

SCUOLA DI INGEGNERIA E ARCHITETTURA

Dipartimento di Ingegneria Industriale
Corso di Laurea Magistrale in Ingegneria Gestionale

Tesi di Laurea Magistrale

in

Logistica Industriale M

**Analisi e ottimizzazione di un Container Management System nel settore automotive:
il caso Robert Bosch GmbH**

Candidato:

Davide Nasci

Relatore:

Prof. Ing. Alberto Regattieri

Correlatore:

Ing. Fabio Fanelli

INDICE

PREMESSA	3
PARTE TEORICA	5
CAPITOLO 1 - OBIETTIVI DEL PROGETTO E METODO UTILIZZATO	5
1.1 - Introduzione	5
1.2 - Motivi e obiettivi dello studio	5
Oggetto dello studio	5
Motivi.....	5
Risultati attesi.....	6
1.3 - Timing del progetto ed organizzazione del lavoro	6
Fase n°1	6
Fase n°2.....	7
Fase n°3.....	7
1.4 - I Returnable Packaging (RP)	7
Aspetti generali sui RP.....	7
I vantaggi e le criticità gestionali dei RP	8
I livelli di imballaggio.....	9
1.5 - Il metodo: le possibili alternative	10
Supply chain mapping.....	10
Supply chain analysis.....	11
1.6 - Il metodo utilizzato.....	11
Metodo degli indici entropici.....	11
Product Flow Number (PFN).....	17
PARTE PRATICA	19
CAPITOLO 2 - IL CONTESTO IN CUI SI SVILUPPA IL PROGETTO.....	19
2.1 - Introduzione	19
2.2 - Il settore di interesse: l'automotive	19
Il settore dell'automotive	19
La supply chain dell'automotive.....	20
2.3 - Il gruppo Robert Bosch GmbH	21
Le origini del gruppo Robert Bosch GmbH.....	21
I settori di business in cui opera Robert Bosch GmbH.....	21
La struttura societaria del gruppo Robert Bosch GmbH.....	22
2.4 - Robert Bosch GmbH branch in Italy	23
I numeri del gruppo in Italia	23
Le sedi principali in Italia	23
2.5 - Robert Bosch GmbH a Torino (PERBIT)	25
Lo sviluppo di PERBIT	25
L'organizzazione attuale di PERBIT	25
2.6 - Il servizio Sales Quality Warranty and Logistics (PERBIT/SQW-LOG).....	27
La struttura organizzativa di PERBIT/SQW-LOG	27
2.7 - La supply chain oggetto di studio	28
Bosch.....	28
Il magazzino intermedio (EDL)	28

Il cliente	29
Le relazioni tra gli attori in sintesi	29
CAPITOLO 3 - ANALISI DEL CASO	31
3.1 - Introduzione	31
3.2 - Fase n°1	31
3.3 - Fase n°2.....	34
3.4 - Fase n°3.....	34
Analisi mediante indici entropici.....	34
Analisi mediante Product Flow Number	35
CAPITOLO 4 - OSSERVAZIONI E VALIDAZIONI CONCLUSIVE	37
4.1 - Introduzione	37
4.2 - Osservazioni sulle analisi effettuate.....	37
Indici entropici.....	37
Product Flow Number.....	39
4.3 - Validazione tecnico economica	40
Ottimizzazione della raccolta dei vuoti	41
4.4 - Sviluppi futuri del progetto	47
4.4 - Conclusioni	48
ALLEGATI.....	51
Allegato 1: Tabella delle relazioni	51
Allegato 2: Matrici di analisi.....	54
Allegato 3: Matrice estesa dei trasferimenti T*	56
Allegato 4: Milk run di raccolta presso i clienti	59
Allegato 5: Milk run di consegna presso gli stabilimenti Bosch.....	60
Allegato 6: Risultati test algoritmo	61
Allegato 7: Codice algoritmo genetico.....	63
main.c.....	63
lettura.h	66
gen_casuale.h.....	69
mutazione.h.....	70
crossover.h	71
residui.h	71
improve.h.....	73
selezione.h	80
BIBLIOGRAFIA E SITOGRAFIA	82

PREMESSA

Il presente elaborato descrive l'attività di progetto svolta durante il periodo di tirocinio presso la business unit "automotive technologies" della Robert Bosch GmbH: la Robert Bosch GmbH Branch in Italy che ha sede a Torino, e che si configura come fornitore di componenti per l'industria automotive. La funzione logistica è l'ufficio in cui si è svolta l'esperienza di tirocinio, che si è sviluppato nell'ambito del progetto di Container Management System. In particolare, è stato analizzato il sistema di gestione dei Returnable Packaging relativi ai componenti che vengono forniti agli stabilimenti dei clienti localizzati in Italia.

L'elaborato è composto da due parti: una parte teorica e una parte pratica. La parte teorica espone gli strumenti teorici sui quali si fondano i contenuti sviluppati nella parte pratica. La parte pratica è volta a descrivere l'attività di progetto da un punto di vista strettamente operativo.

Il primo capitolo illustra i motivi che hanno determinato l'avvio del progetto. Sono poi messi in evidenza quali sono gli obiettivi intermedi e finali che si intendono raggiungere, declinandoli in termini di organizzazione del lavoro. Sono qui esposte le basi teoriche del metodo utilizzato e della disciplina a cui si fa riferimento. Viene inoltre dato spazio alla trattazione di alcuni topic nell'ambito dei Returnable Packaging, approfondendo l'argomento per il settore automotive.

Il secondo capitolo descrive la struttura organizzativa, i settori di business e le attività svolte dal gruppo Robert Bosch GmbH nel mondo e in Italia. Viene dato particolare rilievo alla sede di Torino ed alla divisione logistica di quest'ultima, in modo tale da descrivere il contesto entro il quale si sviluppa il progetto. Il capitolo presenta infine gli attori che operano nella catena logistica analizzata, descrivendone le attività svolte e caratterizzando la rete logistica studiata al fine di definire i confini entro i quali si sviluppa il progetto.

Il terzo capitolo presenta l'analisi effettuata sul caso in esame, descrivendone le modalità operative per ciascuna fase.

Il quarto capitolo presenta delle osservazioni sull'analisi effettuata, la validazione tecnico economica delle soluzioni proposte e le considerazioni conclusive.

PARTE TEORICA

CAPITOLO 1 - OBIETTIVI DEL PROGETTO E METODO UTILIZZATO

1.1 - Introduzione

Il presente capitolo illustra i motivi che hanno determinato l'avvio del progetto e gli obiettivi intermedi e finali che si intendono raggiungere. Tali obiettivi sono messi in relazione con la tempistica di progetto in termini di organizzazione delle fasi di lavoro. Sono poi esposti i fondamenti teorici del metodo utilizzato per lo studio del caso in esame.

1.2 - Motivi e obiettivi dello studio

Oggetto dello studio

In questo studio sono analizzati i flussi logistici relativi ai componenti prodotti e forniti da Bosch automotive technologies agli impianti dei clienti situati in Italia. Tra questi codici, sono analizzati quelli gestiti con consegna che avviene tramite magazzino intermedio (detto EDL) ed utilizzando Returnable Packaging (RP).

Motivi

La gestione dei materiali in una rete logistica come quella in esame, è particolarmente complessa. Questo può comportare l'insorgere di alcune inefficienze. La complessità deriva da alcune cause principali [1]:

Il flusso di materiali e informazioni in una supply chain può formare una rete di attività che può raggiungere numerosi fornitori, produttori e clienti. Ogni attore della rete può inoltre essere coinvolto in processi relativi ad altre supply chain, ciascuna con le sue caratteristiche. La complessità aumenta a causa delle molte variabili aleatorie in gioco. La supply chain è composta molte organizzazioni, ognuna delle quali si pone diversi obiettivi [1].

Di conseguenza è estremamente importante che, sistemi di questo tipo, vengano gestiti nella maniera più razionale possibile. Metodi di gestione inefficienti comportano il venire meno dei

vantaggi derivanti dall'adozione di soluzioni come quella relativa all'utilizzo dei RP. Prima di poter intraprendere qualsiasi tipo di decisione, è necessario avere una chiara visione delle dinamiche in gioco.

Questo ha determinato l'avvio di uno studio sui flussi di materiale gestito in RP nella supply chain in cui opera la Robert Bosch GmbH Branch in Italy [1].

Risultati attesi

Ciò che ci si attende da questo studio è un'analisi dei processi presenti nella situazione as-is. Questa analisi consentirà in seguito di identificare quali sono le aree critiche e quindi di stabilire opportunità di ottimizzazione.

Il fine pertanto è duplice [1]:

1. Ottenere un'immagine dei processi esistenti nella situazione "as-is";
2. Verificare se sussistono opportunità di migliorare i metodi di gestione dei RP.

Per ottenere ciò, è innanzitutto necessario avere una conoscenza precisa di come è fatta tutta la rete, non solo dei nodi che la formano ma anche di come insistono i vari flussi su questi nodi [2].

Il fine ultimo del progetto è quello di introdurre un sistema di registrazione fisico/contabile dei vuoti che transitano in inbound e outbound nell'EDL [1].

1.3 - Timing del progetto ed organizzazione del lavoro

Fase n°1

La prima fase di progetto prevede l'identificazione dei componenti coinvolti nello studio secondo le specifiche esposte nel paragrafo "Oggetto dello studio", degli impianti Bosch in cui tali codici (o part number, p/n) vengono prodotti, e degli impianti dei clienti ai quali sono forniti.

In questa fase verranno quindi identificate le relazioni che sussistono tra impianto, sia di produzione che di consegna, e codice [1]. Viene così definita univocamente la corrispondenza tra p/n, impianto di produzione Bosch del p/n, EDL di transito e impianto cliente a cui il p/n viene fornito (ship to party) [1].

I flussi così identificati verranno poi quantificati in termini di pezzi/anno. L'orizzonte temporale considerato è di un anno (anno 2014). Verranno inoltre identificati i codici degli imballi di primo e secondo livello di ogni p/n gestito in RP [1].

Fase n°2

A questo punto, sarà possibile costruire una mappa completa della rete logistica attraverso una rappresentazione matriciale delle relazioni individuate. Tali relazioni verranno identificate sia da un punto di vista qualitativo che quantitativo. Lo schema seguito per la rappresentazione matriciale dei flussi prevede che ogni elemento i, j della matrice rappresenta il flusso che sussiste tra il nodo i ed il nodo j della rete [1].

Fase n°3

A questo punto è disponibile una mappa dettagliata della rete logistica. Questa mappa descrive le relazioni presenti nella supply chain da un punto di vista qualitativo, si tratterà poi di proporre delle analisi supportate da metodi numerici [2]. A tale scopo, verrà quindi proposta un'analisi della rete basata su metodi di tipo quantitativo. L'analisi verrà effettuata attraverso la determinazione di alcuni indici che forniranno una caratterizzazione della rete da un punto di vista globale [2].

1.4 - I Returnable Packaging (RP)

Aspetti generali sui RP

Generalmente, imballi pensati per un utilizzo multiplo, sono denominati “Reusable Packaging” o “Returnable Packaging”. Lo stesso imballo è utilizzato dall'inizio della produzione presso il fornitore fino all'installazione finale sul veicolo [3].

Un sistema di gestione in anello chiuso identifica un percorso costituito da punti di produzione, di spedizione e punti di consegna.

Quando il materiale si muove lungo la supply chain, deve essere imballato in modo appropriato. L'imballo ha la funzione di proteggere il materiale durante il trasporto e lo stoccaggio [3]. Il materiale deve essere protetto da contaminazione, danni, influenze ambientali e perdite durante il trasporto [3]. L'imballo inoltre fornisce informazioni come il materiale contenuto, modalità di stoccaggio e movimentazione, informazioni di sicurezza, destinazione, luogo di partenza, destinazione e quantità. I RP, sono un tipo di imballo che si caratterizza per il fatto che è utilizzabile più volte. La divisione automotive di Bosch (BBM), utilizza questo tipo di imballi per alcuni dei part number che fornisce ai suoi clienti in Italia [3].

I vantaggi e le criticità gestionali dei RP

L'utilizzo di RP comporta alcuni vantaggi. Un primo vantaggio dei RP è un vantaggio di costo, in quanto sono utilizzabili più volte e quindi non necessitano di essere smaltiti: questo comporta minori imposte generate dallo smaltimento dei rifiuti. Devono subire un processo di pulitura e possono essere impiegati nuovamente [2].

Un secondo vantaggio deriva dalla facilità di stoccaggio ed handling: avendo dimensioni standardizzate consentono un impiego più efficiente dello spazio. I RP comportano inoltre vantaggi dal punto di vista dell'impatto ecologico-ambientale [4].

La valutazione di impiegare un sistema di RP nel settore automotive riveste una particolare importanza in quanto le società riferiscono che i costi di trasporto corrispondono al 5% - 7% delle loro entrate e sono in aumento [4].

