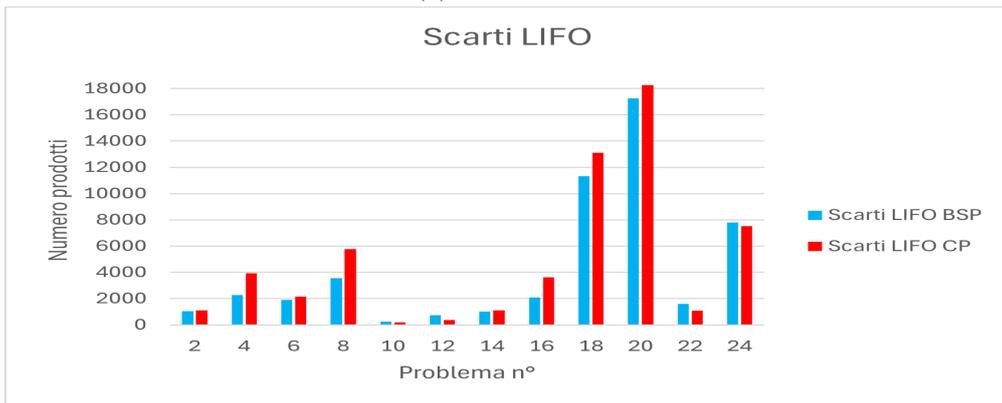
Figura 4.3: Vendite

prodotti per soddisfare interamente le richieste dei negozi. Prendendo invece la soluzione ottimale della politica base-stock troviamo che il valore medio tra tutti i problemi è pari a 1245 ± 139 , ma non c'è un pattern specifico tra le soluzioni trovate per i rivenditori. Possiamo quindi vedere come le soluzioni trovate per la policy costante sono meno variabili di quelle trovate per la base-stock policy.

Un altro risultato significativo si osserva tra i prodotti scartati. Quando la modalità di acquisto dei clienti è all'estremo più sfavorevole, LIFO (figura 4.4b), si trova sempre una quantità positiva di merce dismessa, che cresce molto quando la penalità associata alle vendite perse è alta; ciò è dato dal fatto che il sistema cerca di avere sempre disponibilità all'interno dei rivenditori trascurando la possibilità di rimanere con esuberi importanti. Nei problemi in cui i clienti scelgono di prelevare la merce dagli scaffali con preferenza verso il prodotto con una vita rimanente minore (figura 4.4a), se la configurazione è favorevole (lead time medio-basso), non vi è un numero elevato di prodotti scartati in alcuno dei due rivenditori, per la BSP. Invece questa caratteristica è preservata in tutte e 12 le configurazioni del sistema quando utilizziamo i valori ottimali, trovati tramite *particleswarm*, della CP.



(a) Scarti FIFO



(b) Scarti LIFO

Figura 4.4: Scarti FIFO LIFO

Confrontando invece le due politiche problema per problema sulla base del profitto, ho riscontrato delle performance simili per poco più della metà dei problemi. I casi in cui invece una delle due policy ha reso di più sono i problemi 11, 12, 15, 16 e 23. Nei primi due la BSP ha reso circa il 24% e 26% in più rispetto alla CP; prestazioni migliori anche nei due successivi dove rispettivamente si vedono un profitto maggiore del 47% e del 90%. Invece nel problema 26 il trend si inverte, con la policy costante che frutta il 53% in più della base-stock.