I RP comportano tuttavia un investimento iniziale superiore rispetto ai tradizionali imballi a perdere. Generalmente, l'introduzione di un sistema di imballi a rendere, comporta anche una complessità addizionale nel sistema di gestione [3].

Pertanto, non sempre è conveniente l'utilizzo di RP: è necessaria una valutazione che ne dimostri una reale convenienza [2].

La valutazione di convenienza deve tenere in considerazione diversi aspetti come: materiale, trasporto, ispezione, pulitura e costi di gestione. In molte situazioni questi costi vengono ripartiti tra diverse società, con diverse strutture di costo.

Bosch, in seguito ad una valutazione di convenienza, ha stabilito per quali codici avviare la gestione mediante imballi a rendere e per quali, invece, mantenere un sistema di imballi a perdere.

Il trend, per quanto è stato possibile osservare sul campo, e dai dati disponibili in letteratura, vede un utilizzo crescente di questo tipo di imballi. Tale trend è ancora più marcato se si fa riferimento allo specifico settore dell'automotive [4].

I sistemi di gestione degli imballi in anello chiuso vengono organizzati allo scopo di consegnare un materiale nella maniera più efficiente possibile [2].

Una gestione efficace delle scorte di imballi dipende dalla disponibilità di dati e risorse presso ogni nodo della rete. Un sistema di gestione deve infatti facilitare la gestione dei dati ed il tracking, tenere conto delle fluttuazioni dei volumi e, quindi, prevenire le carenze di imballi presso i diversi punti della rete. I sistemi di tracking nella maggior parte dei casi sono, tuttavia, diversi per ogni attore della rete [5].

Possibili problemi derivanti da sistemi di gestione scorretti sono [3]:

- Perdite dovute a danni;
- Tempi di transito diversi da quelli previsti;
- Insufficiente quantitativo di imballi.

Queste problematiche sono fonte di un aumento dei costi che non aggiungono valore come, ad esempio, quelli derivanti da fermi linea.

I livelli di imballaggio

L'imballaggio viene generalmente suddiviso su tre livelli [4]:

- Imballaggio primario: imballaggio pensato in modo da costituire un'unità di vendita per l'utente finale o il consumatore;

- Imballaggio secondario: imballaggio pensato in modo da costituire il raggruppamento di un certo numero di unità di vendita;
- Imballaggio terziario: imballaggio pensato per facilitare la manipolazione ed il trasporto di merci, dalle materie prime ai prodotti finiti, di un certo numero di unità di vendita oppure di imballaggi multipli per evitare la loro manipolazione ed i danni connessi al trasporto.

Per questo motivo la tabella delle relazioni in allegato 1 riporta i campi indicati come “packing level 1” e “packing level 2”, allo scopo di indicare rispettivamente il codice dell’imballo primario ed il codice dell’imballo secondario.

1.5 - Il metodo: le possibili alternative

I metodi proposti dalla letteratura per l’analisi di una rete logistica sono classificabili secondo due categorie principali [6]:

- Supply chain mapping, che prevedono l’utilizzo di rappresentazioni grafiche;
- Supply chain analysis, metodi che analizzano la rete utilizzando indici di tipo quantitativo o qualitativo.

Queste due categorie principali si possono a loro volta scomporre in più tecniche come mostrato nei paragrafi successivi.

Supply chain mapping

Per quanto riguarda il supply chain mapping, la letteratura propone le seguenti tecniche [6]:

- Value Stream Mapping;
- P-Trans-Net;
- Supply Network Configuration;
- Casual Loop Diagram;
- Mappatura dinamica;
- Mappatura geografica.

Supply chain analysis

Il supply chain analysis, lo si può suddividere in ulteriori tre partizioni [6]:

- Metodologia di tipo qualitativo
 - Balance Measurement Method.
- Metodologie di tipo quantitativo
 - Product Flow Number (*PFN*);
 - Metodo degli indici entropici.
- Metodologia di tipo ibrido
 - *MFC*.

Non esiste una tecnica univocamente identificata come migliore delle altre. Esiste però una tecnica più adatta al contesto di analisi in cui ci si trova e dai risultati attesi dall'analisi [6].

1.6 - Il metodo utilizzato

Metodo degli indici entropici

La scelta di questo metodo scelta è stata fatta per avere a disposizione indicazioni numeriche sul processo analizzato e a causa dell'impossibilità di visitare fisicamente gli impianti di produzione e dei clienti allo scopo di predisporre una Value Stream Map. Driver di scelta è stata anche la disponibilità immediata e precisa, grazie al sistema SAP POE, di tutti i dati necessari a condurre un'analisi di tipo numerico.

Il metodo impiegato nel caso specifico fa parte della categoria supply chain analysis.

L'analogia tra le reti ecologiche e quelle industriali permette, con opportuni adeguamenti, di analizzare le seconde con metodi derivanti dall'analisi delle prime [12][13].

Regattieri et al. (2006) propongono una nuova metodologia per l'ottimizzazione delle supply chain, basata sul modello degli indici entropici. Questa tecnica si fonda sull'analisi dei network, diffusa soprattutto per lo studio degli ecosistemi naturali [7].

Un ecosistema è costituito da catene e reti alimentari. Le catene e le reti alimentari sono raggruppamenti di specie organizzate in strutture a rete, nelle quali l'energia e la materia vengono trasferite o trasformate [7].

Strutture come queste sono rappresentabili mediante grafi orientati che identificano le interazioni tra le diverse specie presenti. Nei grafi si individuano i nodi, cioè i vari compartimenti, collegati da archi che rappresentano gli scambi che avvengono nel sistema e con l'ambiente esterno [8][9].

I modelli entropici sono modelli matematici che derivano dalla teoria dell'informazione, teoria di cui Shannon e Weaver (1948) sono i precursori [7].

L'analisi delle reti ecologiche (Ecological Network Analysis - ENA) consiste in una serie di strumenti volti ad analizzare un ecosistema [8].

Con questi strumenti gli ecologisti sono riusciti a ricavare degli indici in grado di quantificare gli attributi di un ecosistema nella sua globalità. Tali misure, opportunamente adattate, si possono applicare anche per studiare le performance delle supply chain industriali [9].

Anche le supply chain si possono rappresentare mediante grafi orientati e studiare applicando le tecniche derivanti dalla network analysis [10].

Una supply chain si può infatti rappresentare mediante un grafo orientato contenente [10]:

- Nodi, che rappresentano agli attori della supply chain oggetto di analisi;
- Archi, per rappresentare le relazioni tra i nodi;
- Pesi associati agli archi, per indicare la dimensione dei flussi.

Ovviamente, bisogna prima individuare la tipologia di flussi che si vuole rappresentare e una unità di misura idonea ad esprimerne il peso degli archi [11]. Nel caso specifico, i pesi degli archi sono pari alla quantità di materiale fornito, misurato in pezzi/anno.

Si possono, quindi, identificare [7]:

- Il vettore I dei flussi in entrata (import vector): si tratta di materiale prime proveniente dall'ambiente esterno verso il network;
- Il vettore E dei flussi in uscita (export vector): è il materiale che viene inviato all'esterno;

- Il vettore D delle dissipazioni (dissipations vector): indica il materiale perso nella rete;
- La matrice T dei trasferimenti (transfers matrix): indica i flussi di materiale tra gli attori all'interno della supply chain considerata.

Si possono unire in una matrice tutte le informazioni sui flussi di una rete. Tale matrice è detta matrice estesa dei trasferimenti T^* [7].

	0	1	2	...	N	N+1	N+2	
0	0	Input [I]					0	0
1	0	Trasferimenti tra i compartimenti [T]					Export [E]	Dissipation [D]
2	0							
...	0							
N	0							
N+1	0	0	0	0	0	0	0	
N+2	0	0	0	0	0	0	0	

Figura 1.1: Matrice estesa dei trasferimenti

Questa tecnica richiede che ogni compartimento sia in stato di equilibrio. Questo significa che, per ogni nodo, deve valere il bilancio delle masse [6]: il flusso di materiale in entrata nel nodo deve essere pari al flusso in uscita dallo stesso nodo, al netto delle perdite.

Ciò è esprimibile dalla seguente relazione [7]:

$$T_i + I_i = T_i + E_i + D_i$$

Dove T_i rappresenta la somma della i -esima colonna, mentre T_i la somma della i -esima riga. I membri della precedente relazione individuano il vettore dei flussi entranti (inflow vector al primo membro - S) ed il vettore dei flussi uscenti (outflow vector al secondo membro - S').

Se il bilancio delle masse è verificato per ogni nodo del network, cioè vale la relazione $S = S'$, allora l'intero sistema si trova in stato di equilibrio [7].

Possono essere valutati otto indici, introdotti in, basati sul concetto di entropia, in grado di misurare il livello di complessità di una rete logistica a livello globale [7].

Per calcolarli è sufficiente disporre della matrice estesa dei trasferimenti T^* .

Gli indici entropici sono [7]:

- Total System Throughput (*TST*);

Il total system throughput fornisce un'indicazione sulle dimensioni dell'intero sistema e sulla quantità di materiale che viene movimentato nella rete.

Si calcola attraverso la seguente equazione [7]:

$$TST = \sum_{I=0}^n \sum_{j=0}^n T_{ij}^* = t_{..}$$

ed è dato dalla somma di tutti gli elementi della matrice estesa dei trasferimenti T^* . I puntini a pedice di t rappresentano la somma lungo le righe (il primo) e la somma lungo le colonne (il secondo) [7].

L'indice *TST* è una misura direttamente collegata allo sviluppo del network. Tuttavia, non fornisce informazioni su come il materiale è distribuito nella rete [7]. Questo significa che supply chain aventi stesso *TST* possono essere caratterizzate da configurazioni differenti.

- Average Mutual Information (*AMI*);

L'Average Mutual Information è un indice che stima il livello di organizzazione di una rete.

Per determinare l'*AMI* ci si basa sul lavoro di Shannon e Weaver, che misurarono l'entropia associata ad un evento X come somma dei prodotti tra la probabilità $p(i)$ di ogni output i ed il rispettivo logaritmo¹ [7]:

$$H_x = - \sum_{i \in X} p(i) \cdot \log[p(i)]$$

Una supply chain può essere vista come un insieme di transizioni tra i diversi attori, ognuna avente una certa probabilità di accadimento [6].

¹ La scelta della base del logaritmo determina l'unità di misura da associare all'entropia. Nel contesto della teoria dell'informazione, la base usata è due: il risultato si può quindi esprimere in bit. Nel seguito della trattazione si sottintende sempre logaritmo in base due.

La probabilità di avere una movimentazione da un nodo i ad un nodo j è proporzionale al flusso che va da i a j [7]:

$$p_{I,O}(i,j) = \frac{t_{ij}}{t_{..}}$$

Considerando, per ogni nodo, i rispettivi flussi in entrata e in uscita, si esprime l'entropia relativa all'intero sistema. In primo luogo, si calcola la joint entropy, definita come la somma delle probabilità che un prodotto esca dal nodo i ed entri nel nodo j [7]:

$$H_{I,O} = - \sum_{i=0}^n \sum_{j=0}^n p_{I,O}(i,j) \log[p_{I,O}(i,j)]$$

$$H_{I,O} = - \sum_{i=0}^n \sum_{j=0}^n \frac{t_{ij}}{t_{..}} \log\left(\frac{t_{ij}}{t_{..}}\right)$$

L'entropia associata ai flussi in uscita da ogni nodo è pari a [7]:

$$H_O = - \sum_{i=0}^n p_O(i) \log[p_O(i)]$$

$$H_O = - \sum_{i=0}^n \frac{t_{i.}}{t_{..}} \log\left(\frac{t_{i.}}{t_{..}}\right)$$

L'entropia associata ai flussi in entrata in ogni nodo è pari a [7]:

$$H_I = - \sum_{j=0}^n p_I(j) \log[p_I(j)]$$

$$H_I = - \sum_{j=0}^n \frac{t_{.j}}{t_{..}} \log\left(\frac{t_{.j}}{t_{..}}\right)$$

A questo punto è possibile determinare l'average mutual information, come somma delle entropie in entrata e in uscita, alla quale va sottratta la joint entropy [7]:

$$AMI = H_O + H_I - H_{I,O}$$

Il contributo di ogni elemento della matrice T^* alla joint entropy è dato da [7]:

$$-\frac{t_{ij}}{t_{..}} \log\left(\frac{t_{ij}}{t_{..}}\right)$$

Al fine di ricavare l'entropia in input H_I si sommano tutti i valori della matrice estesa T^* lungo ogni colonna j , in modo da ricavare t_j per ogni nodo j [7].

Il contributo del nodo j è quindi dato da:

$$-\frac{t_j}{t_{..}} \log \left(\frac{t_j}{t_{..}} \right)$$

Tale valore, sommato agli altri, fornisce l'entropia in input.

Sommando i coefficienti lungo ogni riga in modo da ottenere t_i , si ricavano i contributi [7]:

$$-\frac{t_i}{t_{..}} \log \left(\frac{t_i}{t_{..}} \right)$$

Questi valori, sommati, rappresentano l'entropia in output.

L'*AMI* misura il grado di organizzazione di una rete. Vale a dire quanto sono vincolati i percorsi dei materiali al suo interno [7].

In una rete in cui ogni nodo è connesso con tutti gli altri e i flussi sono uguali, l'*AMI* è nullo. In tal caso, nota la posizione di un materiale nella rete, non è possibile determinare la sua destinazione successiva [7].

- Ascendancy (*ASC*);

Essendo l'*AMI* adimensionale, può essere pesato con il *TST*, ottenendo l'indice chiamato ascendancy [7]:

$$ASC = TST \cdot AMI$$

L'ascendancy valuta il livello di sviluppo di un sistema sulla base della sua dimensione (*TST*) e del suo grado di organizzazione (*AMI*). Valutato per una rete logistica, l'ascendancy misura la frazione di materiale distribuito in modo efficiente [6].

- Development Capacity (*C*);

Valori elevati di ascendancy rappresentano una rete matura. In tal caso i nodi sono specializzati, gli scambi sono strutturati e i trasferimenti sono efficienti. Per una rete che ha raggiunto il suo massimo sviluppo, l'ascendancy è uguale alla sua massima capacità di sviluppo [7]. Il massimo

potenziale di sviluppo di una rete è misurato attraverso la development capacity. Tale valore rappresenta il limite superiore per l'ascendancy [6].

La development capacity si determina dal prodotto tra la joint entropy e l'indice TST come mostrato di seguito [7]:

$$C = TST \cdot H_{I,O}$$

Nel contesto industriale, questo indice, rappresenta il massimo potenziale di sviluppo di una rete.

La development capacity si può scomporre in due componenti: l'ascendancy, che misura il livello di organizzazione, e l'overhead, che rappresenta la ridondanza dei flussi [6].

- Overhead (Φ).

L'overhead indica la parte di development capacity che rimane non organizzata, ovvero [6]:

$$\Phi = C - ASC$$

L'overhead è dato da quattro contributi:

- Overhead in input Φ_I ;
- Overhead in export Φ_E ;
- Overhead in dissipation Φ_D ;
- Redundancy R .

Le prime tre componenti sono legate agli scambi con l'esterno, mentre la redundancy deriva dalle ridondanze dei percorsi nella rete. Questo significa che valori elevati di redundancy indicano la presenza di molti percorsi paralleli [6][7].

È possibile esprimere i quattro indici appena esposti in termini percentuali rispetto alla development capacity. Questo permette il confronto di reti diverse [7].

Product Flow Number (PFN)

Il flusso di materiali in una supply chain può essere assimilato al moto di un fluido in una condotta. Esso, può quindi essere studiato attraverso gli stessi metodi numerici. Il Product Flow Number si basa infatti sul numero di Reynolds (R_e) [15][16]. Il numero di Reynolds si definisce come rapporto tra le forze inerziali (ρv^2) e le forze viscosi ($\mu v/L$). L rappresenta la lunghezza

caratteristica e dipende dalla conformazione della condotta in cui si muove il fluido, μ rappresenta la viscosità dinamica del fluido e ρ è la sua densità [15] [16]. Il numero di Reynolds misura la tendenza del moto di un fluido ad essere laminare piuttosto che turbolento. Bassi valori del numero di Reynolds indicano che il flusso è laminare, al contrario si tratta di un flusso turbolento [15] [16]. Se il flusso è laminare prevalgono le forze viscosse. Se il flusso è turbolento prevalgono le forze inerziali. Si assume che la condotta abbia sezione circolare. In tal caso il flusso è considerato laminare per R_e inferiori a 2300, in regime transitorio per valori compresi tra 2300 e 4000 e turbolento per valori superiori a 4000 [15] [16].

Il numero di Reynolds può essere calcolato come:

$$R_e = \frac{Q_t \cdot n \cdot c \cdot m}{r \cdot t}$$

Q_t : quantità totale ordinata [pezzi/anno]

n : numero di fornitori

c : numero di possibili configurazioni di prodotto

m : fattore di caratterizzazione

r : ritmo di produzione [pezzi/mese]

t : tempo di consegna [mese]

Le forze inerziali sono esprimibili come:

$$\frac{Q_t \cdot n \cdot c}{r}$$

e le forze viscosse come:

$$\frac{t}{m}$$

PARTE PRATICA

CAPITOLO 2 - IL CONTESTO IN CUI SI SVILUPPA IL PROGETTO

2.1 - Introduzione

Questo capitolo descrive in primo luogo alcune peculiarità del settore di riferimento per questo studio: quello dell'automotive. Viene poi presentato il gruppo Bosch, descrivendone le origini e lo sviluppo, i settori di business in cui è attivo e le divisioni commerciali in cui si organizza. Verrà illustrato il ruolo della sede di Torino e della sua funzione logistica, che costituisce l'oggetto della trattazione. Vengono infine presentati gli attori che operano nella supply chain in esame.

2.2 - Il settore di interesse: l'automotive

Il settore dell'automotive

Il settore automotive rappresenta un mercato di estrema rilevanza nell'economie europea, giapponese e americana. La complessità tecnica di un'automobile spinge a cercare di implementare migliori tecniche gestionali, che vengono trasferite anche ad altri settori produttivi. È dunque possibile affermare che l'industria automobilistica rappresenta una forza trainante nello sviluppo di un Paese [17].

Si tratta di un settore di estrema rilevanza per il fatturato registrato, l'occupazione e l'indotto generati. Con riferimento all'anno 2008, il fatturato del settore è stato superiore ai 710 miliardi di euro in Europa, di cui 95 miliardi in Italia [17].

L'automotive è un settore dove l'innovazione ha importanza strategica. Per questo, la spesa in ricerca e sviluppo è la più elevata con riferimento agli investitori privati.

Infatti, si hanno 20 miliardi di euro spesi nel 2008 a livello europeo e 3 miliardi in Italia.

Nell'ambito di tali supply chain i produttori di autovetture rappresentano i player principali [17].

Le fusioni avvenute negli ultimi 15 anni, hanno dato vita a grandi gruppi. Tra questi, i principali dieci detengono ad oggi l'80% della produzione globale di automobili [17].

La supply chain dell'automotive

L'industria automobilistica non è rappresentata solo dal prodotto finito. La catena del valore inizia infatti dal mercato della fornitura delle materie prime e termina con il ritiro a fine vita delle autovetture o con il riciclo di alcune delle loro parti [17].

I prodotti del settore automotive sono manufatti di notevole complessità tecnologica e caratterizzati da un'elevata numerosità di componenti.

A tale complessità si aggiunge la complessità dei processi di approvvigionamento e di produzione. I fornitori sono infatti localizzati in diverse aree geografiche che si sviluppano su scala globale [17].

Tali caratteristiche pertanto determinano le criticità gestionali a livello di logistica e di distribuzione.

Nel suo insieme, la filiera dell'automotive è molto articolata. Tuttavia si può considerare composta da due parti principali: quella della produzione e assemblaggio dei veicoli e quella di distribuzione [17].

Nella parte di produzione e assemblaggio, sono presenti flussi provenienti da molti fornitori diretti agli impianti produttivi secondo vari metodi gestionali, modalità di trasporto e frequenze di spedizione.

La parte relativa alla distribuzione può sembrare meno complessa a causa del ridotto numero di attori coinvolti. Infatti, sono presenti operatori logistici che si fanno carico di movimentare lungo tratte nazionali e internazionali i veicoli fino al punto vendita [17].

Tuttavia, le complessità legate alla logistica, caratterizzano tale sezione della rete in termini di elevata criticità. Infatti, soprattutto in tale sezione, trovano spazio i problemi connessi con i passaggi di responsabilità relativa ad un prodotto ad alto valore [17].

2.3 - Il gruppo Robert Bosch GmbH

Le origini del gruppo Robert Bosch GmbH

Le origini del gruppo risalgono al 1886, anno in cui Robert Bosch avvia a Stoccarda un laboratorio di meccanica di precisione ed ingegneria elettronica [1]. La prima innovazione fu il magnete per l'accensione a bassa tensione (1897). Esso rappresenta il punto di partenza per l'espansione dell'azienda, portando all'apertura del primo stabilimento industriale nel 1901 [1]. Attualmente la Robert Bosch GmbH è la maggiore produttrice mondiale di componenti per il settore automotive ed opera in oltre 150 paesi nel mondo con 300 filiali e 70 stabilimenti produttivi [1]. L'azienda conta oltre 300000 dipendenti di cui 42000 di essi dedicati alla ricerca e sviluppo. Nel 2012 il fatturato complessivo è stato di 46,1 miliardi di euro di cui il 63 % derivante dal business della tecnica per autoveicoli [1].

La Robert Bosch GmbH è attiva anche in altri settori.

I settori di business in cui opera Robert Bosch GmbH

Oltre alla business unit automotive technologies (BBM - Bosch Mobility Solutions), la Robert Bosch GmbH è attiva nei mercati [1]:

- Beni di consumo (BBG): questa area di business offre una gamma di prodotti e soluzioni per elettrodomestici ed è leader di mercato mondiale nel settore degli elettrodomestici e dei relativi accessori.
- Tecnologia industriale (BBI): il settore di tecnologia industriale comprende le divisioni drive and control technology. In questo contesto opera Bosch Rexroth, che si occupa di tecnologie di comando ed azionamento e tecnica per l'imballaggio.
- Tecnologie energetiche e costruttive (BBE): l'area di business energia e tecnologie costruttive offre prodotti e soluzioni per termotecnica, sistemi di sicurezza e energia solare.

Il sistema di business di Bosch (Bosch Business System - BBS) si fonda su tre processi core [1]:

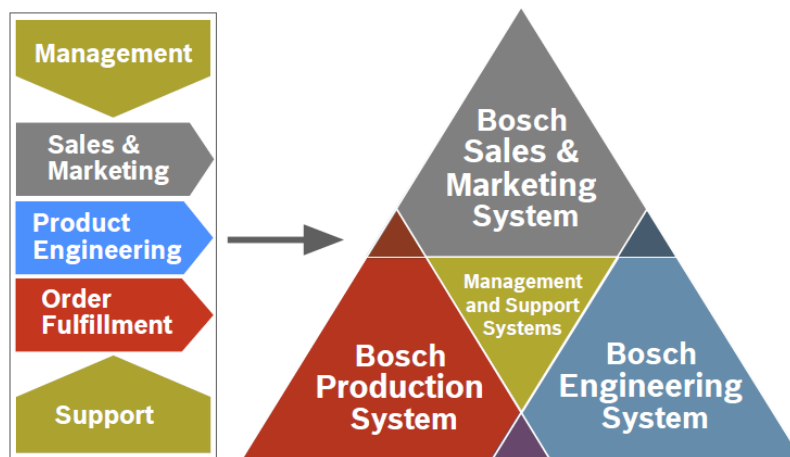


Figura 2.1: Bosch Business System (BBS).

La struttura societaria del gruppo Robert Bosch GmbH

La Robert Bosch ha una particolare struttura societaria. È infatti una società privata quasi completamente controllata da una fondazione senza scopo di lucro e non è quotata nei mercati finanziari. Questo tipo di struttura, le garantisce una grande autonomia finanziaria [1].

In linea con i principi del suo fondatore, i profitti vengono reinvestiti nella società per favorirne lo sviluppo e gli utili sono destinati a cause benefiche.

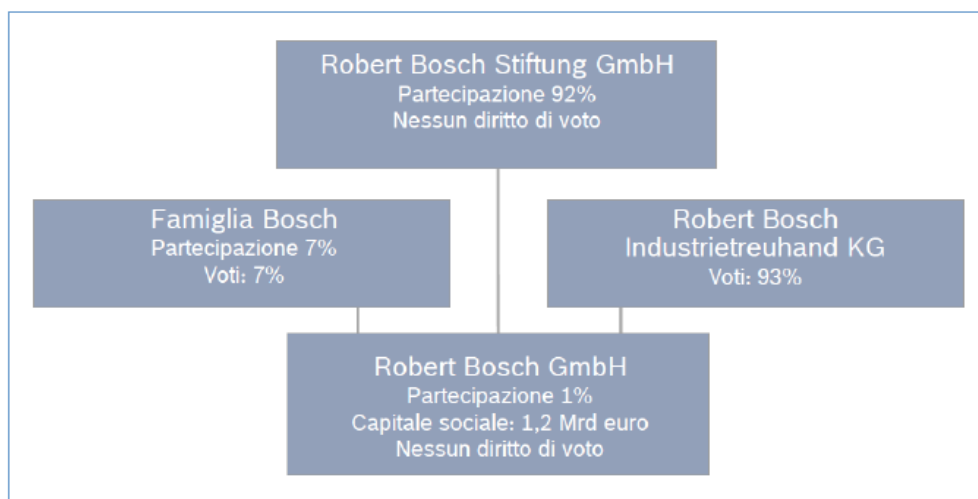


Figura 2.2: Struttura societaria Robert Bosch GmbH.