BSP	Ordini	Vendite	Lost	Scarti R	Profitto	S		
						W	R1	R2
1	105671	104667	4041	0	547245	1186	396	938
2	106744	104620	4088	1116	546135	1202	388	938
3	107211	105167	3541	0	576634	1424	645	1414
4	107934	103615	5093	2325	566709	1399	639	1404
5	107344	106240	2468	0	528557	1210	422	977
6	108694	105703	3005	1907	525155	1229	409	955
7	108469	106170	2538	84	558539	1549	667	1438
8	110448	104691	4017	3663	540369	1450	665	1436
9	102120	101203	7505	0	129232	1132	356	902
10	102093	100936	7772	280	128330	1133	346	903
11	104697	102773	5935	0	152738	1388	609	1366
12	102820	100212	8496	766	144526	1331	582	1341
13	106458	105410	3298	0	101075	1200	409	948
14	106589	104524	4184	1060	94646	1200	388	935
15	107503	105441	3267	0	126701	1430	652	1419
16	107502	103397	5311	2126	104793	1391	634	1399
17	104207	100559	8149	966	506535	1150	966	2104
18	101738	85753	22955	13600	330536	1118	1002	2048
19	106700	102587	6121	1318	466083	1180	996	2195
20	121232	99556	9152	18375	379907	1345	1118	2326
21	99198	95985	12723	750	96105	1092	841	2032
22	83677	80247	28461	1509	66851	920	725	1561
23	105440	101659	7049	1038	42089	1166	987	2135
24	103719	92886	15822	8243	-62059	1140	980	1950

Tabella 4.3: BSP Scenario 1

BSP	Ordini	Vendite	Lost	Scarti R	Profitto	S		
						W	R1	R2
1	106463	105444	3410	0	553526	1190	392	960
2	107166	105023	3831	1100	549827	1198	383	957
3	107388	105290	3564	12	580213	1426	640	1415
4	108483	104109	4745	2298	569451	1405	633	1423
5	107678	106615	2239	0	537291	1212	412	980
6	109820	106560	2294	2138	532664	1234	412	979
7	107805	105664	3190	30	560052	1454	658	1413
8	110312	105081	3773	3117	544296	1397	650	1467
9	103585	102643	6211	0	131874	1150	353	931
10	102477	101290	7564	283	129453	1137	338	916
11	105391	103369	5485	0	153538	1389	602	1390
12	104005	101123	7731	942	143578	1351	603	1346
13	107000	105964	2890	0	107676	1198	402	968
14	107228	105070	3784	1114	98190	1199	385	956
15	107706	105580	3274	20	129568	1445	648	1418
16	108226	104023	4831	2134	107534	1391	623	1431
17	102013	98811	10043	653	502125	1123	984	2000
18	107760	94560	14294	10468	448391	1192	1017	1999
19	102566	99326	9528	662	443593	1132	1002	2000
20	118675	98931	9923	16602	382232	1314	1078	2289
21	98989	95970	12884	580	99661	1089	868	1984
22	84721	81420	27434	1358	71605	931	755	1559
23	106056	102373	6481	977	51218	1167	979	2183
24	101305	91384	17470	7396	-65623	1120	977	1832

Tabella 4.4: BSP Scenario 2

BSP	Ordini	Vendite	Lost	Scarti R	Profitto	S		
						W	R1	R2
1	105934	104817	3566	0	549444	1180	389	960
2	106343	104337	4046	904	547591	1181	387	950
3	106728	104616	3767	0	574577	1446	634	1410
4	107244	103011	5372	2197	563336	1385	635	1395
5	106812	105650	2733	0	531093	1197	403	970
6	108189	105355	3028	1667	526288	1210	401	964
7	107419	105266	3117	0	554305	1456	651	1424
8	110342	104304	4079	3867	535440	1448	662	1440
9	102408	101412	6971	0	130378	1137	343	916
10	102078	100933	7450	159	129552	1131	340	911
11	104574	102599	5784	0	151435	1376	612	1365
12	101793	99811	8932	505	148318	1311	573	1332
13	106296	105165	3218	0	104700	1186	394	965
14	106114	104193	4190	823	97997	1179	385	946
15	107052	104922	3461	0	125411	1451	642	1415
16	106846	102797	5586	2030	102219	1378	630	1391
17	104688	101146	7237	801	509955	1159	951	2125
18	106672	94128	14255	9874	450847	1189	992	1978
19	106174	102422	5961	915	470803	1181	983	2160
20	118477	98594	9789	16806	380845	1320	1093	2260
21	98569	95610	12773	526	99876	1084	875	1991
22	85358	81492	26891	1889	67970	942	719	1597
23	105005	101428	6955	814	47882	1164	958	2130
24	102358	92053	16330	7776	-58833	1140	973	1859