Il Gruppo è partecipato al 92% dalla fondazione Robert Bosch, al 7% dalla famiglia Bosch e all' 1% dalla Robert Bosch GmbH. Per quanto riguarda i diritti di voto il 93% di essi è detenuto dalla Robert Bosch Industrietreuhand KG mentre il 7% dalla famiglia Bosch [1]. La fondazione Robert Bosch nasce nel 1964 e riflette lo spirito del suo fondatore, che ha sempre cercato di

raggiungere l'armonia tra il mondo tecnologico ed economico e i bisogni della società. Le attività della fondazione hanno diffusione globale e si concentrano in tre ambiti: salute e scienza, educazione e cultura, relazioni internazionali [1].

"It is my intention, apart from the alleviation of all kinds of hardship, to promote the moral, physical and intellectual development of the people."

Robert Bosch

2.4 - Robert Bosch GmbH branch in Italy

I numeri del gruppo in Italia

Nel 1904 viene inaugurato il primo ufficio di rappresentanza Bosch a Milano. Oggi in Italia la Robert Bosch GmbH conta oltre 5800 collaboratori, per un fatturato di 1,7 Miliardi di Euro nel 2014 con 19 sedi commerciali e produttive e tre centri di ricerca e sviluppo [1].

Le sedi principali in Italia

Di seguito sono descritte brevemente le sedi Bosch in Italia e le relative attività principali [1].

- Brembate (BG), Aresi S.p.A.: accessori per elettrotensili.
- Cernusco (MI), Bosch Rexroth S.p.A.: vendita componenti e soluzioni "drive and control".
- Cinisello Balsamo (MI), Bosch Energy and Building Solutions Italy S.r.l.: sviluppo soluzioni per l'efficienza energetica degli edifici.
- Correggio (RE), Sicam S.r.l.: produzione di smontagomme ed equilibratrici.
- Lugo (RA), Deca S.r.l.: soluzioni per la documentazione tecnica e diagnostica OEM.
- Milano, Bosch Security Systems S.p.A.: vendita sistemi di sicurezza.
- Milano, Robert Bosch S.p.A.: sede commerciale dei settori Automotive Aftermarket, Elettrotensili, Termotecnica.

- Modugno (BA), Centro Studi Componenti per Veicoli S.p.A.: centro di sviluppo per pompe ad alta pressione.
- Modugno (BA), Tecnologie Diesel e Sistemi Frenanti S.p.A.: produzione pompe per sistemi “Common Rail”.
- Nonantola (MO), Bosch Rexroth Oil Control S.p.A.: sviluppo e fabbricazione di componenti idraulici per applicazioni mobili.
- Offanengo (CR), VHIT S.p.A.: progettazione, produzione e vendita di pompe del vuoto e pompe idrauliche.
- Sala Baganza (PR), Bosch Automotive Service Solutions S.r.l.: progettazione, produzione e distribuzione di apparecchi e servizi per la manutenzione di autoveicoli, motocicli e veicoli pesanti.
- San Lazzaro di Savena (BO), Holger Christiansen Italia S.r.l.: vendita di ricambi elettrici per automotive.
- Tavagnacco (UD), Freud produzioni industriali S.p.A.; Freud S.p.A.: produzione di lame circolari, punte, metallo duro e teste per la lavorazione del legno.
- Torino, ETAS GmbH Branch in Italy: soluzioni integrate, servizi di engineering, consulenza, formazione e supporto per lo sviluppo di sistemi nell'industria Automotive.
- Torino, Robert Bosch GmbH Branch in Italy: vendita, assistenza e marketing clienti settore automotive. Questa è la sede oggetto dello studio.
- Treviso (TV), aleo solar Distribuzione Italia S.r.l.: costruzione, produzione, riparazione e compravendita di prodotti, sistemi, impianti e dispositivi per la produzione di energia da fonti rinnovabili. Si occupa inoltre di compravendita e distribuzione di moduli ed impianti fotovoltaici.

2.5 - Robert Bosch GmbH a Torino (PERBIT)

Lo sviluppo di PERBIT

La storia della Bosch a Torino inizia nel 1932, tuttavia i prodotti Bosch di primo equipaggiamento per autoveicoli erano già ampiamente utilizzati sulle vetture Fiat. La crescita degli anni seguenti è avvenuta di pari passo con quella del cliente Fiat e degli altri produttori di auto nazionali fino al completo rinnovo della sede nel 2006 [1].

Il settore di tecnologie per l'automotive rappresenta attualmente il core business di Robert Bosch GmbH. In Italia i servizi e le vendite relative ad esso sono gestite dalla sede PERBIT (Permanent Establishment of Robert Bosch in Italy). La sua posizione è dovuta alla vicinanza agli stabilimenti Fiat di Mirafiori [1].

Gli investimenti del gruppo Bosch nella sede torinese hanno poi portato alla realizzazione di un importante polo per l'applicazione e la vendita dei prodotti per il primo equipaggiamento.

Inizialmente la prima unità operativa è stata la cosiddetta "Torino1" nel 1994, che costituiva la sede Bosch all'interno degli uffici e stabilimenti del cliente Fiat. Tale sede aveva lo scopo di impegnare i tecnici Bosch direttamente nelle attività di perfezionamento delle applicazioni dei prodotti Bosch [1].

La presenza dei dipendenti del fornitore nello stabilimento del cliente permette una maggiore collaborazione degli stessi e una maggiore rapidità di esecuzione degli adeguamenti ai sistemi e alle applicazioni sulle vetture.

La sede commerciale "Torino2" nasce nel 1997 con le divisioni relative ai sistemi frenanti, sistemi tergicristallo e prodotti after market. Dalle persone impiegate nel 1994 presso gli stabilimenti Fiat, si è giunti ad una forza lavoro di venti persone con la nascita del polo commerciale nel 1997 fino ad arrivare ai 150 dipendenti circa attuali [1].

L'organizzazione attuale di PERBIT

Attualmente Bosch a Torino è presente sottoforma di Permanent Establishment (PE). Questa sede viene infatti identificata dalla sigla PERBIT [1].

Il polo torinese di Robert Bosch GmbH si articola nelle attività di:

- Automotive Aftermarket - AA

La divisione Automotive Aftermarket è dedicata alla gestione delle forniture dei ricambi per le aziende del Gruppo Fiat a livello mondiale.

- Automotive Electronics - AE

La divisione Automotive Electronics si occupa di vendita e supporto per diodi, parkpilot e centraline elettroniche.

- Chassis Systems Control - CC

La divisione Chassis Systems si articola in diversi team che collaborano strettamente tra di loro. Acquisizione e pre-selling si dedicano alla partnership con il cliente. La divisione fornisce ABS, TCS, ESP, sensori, dispositivi elettronici per la sicurezza passiva e assistenza alla guida basata su sistemi radar e sensori video.

- Car Multimedia - CM

La divisione Car Multimedia si occupa di vendita e applicazione di autoradio, cd-changer, sistemi di navigazione, sistemi telematici e di assistenza per il conducente.

- Diesel Systems - DS

La divisione Diesel Systems fornisce sistemi completi con componenti idraulici, centraline, sensori, regolatori e gruppi di alimentazione carburante e componenti di controllo come la centralina, debimetro, sensori di pressione e di scarico (sonda lambda), candele di preriscaldamento e relativa centralina. Le attività di sviluppo di nuovi motori e modelli di vetture vengono svolte in collaborazione diretta con il cliente Fiat Powertrain Technologies.

- Electrical Devices - ED

La divisione Electrical Drives fornisce attuatori elettrici e sistemi di gestione raffreddamento motore e climatizzazione.

- Gasoline Systems - GS

La divisione Gasoline Systems fornisce sistemi ad iniezione diretta e indiretta, aspirati e sovralimentati. Fornisce inoltre centraline controllo motore per motori benzina, moduli di aspirazione, sensoristica, corpi farfallati, bobine di accensione, sonde lambda e componentistica per motori ad iniezione diretta, molti dei quali in comune con i motori Diesel. Altri componenti che sono in comune alla divisione DS sono: debimetri, moduli pedale acceleratore e gruppi di alimentazione carburante.

- Starter Motors and Generators - SG

La divisione Starter and Generators fornisce motorini di avviamento e alternatori.

2.6 - Il servizio Sales Quality Warranty and Logistics (PERBIT/SQW-LOG)

La struttura organizzativa di PERBIT/SQW-LOG

Il servizio PERBIT/SQW-LOG, si occupa della gestione della qualità e della gestione delle attività logistiche [1]. PERBIT/SQW3 è l'ufficio oggetto dell'esperienza di tirocinio.

Il servizio è organizzato in funzioni, le cui sigle ed attività corrispondenti sono riportate di seguito [1]:

- SQW-FI/SQW svolge la funzione di direzione del servizio SQW-LOG.
- SQW1 si occupa di customer claims management, quality data reporting, customer debit notes management e internal warranty costs management.
- SQW3 è preposto alla gestione delle attività di logistica relative al primo equipaggiamento.

La seguente figura propone un'organigramma sintetico relativo all'ufficio in cui si è svolta l'esperienza di tirocinio [1].

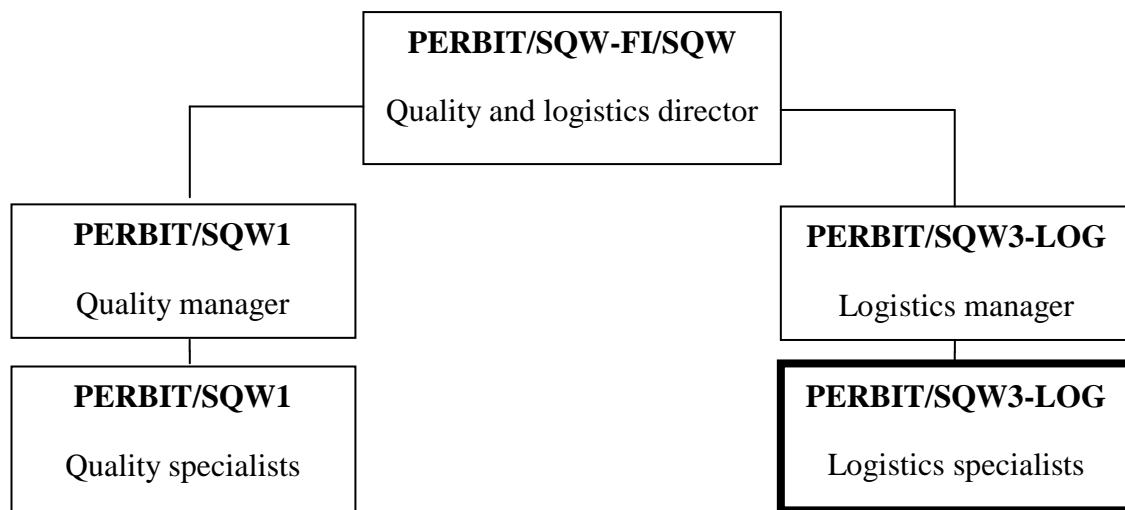


Figura 2.3: Organigramma sintetico di PERBIT/SQW-LOG.

2.7 - La supply chain oggetto di studio

Bosch

Bosch è un fornitore di componenti per l'industria automobilistica. La sede di Torino serve gli stabilimenti del gruppo Fiat Chrysler Automobile. Dalla sede di Torino sono serviti anche altri clienti minori, non appartenenti al gruppo FCA, ma localizzati in Italia. Gli stabilimenti produttivi sono dislocati in tutto il mondo.

Il magazzino intermedio (EDL)

Il provider di servizi logistici è esterno a Bosch e si occupa di effettuare fisicamente le spedizioni tra i diversi livelli della supply chain, della gestione del magazzino intermedio e di tutte le attività ad esso connesse.

Come esposto in precedenza, il cliente finale può essere raggiunto in 2 modi:

- Spedizione diretta dallo stabilimento Bosch al cliente finale;
- Spedizione indiretta tramite magazzino intermedio.

Nel secondo caso, quanto prodotto presso le unità Bosch, non viene consegnato direttamente allo stabilimento del cliente. In questa situazione il materiale spedito dagli stabilimenti produttivi Bosch transita per un magazzino intermedio gestito da un provider esterno di servizi logistici. In

seguito, dal magazzino intermedio il materiale giunge allo stabilimento del cliente. Le attività svolte nel magazzino intermedio ed i trasporti sono gestiti da un provider esterno di servizi logistici. I magazzini intermedi di cui PERBIT si avvale sono due:

- Settimo Torinese per i clienti nel Nord Italia;
- Maddaloni per i clienti nel Sud Italia.

Il cliente

I clienti ricevono il materiale spedito da Bosch secondo una delle due modalità precedentemente indicate. Il principale cliente della sede torinese di Bosch è il gruppo Fiat Chrysler Automobile. PERBIT si occupa quindi dell'asservimento degli stabilimenti FCA. Altri clienti minori non appartenenti al gruppo Fiat Chrysler Automobile, serviti da PERBIT, sono Automobili Lamborghini, Ducati, Piaggio, Lombardini e Denso Thermal Systems.

Le relazioni tra gli attori in sintesi

Una supply chain è costituita da tutte le parti coinvolte nel soddisfare la richiesta di un cliente. Quando un cliente ordina un materiale, Bosch si approvvigiona presso i suoi fornitori per produrre il materiale richiesto dal suo cliente e soddisfarne la domanda. Pertanto, le parti coinvolte in questo processo, sono il fornitore di Bosch, il produttore (Bosch stessa), il cliente ed il trasportatore. I materiali transitano lungo la supply chain attraverso ciascuno di questi step come mostra la figura seguente.



Figura 2.4: Rappresentazione schematica rete Bosch.

Esaminando il flusso con maggior dettaglio si osservano le seguenti relazioni. Bosch invia un ordine di acquisto al proprio fornitore ed il fornitore spedisce il materiale richiesto. Una volta

che il materiale è arrivato nell'area di ricezione merci, viene scaricato ed immagazzinato nell'apposita area di stoccaggio.

Da qui, il materiale viene spostato nelle differenti aree di produzione di un impianto. Dopo la produzione, i prodotti finiti vengono inviati nell'area di stoccaggio prodotti finiti. In conclusione il prodotto finito viene inviato al cliente che ne aveva fatto richiesta.

Tale consegna avviene secondo una delle due modalità descritte in precedenza.

Se la circolazione di materiali avviene solo tra due business partners, la gestione dei flussi è relativamente semplice.

Tuttavia nel contesto studiato si è in presenza di molti business partners: altri impianti, fornitori, clienti, fornitori di servizi esterni. Questi attori sono tutti coinvolti nel flusso del materiale.

A causa dell'elevato numero di attori coinvolti, e della complessità delle relazioni che li lega, la circolazione di materiale nella supply chain diventa particolarmente complessa e di difficile tracciabilità.

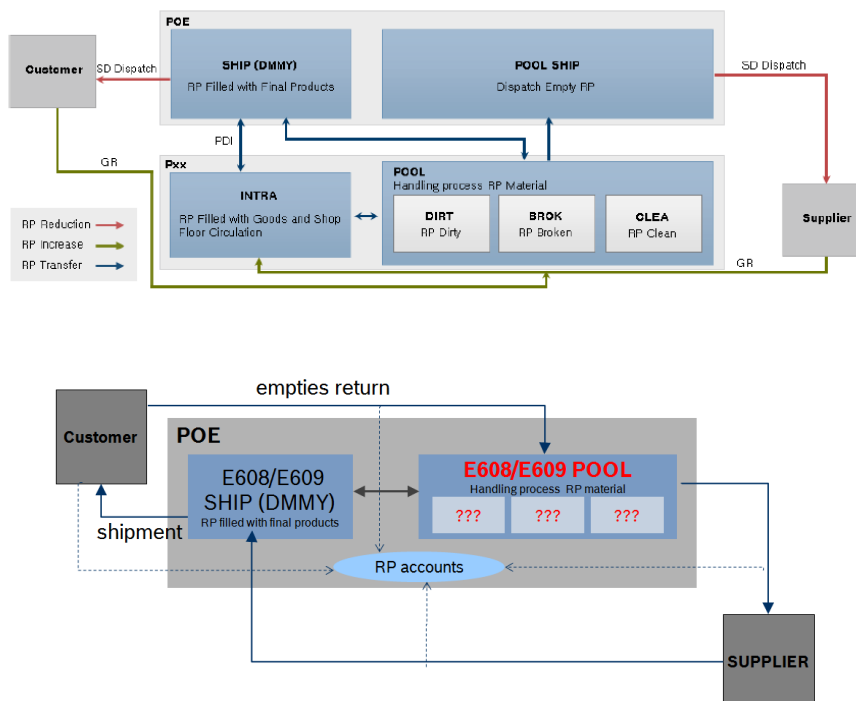


Figura 2.5: Flusso informativo e fisico relativo ai RP.

CAPITOLO 3 - ANALISI DEL CASO

3.1 - Introduzione

Il terzo capitolo presenta l'analisi effettuata sul caso in esame. Sono descritte in primo luogo le relazioni che legano gli attori della supply chain oggetto di studio. In secondo luogo vengono studiati i flussi di materiale da un punto di vista qualitativo e quantitativo.

L'esposizione di questo capitolo rispecchia le fasi di progetto precedentemente descritte.

3.2 - Fase n°1

Nel corso di questa fase sono stati raccolti i dati necessari a definire le corrispondenze tra part number (p/n), stabilimento di produzione Bosch, magazzino intermedio e ship to party.

Sono stati determinati in primo luogo tutti i codici gestiti in modalità indiretta. In seguito, con riferimento a tali codici, sono stati identificati quelli gestiti con RP.

Queste informazioni sono state dedotte dai report mensili CONAI.

Si è successivamente proseguito definendo, per ciascuno dei codici identificato come materiale gestito in modalità indiretta con RP, i seguenti attributi:

- Impianto di produzione Bosch;
- Magazzino intermedio (EDL) di transito;
- Impianti di consegna (ship to party);
- Quantità fornita per ciascuna ship to party nell'anno 2014;
- Modalità di ritorno dei vuoti.

Questi dati sono state raccolti dal sistema SAP POE.

Si osservi che, fissato il p/n, l'impianto di produzione Bosch è univocamente determinato, esiste quindi un unico punto di origine per ogni p/n. Allo stesso p/n possono però corrispondere più ship to party. Pertanto ciascuna ship to party può essere servita da più stabilimenti produttivi

Bosch. Specularmente, uno stabilimento produttivo Bosch, può servire più ship to party. Ogni impianto Bosch, infine, produce più p/n.

Per una migliore comprensione delle relazioni appena descritte si faccia riferimento ai dati raccolti in tabella riportata in allegato 1.

Si riporta di seguito la descrizione dei campi presenti nella tabella.

- BOSCH P/N: Bosch part number, indica il codice del componente secondo il sistema di codifica di Bosch;
- CUST. P/N: customer part number, indica il codice del componente secondo il sistema di codifica del cliente;
- DESCRIPTION: fornisce una descrizione sintetica del componente;
- GB: indica la divisione commerciale che gestisce il p/n;
- PLANT: indica il codice dello stabilimento in cui viene prodotto il p/n;
- CITY: indica la località del plant di produzione del p/n;
- EDL: indica l'EDL in cui transita il p/n;
- SHIP TO: indica il nome commerciale del cliente a cui viene fornito il p/n;
- SHIP TO CODE: indica il codice con il quale si identifica la ship to party;
- CITY: indica la località del plant della ship to party;
- PACKING CODE LV.1: indica il codice dell'imballo di primo livello del p/n;
- PACKING CODE LV.2: indica il codice dell'imballo di secondo livello del p/n;
- Q.TY 2014: indica la quantità fornita nel 2014 per il p/n, espressa in pezzi [pz];
- RETURN MODE: indica la modalità con cui avviene il viaggio di ritorno dei vuoti;

Di seguito è fornita una breve descrizione delle codifiche utilizzate per alcuni dei campi appena elencati che compongono la tabella delle relazioni riportata in allegato 1.

I GB sono così codificati:

Codice	GB	Codice	GB
AE	Automotive Electronics	ED	Electrical Drives
CC	Chassis Systems Control	GS	Gasoline Systems
CM	Car Multimedia	SG	Starter and Generators
DS	Diesel Systems		

I plant di produzione Bosch seguono la seguente codifica:

PLANT	Codice	PLANT	Codice
Ansbach	085W	Castellet	964W
Bamberg	060W	Eisenach	816W
Blaichach	050W	Feuerbach	011W
Blaj	RO1W	Hallein	936W
Braga	815W	Hatvan	658W
Bursa	939W	Jihlava	515W
Llica	640W	Madrid	965W
Miskolc	300W	Pecinci	287W
Penang	905W	Reutlingen	078W
Rutesheim	020W	Tienen	917W

La codifica utilizzata prevede di identificare i due EDL come segue:

- E608 o EAJE: Settimo Torinese;
- E609 o EAHN: Maddaloni.

Il viaggio di ritorno dei vuoti (RETURN MODE) può avvenire in modalità diretta o via EDL.

Gli stabilimenti di produzione Bosch, le ship to party e i magazzini intermedi sono rappresentabili mediante dei nodi di una rete. La rete che ne risulta è pertanto costituita da nodi che rappresentano gli stabilimenti di produzione Bosch, due nodi che rappresentano i magazzini intermedi e nodi che rappresentano le ship to party. Le relazioni tra i nodi della rete, qualora esistano, sono invece rappresentabili mediante degli archi ai quali è possibile assegnare un peso pari alla quantità fornita.

3.3 - Fase n°2

A questo punto, sulla base dei dati disponibili dalla tabella delle relazioni, è possibile costruire le matrici necessarie alla rappresentazione schematica dei flussi presenti nella rete in esame.

Attraverso queste matrici sarà poi possibile proseguire con l'analisi quantitativa secondo le tecniche precedentemente esposte. In allegato 2 si riportano tali matrici.

La prima tabella in allegato 2 riporta i valori in termini di numero di flussi (numero di p/n forniti) tra impianto di produzione Bosch, EDL e ship to party. La seconda tabella in allegato 2 riporta invece i flussi quantificati con riferimento ad un orizzonte di un anno (2014).

In allegato 3 si riporta la matrice estesa dei trasferimenti T^* , base di partenza per la successiva analisi quantitativa.

Si riportano di seguito i valori calcolati per gli indici esposti in precedenza, con riferimento alla situazione as-is.

3.4 - Fase n°3

Analisi mediante indici entropici

Dalla somma di tutti i flussi presenti nella matrice estesa dei trasferimenti T^* è possibile ricavare il TST:

$$TST = 9928371 \text{ pezzi/anno.}$$

Noto tale indice, si tratta di procedere con il calcolo della joint entropy $H_{I,O}$, per il caso in esame si ha che:

$$H_{I,O} = 5,092 \text{ bit.}$$

Per quanto riguarda l'entropia associata ai flussi in uscita H_O si ottiene:

$$H_O = 3,908 \text{ bit.}$$

In modo analogo, per quel che riguarda l'entropia associata ai flussi in ingresso H_I risulta:

$$H_I = 3,800 \text{ bit}$$

Pertanto, dalla relazione $AMI = H_O + H_I - H_{I,O}$ risulta immediato il calcolo dell'AMI:

$$AMI = 2,617 \text{ bit}$$

La determinazione dei rimanenti indici risulta immediata. Risulta infatti:

$$ASC = TST \cdot AMI = 25970872,76 \text{ bit}$$

e che la development capacity C è pari a:

$$C = TST \cdot H_{I,O} = 50541462,22 \text{ bit}$$

Dalla relazione $\Phi = C - ASC$ si determina infine che:

$$\Phi = 24570590,46 \text{ bit}$$

Tale indice, come già esposto in precedenza, è costituito da quattro componenti, per le quali risulta:

$$\Phi_I = \Phi_E = \Phi_D = 0 \text{ bit}$$

Da cui segue che:

$$R = \Phi$$

Il valore assunto da questi parametri consentirà in seguito di effettuare alcune osservazioni sulla supply chain in esame.

Analisi mediante Product Flow Number

Con l'obiettivo di determinare un ulteriore indicatore in grado di fornire una rappresentazione dello stato della supply chain analizzata si è scelto di valutare l'indice *PFN*.

A tal fine sono stati determinati i parametri necessari al suo calcolo.

In primo luogo la quantità totale di pezzi ordinata può essere ricavata dalla matrice estesa dei trasferimenti T^* , sommandone tutti gli elementi.

La quantità totale di pezzi forniti nell'orizzonte considerato risulta pertanto:

$$Q_t = 9925371 \text{ pezzi/anno.}$$

Il numero di fornitori è invece pari al numero di impianti produttivi Bosch. Si ha che:

$$n = 20$$

Il numero di possibili configurazioni di prodotto è invece pari a 92. Risulta infatti:

$$c = 92$$

Il ritmo di produzione r degli impianti Bosch è stato determinato considerando la situazione più “sfavorevole”. Questo è stato fatto allo scopo di mettersi in condizioni di maggiore sicurezza. Il parametro r può infatti ridurre il PFN in modo significativo.

A tal scopo, la capacità produttiva degli impianti Bosch, è stata considerata pari a quella effettivamente realizzata e non a quella massima.