Tabella 4.5: BSP Scenario 3

CP	Ordini	Vendite	Lost	Scarti R	Profitto	S		
						W	R1	R2
1	104468	103533	5175	0	545185	287	821	1631
2	106652	104503	4205	1160	547332	293	810	1617
3	104468	102672	6036	0	536673	287	1002	2000
4	109200	103226	5482	4059	522432	300	983	2000
5	105924	104955	3753	0	519746	291	909	1814
6	109200	105900	2808	2225	526560	300	598	1160
7	105924	104082	4626	0	505929	291	684	1365
8	113204	105234	3474	5930	498344	311	335	683
9	101192	100338	8370	0	131086	278	526	1061
10	101556	100474	8234	221	129814	279	972	1939
11	101192	99504	9204	0	122727	278	986	1994
12	98280	96215	12493	458	115523	270	942	1800
13	105196	104247	4461	0	96701	289	857	1710
14	106652	104503	4205	1160	95494	293	968	1922
15	105196	103380	5328	0	82957	289	1002	1999
16	109200	103226	5482	4059	52740	300	837	1697
17	104468	101811	6897	0	527345	287	1008	1999
18	112840	97080	11628	12832	456578	310	976	1996
19	106288	103487	5221	69	491565	292	737	1478
20	122304	100544	8164	18504	397699	336	977	1941
21	101192	98670	10038	0	113651	278	915	1845
22	83720	80800	27908	1008	76732	230	1036	2000
23	104832	102162	6546	0	68585	288	922	1842
24	99736	90520	18188	6755	-62574	274	1159	1998

Tabella 4.6: CP Scenario 1

CP	Ordini	Vendite	Lost	Scarti R	Profitto	S		
						W	R1	R2
1	105560	104626	4228	0	551822	290	1221	2465
2	107380	105177	3677	1235	550859	295	691	1388
3	105560	103756	5098	0	543189	290	993	1998
4	111020	104563	4291	4506	526627	305	668	1381
5	107016	106054	2800	0	531872	294	739	1491
6	109200	106200	2654	1978	532636	300	773	1543
7	107016	105172	3682	0	517975	294	805	1624
8	113204	105564	3290	5631	503854	311	907	1875
9	103012	102123	6731	0	132720	283	819	1699
10	101192	100192	8662	145	131144	278	979	2000
11	103012	101274	7580	0	124117	283	769	1595
12	97552	95634	13220	329	116507	268	846	1655
13	106652	105698	3156	0	104954	293	961	1944
14	107380	105177	3677	1235	99277	295	680	1364
15	106288	104465	4389	0	91068	292	872	1756
16	110656	104355	4499	4360	56972	304	150	311
17	105560	102886	5968	0	533845	290	949	1913
18	113932	97514	11340	13449	456594	313	665	1424
19	107016	104290	4564	0	503222	294	432	872
20	122668	100801	8053	18600	399028	337	931	1830
21	102648	100073	8781	0	114960	282	724	1519
22	83720	80878	27976	929	77824	230	909	1804
23	106288	103588	5266	0	76069	292	910	1847
24	106652	94036	14818	9903	-62123	293	976	1999