Si ha quindi:

$$r = 827114 \text{ pezzi/mese.}$$

L’orizzonte di pianificazione considerato è invece pari a 12 mesi, si ha pertanto:

$$t = 12 \text{ mesi.}$$

Il valore da assegnare al fattore di caratterizzazione m è determinabile da tabella e risulta:

$$m = 1,5$$

<i>Intensity of Importance</i>	<i>Definition</i>	<i>Description</i>
0	No effect	This value means that the particulars have no effect and hence are not produced in actual practice.
1	Low effect	This value means that the particulars have low effect and hence signifies easy production.
2	Moderate effect	This value means that the particulars have moderate effect in terms of production.
3	Strong effect	This value means that the particulars have strong effect in terms of production.
4	Very strong effect	This value means that the particulars have very strong effect in terms of production.
5	Extreme importance	This value means that the particulars have extremely important influence in terms of production.

Figura 3.1: Fattore di caratterizzazione.

Con riferimento alla relazione che consente di determinare l’indice PFN :

$$PFN = \frac{Q_t \cdot n \cdot c \cdot m}{r \cdot t}$$

sostituendo, risulta:

$$PFN = 1840.$$

CAPITOLO 4 - OSSERVAZIONI E VALIDAZIONI CONCLUSIVE

4.1 - Introduzione

In questo capitolo vengono in primo luogo espone alcune osservazioni effettuate sui risultati delle analisi che sono state condotte. Verranno quindi interpretati i dati ottenuti e, su queste basi, identificate delle opportunità di miglioramento delle modalità di gestione degli RP ed esposti gli sviluppi futuri del progetto e validata la proposta di intervento. Infine, sono espone le conclusioni relative all'intero lavoro.

4.2 - Osservazioni sulle analisi effettuate

Indici entropici

Sulla base delle analisi condotte è possibile proporre alcune considerazioni riguardanti lo stato della supply chain in esame, con riferimento ai flussi dei p/n gestiti in RP.

Si riporta di seguito una tabella riassuntiva dei valori rilevati.

INDICE	VALORE	PERCENTUALE W.R.T. C
$TST = \sum_{i=0}^n \sum_{j=0}^n T_{ij}^* = t_{..}$	9925371,0	
$AMI = H_0 + H_I - H_{I,0}$	2,617	
$C = TST \times H_{I,0} = \sum_{i=0}^{N+2} \sum_{j=0}^{N+2} t_{ij} \log \left(\frac{t_{ij}}{t_{..}} \right)$	50541462,2	100%
$ASC = TST \cdot AMI$	25970871,7	51,4%
$\Phi = C - ASC$	24570590,4	48,6%
$R = \Phi$	24570590,4	48,6%

Il contributo della redundancy alla development capacity dipende dalle sovrapposizioni dei percorsi all'interno del sistema: valori elevati di redundancy rispecchiano un'elevata presenza di flussi paralleli. Questo è esattamente quanto accade per la rete in esame essendo nulle le

componenti di overhead in input, output e dissipation. Si ha quindi che l'overhead è pari alla redundancy.

Una considerazione che può essere effettuata riguarda i parallelismi tra i flussi. Tale considerazione deriva dall'osservazione dei valori assunti dai parametri overhead e redundancy.

Una sovrapposizione dei flussi rappresenta una voce di costo significativa e che non concorre alla creazione di valore, se non correttamente gestita. Questo è sottolineato dagli autori in [3].

Tuttavia questo costituisce anche un'opportunità se sfruttato in modo razionale a proprio vantaggio. La sovrapposizione dei flussi è infatti un indicatore della potenziale sovrapposizione dei percorsi dei veicoli. Questo fatto è indicatore dell'opportunità di gestire le spedizioni in modo aggregato. Questo vale in particolar modo per le spedizioni di ritorno dei vuoti in quanto non aggiungono valore al processo.

Con riferimento alle relazioni presenti tra gli attori della supply chain in esame è possibile osservare che sussiste la possibilità di aggregazione delle spedizioni. Si tratta quindi di capire se le logiche di pianificazione dei clienti, e quelle di produzione Bosch, lo consentono anche operativamente. Per quel che riguarda i viaggi di ritorno dei vuoti, la fattibilità dipende infatti anche dalla necessità degli imballi dei diversi p/n presso gli impianti produttivi Bosch. Questo è comunque facilitato dalla standardizzazione degli imballi.

Al fine di implementare una logica di questo tipo, è comunque necessaria la disponibilità di un sistema di tracciabilità. Tale sistema deve consentire ai logistics specialists Bosch di determinare la posizione dei materiali nella supply chain, e pianificare le spedizioni di conseguenza.

Questo allo stato attuale non è disponibile: il sistema rende infatti visibili informazioni sulle giacenze, la quantità consegnata cumulativa ad una certa data ed il viaggiante per un dato p/n. Il fine ultimo del progetto sarà infatti quello di introdurre un sistema di registrazione fisico/contabile dei vuoti che transitano nell'EDL.

Product Flow Number

Si richiama anzitutto la classificazione dei flussi in base al valore assunto dall'indice PFN .

PFN	TIPO DI FLUSSO
< 2300	Laminare
$4000 < PFN < 4000$	Transitorio
> 4000	Turbolento

Nel caso in esame, si è rilevato un valore di PFN pari a 1840, questo consente di classificare il flusso come laminare.

Si tratta dunque di analizzare l'impatto che avrebbero variazioni dei dati di input su tale valore.

Una prima considerazione da fare riguarda il parametro c : ovvero il numero di possibili varianti del prodotto. Un incremento di tale parametro avrebbe un duplice effetto sul valore del PFN . Questo deriva dal fatto che a numeratore si avrebbe un incremento di due parametri: c stesso e la quantità totale del consegnato Q_t . Questo effetto potrebbe essere ottenuto, per esempio, da un aumento del numero di codici gestiti in RP o, più semplicemente, da un aumento del venduto.

Un secondo effetto può essere dettato dal numero di fornitori: una razionalizzazione del parco fornitori (stabilimenti produttivi Bosch) comporterebbe un decremento del PFN .

Infine, una considerazione può essere fatta con riferimento alla capacità produttiva degli impianti Bosch. Si ricorda infatti che, il valore ottenuto per il PFN , è stato ottenuto non considerando la capacità massima di tali impianti.

Tra le considerazioni appena esposte circa i parametri che concorrono al calcolo del PFN , quella di maggiore interesse è quella riguardante l'incremento del numero di codici gestiti in RP. Il trend è infatti quello di un crescente ricorso a questa tipologia di imballi. Questo vale in particolare per il settore automotive. Appare dunque ragionevole ipotizzare che, per il futuro, ci si dovrà attendere uno spostamento in questa direzione. Questo, se da un lato rappresenta un punto di forza per la rete analizzata, dall'altro lato rappresenta una tendenza da mantenere costantemente monitorata alla luce dei risultati di questa analisi.

4.3 - Validazione tecnico economica

Sono possibili diverse logiche di aggregazione delle spedizioni di ritorno dei vuoti.

Si è scelto di realizzare un raggruppamento in cluster sia sugli impianti dei clienti che sugli impianti di produzione Bosch.

I cluster sono stati definiti in base alla posizione geografica degli impianti.

La raccolta dei vuoti presso gli stabilimenti dei clienti viene effettuata secondo i cluster che sono stati definiti (milk run). Terminata la raccolta i veicoli rientrano presso il magazzino intermedio.

A questo punto avviene la spedizione dei vuoti verso gli impianti di produzione Bosch, anch'essi raggruppati in cluster per area geografica. Anche queste spedizioni, naturalmente, vengono realizzate secondo i cluster in cui sono stati raggruppati gli impianti di produzione Bosch.

Allo scopo di mostrare l'effettiva convenienza economica derivante dall'aggregazione delle spedizioni, se ne propone una valutazione.

Si riportano di seguito queste considerazioni svolte focalizzandosi sul ritorno dei vuoti.

Per determinare il costo globale sostenuto nella situazione as-is si tratta di determinare i chilometri percorsi per ogni tratta, il numero di spedizioni effettuate nell'orizzonte di pianificazione considerato per ogni tratta e, quindi, il relativo chilometraggio totale. Noto il costo unitario per chilometro percorso è possibile stabilire il costo totale che è stato sostenuto nell'orizzonte di pianificazione considerato.

A questo punto si tratta di stimare il costo totale che sarebbe stato sostenuto organizzando le spedizioni in milk run.

Si determina anzitutto il chilometraggio relativo ad ogni milk run. Noto il costo unitario per chilometro percorso e una stima del numero di milk run che sarebbero stati effettuati nel 2014, si definisce, infine, il costo globale relativo al metodo di gestione con milk run.

In allegato 4 si riportano i milk run definiti per la raccolta presso gli stabilimenti dei clienti. In allegato 5 si riportano i milk run definiti per la consegna presso gli stabilimenti produttivi Bosch.

Con riferimento alla situazione con milk run, il costo totale stimato per la gestione delle spedizioni effettuate nell'orizzonte di pianificazione considerato si riduce di ~ 5 %.

Ottimizzazione della raccolta dei vuoti

La fase di raccolta dei vuoti risulta essere critica ai fini del mantenimento del livello di servizio desiderato. Il motivo risiede nel fatto che è indispensabile garantire la disponibilità di imballi presso gli stabilimenti di produzione Bosch . In caso contrario non sarebbe possibile dare continuità alle consegne presso gli impianti dei clienti a causa di shortage di imballi o, allo scopo di non interrompere le spedizioni, si dovrebbe ricorrere ai tradizionali imballi a perdere.

Al fine di mettere a disposizione dei logistici uno strumento di supporto alla pianificazione per la raccolta dei vuoti, tale problema è stato risolto attraverso un approccio di programmazione lineare. La funzione obiettivo che si intende massimizzare è data dalla priorità degli imballi trasportati.

Tale situazione può infatti essere vista come un problema dello zaino bidimensionale, noto come “2KP”. Tale problema considera un set di n oggetti. Ad ogni oggetto sono associati un profitto p_j con $j, j \in \{1 \dots n\}$ e due tipi di peso: w_j e v_j . Allo zaino sono pertanto associati due tipi di capacità: C e D . L'obiettivo del *Two-Constraint 0–1 Knapsack Problem* (2KP) è quello di selezionare un sottoinsieme di oggetti in modo tale che il profitto totale degli oggetti selezionati sia massimizzato e tale che i vincoli di capacità del contenitore siano rispettati per entrambe le capacità.

Si osservi che se $D \geq \sum_{j=1}^n v_j$ o $C \geq \sum_{j=1}^n w_j$ il 2KP si riduce al classico 1KP. Poiché il problema 1KP è NP-hard, lo stesso vale per il 2KP. Tuttavia, il problema dell'ammissibilità di una soluzione ha complessità polinomiale.

Si assume che i pesi sono interi non negativi. Si assume inoltre che i profitti e le capacità sono interi positivi tali che:

$$\sum_{j=1}^n w_j > C \text{ e } \sum_{j=1}^n v_j > D \quad (1)$$

$$\max_j \{w_j\} \leq C \text{ e } \max_j \{v_j\} \leq D$$

Oggetti con profitto non positivo possono, ovviamente, essere rimossi dal problema considerato. Se una delle condizioni in (1) non è soddisfatta, il corrispondente vincolo può essere rimosso riconducendosi al 1KP. Inoltre, un oggetto j tale che $w_j > C$ o $v_j > D$, può essere rimosso dall'istanza considerata.

Nel caso specifico, il profitto può essere considerato pari alla priorità associata ad uno specifico p/n . I pesi relativi all'oggetto j sono considerati pari al peso dell'imballo (w_j) ed al suo volume (v_j). Le capacità del contenitore sono considerate pari alla capacità dei veicoli impiegati per la raccolta dei vuoti presso gli stabilimenti dei clienti.

Il modello di riferimento diventa pertanto il seguente:

- n è il numero di oggetti
- $j: 1, \dots, n$ è l'indice associato agli oggetti
- p_j è la priorità relativa all'oggetto j
- w_j è il peso relativo all'oggetto j
- v_j è il volume relativo all'oggetto j
- C è la capacità del contenitore in termini di peso
- D è la capacità del contenitore in termini di volume
- x_j è una variabile binaria che dice se un oggetto è selezionato o meno.

x_j è pari a 1 se l'oggetto j è inserito nel contenitore, 0 altrimenti.

$$\max \sum_{j=1}^n p_j x_j$$

subject to

$$\sum_{j=1}^n w_j x_j \leq C$$

$$\sum_{j=1}^n v_j x_j \leq D$$

$$x_j \in \{0, 1\} \quad j = 1 \dots n$$

In questo studio si propone una soluzione a tale problema mediante un algoritmo genetico.

Gli algoritmi genetici sono algoritmi metaeuristici basati su una popolazione. L'obiettivo degli algoritmi meta euristici è quello di migliorare le soluzioni di un problema di ottimizzazione ottenute da tecniche euristiche. Gli algoritmi genetici propongono un'analogia tra l'evoluzione di una specie e l'evoluzione delle soluzioni di un problema di ottimizzazione. Le soluzioni vengono infatti assimilate agli individui di una popolazione, ciascuna componente della soluzione è invece assimilato ai geni dell'individuo. Le soluzioni vengono combinate tra loro (crossover) dopo avere subito un processo evolutivo (mutazione) allo scopo di ottenere delle soluzioni "figlie" e, iterando, si simulano diverse generazioni e, quindi, il processo evolutivo. L'idea è quella di far sopravvivere, ad ogni generazione (iterazione), solo le soluzioni migliori secondo uno specifico criterio di selezione.

L'algoritmo genetico nel caso specifico è stato implementato come illustrato di seguito.

- Step 1: Inizializzazione

L'obiettivo del primo step è quello di costruire una popolazione iniziale di n_{pop} soluzioni. In questo studio viene proposto un metodo di generazione delle soluzioni iniziali random.

Il metodo di generazione random inizializza una matrice s con n_{pop} righe ed n colonne tale che ogni elemento della matrice s ha valore intero determinato in modo casuale nell'intervallo $[0, 1]$. In questo modo gli oggetti sono selezionati in modo casuale.

- Step 2: Mutazione

Le soluzioni in s vengono mutate allo scopo di ottenere ulteriori n_{pop} soluzioni. Queste soluzioni vengono memorizzate nella matrice sm di n_{pop} righe ed n colonne.

Mutare una soluzione significa cambiare valore ai suoi geni da 0 a 1 e viceversa. In questo studio viene implementato il seguente metodo di mutazione.

Viene generato un array di lunghezza pari al 20% del numero di oggetti n . Ad ogni elemento di tale array viene assegnato un valore casuale nell'intervallo $[1, n]$. I valori memorizzati in questo array indicano infine quali sono i geni della soluzione coinvolti nel processo di mutazione da 0 a

1 o viceversa. L'array di mutazione è diverso per ogni soluzione in s . I geni mutati sono pertanto diversi per ogni soluzione in s .

- Step 3: Crossover

Le soluzioni in s ed sm vengono combinate tra loro allo scopo di ottenere $npop$ nuove soluzioni. Tali soluzioni vengono memorizzate nella matrice sc di $npop$ righe e n colonne.

Combinare tra loro le soluzioni a due a due significa ottenere una terza soluzione a partire dalle altre due. La terza soluzione viene ricavata dalle due soluzioni genitore prendendo una parte uguale a una soluzione genitore e una parte uguale all'altra soluzione genitore. Il metodo implementato in questo studio è il cosiddetto "single point crossover".

In questo caso la nuova soluzione [i -esima] in sc è composta da due parti. La prima parte (da 0 a $n/5$, $n/5$ incluso) è uguale alla corrispondente soluzione [i -esima] in sm , la seconda parte (da $n/5$ a n) è uguale alla corrispondente soluzione [i -esima] in s .

- Step 4: Selezione

A questo punto si effettua un controllo per eliminare la presenza di soluzioni gemelle. Le soluzioni vengono confrontate a coppie mediante la distanza (o norma) di Hamming. Per valutare la norma di Hamming si contano quanti sono i geni, di due soluzioni, che hanno valore diverso. Se questo contatore, alla fine del processo, avrà valore minore del 95% di n , viene eseguita la mutazione su una delle due soluzioni.

Infine, le soluzioni in s , sm e sc (memorizzate in un'unica matrice st di $3npop$ righe e n colonne), devono essere selezionate allo scopo di ridurre la popolazione di soluzioni da $3npop$ a $npop$. Le $npop$ soluzioni così selezionate costituiranno la popolazione iniziale per l'iterazione successiva.

Il processo di selezione avviene in base alla priorità totale associata ad ogni soluzione.

Le $npop$ soluzioni selezionate vengono memorizzate nella matrice s e l'algoritmo riparte dallo step 2. Il processo dallo step 2 allo step 4 viene ripetuto fino a quando viene raggiunto il criterio di stop. Nello specifico caso il criterio di stop è definito in termini di massimo numero di iterazioni dell'algoritmo ($ngen$). Chiaramente, durante i processi di generazione, mutazione e

crossover, possono essere generate soluzioni non ammissibili. Fino a questo momento l'ammissibilità delle soluzioni non era stata presa in considerazione.

Allo scopo di raggiungere l'ammissibilità di tutte le soluzioni prima del processo di selezione, una parte di algoritmo lavora in questo senso e con l'obiettivo di migliorare la qualità delle soluzioni mediante procedure di ricerca locale.

L'ammissibilità viene garantita nel seguente modo.

Se una o entrambe le capacità residue del contenitore sono negative, l'algoritmo inizia a rimuovere oggetti dalla soluzione, fino a quando entrambe le capacità residue non sono entrambe non negative. L'ordine con il quale gli oggetti vengono rimossi viene determinato calcolando, per ogni oggetto, il rapporto $(p / w + v)$. Gli oggetti vengono quindi rimossi in ordine crescente rispetto ai valori di $(p / w + v)$.

La procedura di ricerca locale viene eseguita come spiegato di seguito.

Per ogni oggetto non selezionato in una soluzione, l'algoritmo valuta se esiste un oggetto con priorità minore e pesi maggiori (anche solo uno dei due) selezionato nella soluzione. Se tale oggetto esiste, la procedura di ricerca locale lo rimuove dalla soluzione inserendo quello non selezionato inizialmente.

L'algoritmo è stato testato su 3 diversi set di istanze come suggerito da Martello e Toth nel loro lavoro "*Silvano Martello, Paolo Toth, (2003) An Exact Algorithm for the Two-Constraint 0-1 Knapsack Problem. Operations Research 51(5):826-835*".

A1: w_j , u_j , e p_j uniformemente casuali in $[1, R]$;

A2: w_j e u_j uniformemente casuali in $[1, R]$, p_j uniformemente casuali in $[w_j - R/10, w_j + R/10]$;

A3: w_j uniformemente casuali in $[1, R]$, u_j e p_j uniformemente casuali in $[w_j - R/10, w_j + R/10]$,

con le capacità pari a $C = 0,5 \cdot \sum_{j=1}^n w_j$ and $D = 0,5 \cdot \sum_{j=1}^n v_j$.

L'algoritmo è stato testato su 12 differenti istanze, 4 istanze per ogni set. Vengono utilizzati due files di input. Il primo contiene i dati relativi alla specifica istanza.

Il format di questo file di input è il seguente:

- Numero di oggetti (n);
- Capacità in termini di peso del bin (C);
- Capacità in termini di volume del bin (D);
- Priorità di ogni oggetto (p);
- Peso di ogni oggetto (w);
- Volume di ogni oggetto (v);

Questi files vengono letti automaticamente uno dopo l'altro dall'algoritmo.

Il secondo file di input contiene i settings dell'algoritmo:

- Numero di individui nella popolazione iniziale ($npop$);
- Numero di generazioni dell'algoritmo genetico ($ngen$);

I quali possono essere controllati dall'utilizzatore per settare l'algoritmo ad ogni run.

La tabella in allegato 6 riporta i risultati ottenuti dai test dell'algoritmo per ogni istanza.

Ogni colonna riporta un'informazione diversa. La prima colonna, denominata "istanza", riporta il codice relativo all'istanza di riferimento.

Il format è il seguente:

<tipo di istanza> - <numero di oggetti> - <valore di R>

La seconda colonna è denominata "valore ottimo". Questo valore è espresso in termini di priorità totale associato alla soluzione ottima per una specifica istanza test.

Le soluzioni ottime sono state ottenute risolvendo il modello in AMPL interfacciato a Cplex.

Le colonne " $npop$ " e " $ngen$ " mostrano i settings dell'algoritmo utilizzati per ogni istanza test.

$Npop$ è il numero di soluzioni nella popolazione iniziale. Quindi $npop$ è il numero di righe delle matrici s , sm and sc . $Ngen$ è il numero di iterazioni dell'algoritmo. Un'iterazione inizia con la mutazione delle soluzioni in s , e prosegue con: crossover, residui, improve e termina con la selezione delle migliori $npop$ soluzioni in termini di priorità totale.

La colonna “best found” mostra la priorità totale della migliore soluzione trovata dall’algoritmo tra le soluzioni della popolazione dell’ultima iterazione.

La colonna “items inseriti” dice quanti sono gli oggetti inseriti nella migliore soluzione metaeuristica “best found”.

La colonna “tempo di computazione” mostra, in secondi, il tempo di computazione dall’avvio al termine di un run dell’algoritmo.

Il campo “Error wrt opt.” Confronta la soluzione ottima con la corrispondente migliore soluzione metaeuristica trovata. Questa informazione è espressa in termini di rapporto (%) tra il valore della migliore soluzione (best found Lower Bound) ed il valore della soluzione ottima come mostrato di seguito.

$$Error\ wrt\ opt. = 1 - \frac{valore\ migliore\ LB\ metaeuristico}{valore\ soluzione\ ottima}$$

L’algoritmo è stato implementato in linguaggio C nell’ambiente DEV-C++ versione 5.6.3².

In allegato 7 si riporta il codice relativo all’algoritmo.

4.4 - Sviluppi futuri del progetto

Come è lecito attendersi da progetti aziendali di questo tipo, quanto sviluppato in questo elaborato, rappresenta solamente la fase di avvio di un progetto che ha un orizzonte di pianificazione di oltre un anno.

Il lavoro svolto in questa fase di progetto ha comunque una valenza fondamentale per la buona riuscita delle successive fasi. L’obiettivo è stato infatti quello di avere una visione dei processi logistici il più possibile aderente alla realtà e di identificare eventuali opportunità di migliorare i metodi di gestione del materiale RP nella supply chain. È indispensabile, per progetti con raggio d’azione di questo tipo, che tali opportunità vengano identificate fin dalle prime fasi di sviluppo per ovvi motivi di costo, lentezza e difficoltà di implementazione qualora venissero identificate

² Eseguito su sistema con processore Intel Core i7-3740QM CPU 2.70 GHz 16 GB RAM.

solo nelle fasi successive. Il fine ultimo del progetto è quello di introdurre un sistema di registrazione fisico/contabile dei vuoti che transitano in inbound e outbound dal magazzino. Il raggiungimento di tale obiettivo risulta anche una condizione necessaria per l'effettiva implementazione delle modalità di gestione suggerite.

4.4 - Conclusioni

Questo lavoro ha mostrato l'applicazione, in un reale contesto aziendale, di un metodo di supply chain analysis. Il metodo utilizzato, quello degli indici entropici, rappresenta una valida alternativa alla mappatura mediante Value Stream mostrandosi un metodo di facile e veloce implementazione. Tale metodo, si è rivelato inoltre un valido strumento per un approccio quantitativo all'analisi di una rete logistica, basato sulla teoria dell'informazione di Shannon e sulle misure entropiche introdotte nell'ambito dei sistemi ecologici.

L'analisi si è basata sugli indici introdotti da Allesina, Azzi, Battini e Regattieri nel loro lavoro *“Performance Measurement in Supply Chain: New Network Analysis and Entropic Indexes“*: il Total System Throughput, l'Average Mutual Information, l'Ascendancy, la Development Capacity, l'Overhead in Input, l'Overhead in Export, l'Overhead in Dissipation, e la Redundancy.

L'indice più significativo, come suggerito dagli stessi autori, appare essere l'ASC, espresso in termini percentuali rispetto alla development capacity, il quale fornisce una misura del grado di sviluppo di un sistema.

È comunque indispensabile valutare tutti gli indici, in quanto ciascuno di essi fornisce una diversa informazione riguardo la struttura della rete.

Ogni azienda può valutare questo set di indici allo scopo di quantificare l'impatto derivante da scelte strategiche, per confrontare la struttura della rete in diversi istanti di tempo, per confrontare diverse supply chain o per identificare potenziali criticità nella rete.

Nel caso specifico, la valutazione di questi indici, è stata utile per avere un riscontro oggettivo inerente il parallelismo tra flussi. A questa conclusione era possibile giungere anche senza il supporto di tale metodo. Il vantaggio che deriva dalla valutazione di questi indici è tuttavia duplice. In primo luogo attraverso la valutazione di tali indici si ha un riscontro oggettivo. In secondo luogo, viene messo a disposizione dei decisori un set di indicatori che permette di confrontare lo stato della rete in momenti diversi.

Il lavoro è stato sviluppato a partire da una presentazione dei motivi che hanno spinto all'avvio del progetto, una trattazione del tutto generale della problematica trattata, degli obiettivi e di una overview sui possibili approcci al problema.

Questi strumenti sono stati quindi calati nel contesto specifico.

Tale approccio ha permesso di verificare che sussiste la possibilità di realizzare le spedizioni in modo più efficiente. Quanto proposto è stato infine validato da un punto di vista tecnico verificandone l'effettiva convenienza rispetto alle attuali modalità di gestione.