Tabella 4.7: CP Scenario 2

CP	Ordini	Vendite	Lost	Scarti R	Profitto	S		
						W	R1	R2
1	104832	103815	4568	0	546309	288	865	1743
2	105560	103602	4781	937	545830	290	859	1732
3	104468	102610	5773	0	537504	287	899	1809
4	107016	101919	6464	3178	521942	294	566	1133
5	105560	104529	3854	0	521750	290	984	1976
6	108836	105465	2918	2267	524655	299	784	1581
7	105560	103659	4724	0	507654	290	881	1774
8	112476	104628	3755	5764	494939	309	962	1985
9	101556	100636	7747	0	131793	279	475	965
10	101192	100102	8281	189	130316	278	959	1998
11	101920	100153	8230	0	123182	280	965	1971
12	96096	94171	14212	336	113922	264	1074	2000
13	105196	104172	4211	0	99962	289	570	1144
14	105560	103602	4781	937	94904	290	906	1826
15	105196	103305	5078	0	85899	289	940	1889
16	105196	100906	7477	2426	56061	289	989	1965
17	104468	101749	6634	0	528178	287	364	732
18	112840	96859	11524	13045	454918	310	915	1889
19	105560	102789	5594	0	493025	290	712	1429
20	120848	99918	8465	17696	397607	332	960	2000
21	101920	99313	9070	0	114045	280	725	1478
22	84084	80878	27505	1281	75103	231	1008	1974
23	105196	102438	5945	0	71318	289	918	1841
24	97552	89291	19092	5871	-60970	268	1116	1985

Tabella 4.8: CP Scenario 3

Capitolo 5

Conclusioni

L'obiettivo di questa tesi era quello di indagare il comportamento di un sistema multi-echelon in cui il prodotto inventariato è deperibile, andandone a simulare la dinamica per poi poter ottimizzare i parametri delle due policy di riordino: base-stock e costante. Il lavoro si è basato, nonostante la sua semplicità, su una serie di fattori impattanti sulla gestione delle scorte come i tempi di consegna e la deteriorazione del prodotto.

Ho potuto verificare, come mi aspettavo, che se si ha una clientela indifferente a certe tematiche come lo spreco alimentare o la salvaguardia dell'ambiente, c'è maggiore possibilità di dover scartare un gran numero di prodotti.

In termini di soluzioni trovate per le due policy, ho notato una maggiore flessibilità da parte di quelle della base-stock, che dipende dallo stato osservato, rispetto a quelle venute fuori per la politica di ordinazione costante.

In conclusione, lo studio del problema della gestione delle scorte di beni deperibili in un sistema multi-echelon ha evidenziato sfide significative e opportunità di miglioramento nell'ottimizzazione della catena di approvvigionamento. Tuttavia, rimangono ancora aree di ricerca aperte e complessità da affrontare, soprattutto nell'adattamento di queste strategie ai contesti specifici delle aziende e nel considerare le dinamiche in continua evoluzione del mercato.

Bibliografia

- Prakash L Abad. Optimal lot size for a perishable good under conditions of finite production and partial backordering and lost sale. *Computers & Industrial Engineering*, 38(4):457–465, 2000.
- Andrew J Clark and Herbert Scarf. Optimal policies for a multi-echelon inventory problem. *Management science*, 6(4):475–490, 1960.
- Ahmed G Gad. Particle swarm optimization algorithm and its applications: a systematic review. *Archives of computational methods in engineering*, 29(5):2531–2561, 2022.
- Kevin Geevers, Lotte van Hezewijk, and Martijn RK Mes. Multi-echelon inventory optimization using deep reinforcement learning. *Central European Journal of Operations Research*, pages 1–31, 2023.
- Daniele Giovanni Gioia, Leonardo Kanashiro Felizardo, and Paolo Brandimarte. Simulation-based inventory management of perishable products via linear discrete choice models. *Computers & Operations Research*, 157:106270, 2023.
- Suresh Kumar Goyal and Bibhas Chandra Giri. Recent trends in modeling of deteriorating inventory. *European Journal of operational research*, 134(1):1–16, 2001.
- Marc R Lambrecht, R Luyten, and J Vander Eecken. Protective inventories and bottlenecks in production systems. *European journal of operational research*, 22(3):319–328, 1985.