ALLEGATI

Allegato 1: Tabella delle relazioni

BOSCH P/N	GB	PLANT	CITY	EDL	SHIP TO	SHIP TO CODE	CITY	PACKING CODE LV.1	PACKING CODE LV.2
01301015061KH	ED	300W	Miskolc, Hungary	E608	DENSO	1000902074	Poirino	6099.100.063	6000.515.255
01301015071KH	ED	300W	Miskolc, Hungary	E608	DENSO	1000902074	Poirino	6099.100.063	6000.515.255
01301015091KH	ED	300W	Miskolc, Hungary	E608	DENSO	1000902074	Poirino	6099.100.063	6000.515.255
01301015101KH	ED	300W	Miskolc, Hungary	E608	DENSO	1000902074	Poirino	6099.100.063	6000.515.255
0263007023129	CC	078W	Reutlingen, Germany	E608	FCA italy	1000901344	Villanova d'Asti	6099.100.021	6000.401.229
02630070231MU	CC	078W	Reutlingen, Germany	E608	FCA italy	1000901344	Villanova d'Asti	6099.100.063	6000.504.476
02650057381LC	CC	658W	Hatvan, Hungary	E608	LAMBORGHINI	1000901037	S.Agata Bolognese	6099.100.063	6099.506.147
02650076711FB	CC	816W	Eisenach, Germany	E608	FCA italy	1000901343	Torino	6099.100.058	6099.736.477
02650077691FB	CC	816W	Eisenach, Germany	E608	FCA italy	1000901343	Torino	6099.100.058	6099.736.477
026500792910L	CC	816W	Eisenach, Germany	E608	LAMBORGHINI	1000901037	S.Agata Bolognese	6099.100.021	6099.506.280
0265020240125	CC	085W	Ansbach, Germany	E608	LAMBORGHINI	1000901037	S.Agata Bolognese	6099.100.108	6099.506.147
02810060551FB	DS	816W	Eisenach, Germany	E608	FCA italy	1000901343	Torino	6099.100.063	6099.506.280
02850037035TZ	AE	965W	Madrid, Spain	E608	FCA italy	1000017577	Torino	6099.100.046	6000.973.162
044402202169E	DS	936W	Hallein, Austria	E608	NEW HOLLAND	1000009793	S.Mauro Torinese	6000.101.207	6000.514.508
044402202169E	DS	936W	Hallein, Austria	E608	IVECO	1000903420	Villanova d'Asti	6000.101.207	6000.514.508
044402202169E	DS	936W	Hallein, Austria	E608	ASTRA	1000901033	Piacenza	6000.101.207	6000.514.508
044402202169E	DS	936W	Hallein, Austria	E608	FPT industrial	1000901599	Pregnana Milanese	6000.101.207	6000.514.508
044402202169E	DS	936W	Hallein, Austria	E608	IVECO	1000903402	Brescia	6000.101.207	6000.514.508
044402305869E	DS	936W	Hallein, Austria	E608	FPT industrial	1000901599	Pregnana Milanese	6000.101.207	6000.514.672
044402305869E	DS	936W	Hallein, Austria	E608	ASTRA	1000901033	Piacenza	6000.101.207	6000.514.672
044402305869E	DS	936W	Hallein, Austria	E608	NEW HOLLAND	1000009793	S.Mauro Torinese	6000.101.207	6000.514.672
044402305869E	DS	936W	Hallein, Austria	E608	IVECO	1000903402	Brescia	6000.101.207	6000.514.672
04440230586GP	DS	936W	Hallein, Austria	E608	IVECO	1000012729	Villanova d'Asti	6099.100.046	\
04440230586GP	DS	936W	Hallein, Austria	E608	IVECO	1000012505	Villanova d'Asti	6099.100.046	\
04440230586GP	DS	936W	Hallein, Austria	E608	IVECO	1000012731	Villanova d'Asti	6099.100.046	\
044404203169E	DS	936W	Hallein, Austria	E608	FPT industrial	1000901599	Pregnana Milanese	6000.101.207	6000.503.185
044404203169E	DS	936W	Hallein, Austria	E608	IVECO	1000903402	Brescia	6000.101.207	6000.503.185
04440420391UH	DS	936W	Hallein, Austria	E608	FPT industrial	1000901599	Pregnana Milanese	6099.100.046	\
044404207469E	DS	936W	Hallein, Austria	E608	FPT industrial	1000901599	Pregnana Milanese	6000.101.207	6000.503.185
044404306469E	DS	936W	Hallein, Austria	E608	IVECO	1000903402	Brescia	6000.101.207	6000.503.185
044404306469E	DS	936W	Hallein, Austria	E608	FPT industrial	1000901599	Pregnana Milanese	6000.101.207	6000.503.185
044404306669E	DS	936W	Hallein, Austria	E608	FPT industrial	1000901599	Pregnana Milanese	6000.101.207	6000.503.185
044404306869E	DS	936W	Hallein, Austria	E608	FPT industrial	1000901599	Pregnana Milanese	6000.101.207	6000.503.185
044404306969E	DS	936W	Hallein, Austria	E608	FPT industrial	1000901599	Pregnana Milanese	6000.101.207	6000.503.185
04450100842V8	DS	515W	Jihlava, Czech Republic	E608	FCA italy - VM	1000019159	Cento	6099.120.005	\
0445020508238	DS	011W	Feuerbach, Germany	E608	FPT industrial	1000901595	Torino	6000.101.207	6000.505.290
04451104006V9	DS	939W	Bursa, Turkey	E608	FCA italy - VM	1000019159	Cento	6099.100.063	6099.940.502
04451105226V9	DS	939W	Bursa, Turkey	E608	FCA italy - VM	1000019159	Cento	6099.101.121	6000.514.862
04451105236V9	DS	939W	Bursa, Turkey	E608	FCA italy - VM	1000019159	Cento	6099.101.121	6000.514.862
04451105436V9	DS	939W	Bursa, Turkey	E608	FCA italy - VM	1000019159	Cento	6099.100.063	6000.515.443
04452160442V8	DS	515W	Jihlava, Czech Republic	E608	FCA italy - VM	1000019159	Cento	6000.101.207	\
04452160502V8	DS	515W	Jihlava, Czech Republic	E608	FCA italy - VM	1000019159	Cento	6000.101.207	\
33934290531L3	ED	964W	Castellet, France	E608	LAMBORGHINI	1000013022	S.Agata Bolognese	6099.100.046	\
76200001626HZ	CM	815W	Braga, Portugal	E608	IVECO	1000903402	Brescia	6099.100.063	6000.504.577
76200001626HZ	CM	815W	Braga, Portugal	E608	IVECO	1000012729	Villanova d'Asti	6099.100.063	6000.504.577

F005V003272HA	AE	658W	Hatvan, Hungary	E608	LAMBORGHINI	1000901037	S.Agata Bolognese	6000.120.260	6099.506.147
F005V011845KA	AE	658W	Hatvan, Hungary	E608	LAMBORGHINI	1000901037	S.Agata Bolognese	6099.100.108	6099.506.280
F005V015495KA	AE	658W	Hatvan, Hungary	E608	LAMBORGHINI	1000901037	S.Agata Bolognese	6099.100.108	6099.506.280
0263007014129	CC	078W	Reutlingen, Germany	E608	FCA italy	1000020109	Villanova d'Asti	6099.100.021	6000.401.229
33981233885PZ	ED	917W	Tienen, Belgium	E608	FCA italy	1000020109	Villanova d'Asti	6000.100.147	\
33981233895PZ	ED	917W	Tienen, Belgium	E608	FCA italy	1000020109	Villanova d'Asti	6000.100.147	\
0281004455129	DS	020W	Rutesheim, Geramny	E608	FCA italy	1000901338	Villanova d'Asti	6099.100.058	6099.736.109
0281004455129	DS	020W	Rutesheim, Geramny	E608	FCA italy	1000901344	Villanova d'Asti	6099.100.058	6099.736.109
0281006055129	DS	816W	Eisenach, Germany	E608	FCA italy	1000901338	Villanova d'Asti	6099.100.058	6099.736.477
0281006055129	DS	816W	Eisenach, Germany	E608	FCA italy	1000020109	Villanova d'Asti	6099.100.058	6099.736.477
0281006055129	DS	816W	Eisenach, Germany	E608	FCA italy	1000901344	Villanova d'Asti	6099.100.058	6099.736.477
028100306568F	DS	905W	Penang, Malaysia	E608	FCA italy	1000020109	Villanova d'Asti	6099.100.058	6000.321.836
028100306568F	DS	905W	Penang, Malaysia	E608	FCA italy	1000901338	Villanova d'Asti	6099.100.058	6000.321.836
028100306568F	DS	905W	Penang, Malaysia	E608	FCA italy	1000901344	Villanova d'Asti	6099.100.058	6000.321.836
028100306568F	DS	905W	Penang, Malaysia	E608	FCA italy	1000901343	Torino	6099.100.058	6000.321.836
0281006287129	DS	816W	Eisenach, Germany	E608	FCA italy	1000020109	Villanova d'Asti	6099.100.058	6099.736.477
0281006287129	DS	816W	Eisenach, Germany	E608	FCA italy	1000901344	Villanova d'Asti	6099.100.058	6099.736.477
0281006287129	DS	816W	Eisenach, Germany	E608	FCA italy	1000901338	Villanova d'Asti	6099.100.058	6099.736.477
0281006287129	DS	816W	Eisenach, Germany	E608	FCA italy	1000012448	Villanova d'Asti	6099.100.058	6099.736.477
0281006287129	DS	816W	Eisenach, Germany	E608	FCA italy	1000017577	Torino	6099.100.058	6099.736.477
0281006287129	DS	816W	Eisenach, Germany	E608	FCA italy	1000901343	Torino	6099.100.058	6099.736.477
0281006287129	DS	816W	Eisenach, Germany	E608	IVECO	1000012509	Suzzara	6099.100.058	6099.736.477
0281004473129	DS	020W	Rutesheim, Geramny	E608	FCA italy	1000901338	Villanova d'Asti	6099.100.058	6099.736.109
33981330161L7	ED	917W	Tienen, Belgium	E608	LAMBORGHINI	1000013022	S.Agata Bolognese	6000.100.145	\
33981330171L7	ED	917W	Tienen, Belgium	E608	LAMBORGHINI	1000013022	S.Agata Bolognese	6000.100.145	\
33981330191L7	ED	917W	Tienen, Belgium	E608	LAMBORGHINI	1000013022	S.Agata Bolognese	6000.100.145	\
026364501939D	CC	658W	Hatvan, Hungary	E608	LAMBORGHINI	1000019426	Castel Maggiore	6000.120.122	\
02810024106NE	DS	RO1W	Blaj, Romania	E608	FPT industrial	1000014778	Villanova d'Asti	6000.120.122	\
02810024106NE	DS	RO1W	Blaj, Romania	E608	FPT industrial	1000901598	Torino	6000.120.122	\
02810024106NE	DS	RO1W	Blaj, Romania	E608	FPT industrial	1000014772	Villanova d'Asti	6000.120.122	\
044404203969E	DS	936W	Hallein, Austria	E608	FPT industrial	1000901599	Pregnana Milanese	6000.101.207	6000.503.185
04451103016V9	DS	939W	Bursa, Turkey	E608	FCA italy - VM	1000019159	Cento	6099.100.063	6000.610.044
04451106572FE	DS	939W	Bursa, Turkey	E608	FPT industrial	1000901595	Torino	6099.101.121	6000.514.862
33981330181L7	ED	917W	Tienen, Belgium	E608	LAMBORGHINI	1000013022	S.Agata Bolognese	6000.100.145	\
02033001401MU	AE	078W	Reutlingen, Germany	E609	SATA	1000902452	Melfi	6099.739.075	6000.930.377
02040541535YL	CC	640W	Llica, Spain	E609	SATA	1000902452	Melfi	6000.121.238	\
02040541545YL	CC	640W	Llica, Spain	E609	SATA	1000902452	Melfi	6000.121.238	\
0263007014129	CC	078W	Reutlingen, Germany	E609	SATA	1000902447	Melfi	6099.100.021	6000.401.229
0265244043129	CC	050W	Blaichach, Germany	E609	SEVEL	1000902463	Atessa	6000.101.207	6000.503.213
0265260708129	CC	050W	Blaichach, Germany	E609	SEVEL	1000902463	Atessa	6000.101.207	6000.503.213
044501051267Y	DS	515W	Jihlava, Czech Republic	E609	FPT industrial	1000901594	Foggia	6000.101.207	6000.505.290
0445010525238	DS	011W	Feuerbach, Germany	E609	FPT industrial	1000901594	Foggia	6000.101.207	6000.505.290
0445010586238	DS	011W	Feuerbach, Germany	E609	FPT industrial	1000901594	Foggia	6000.101.207	6000.505.290
044511041811U	DS	060W	Bamberg, Germany	E609	FPT industrial	1000901594	Foggia	6099.101.121	6000.514.862
0445110419129	DS	060W	Bamberg, Germany	E609	FCA italy	1000901413	Pratola Serra	6099.101.121	6000.113.511
04451105242FE	DS	060W	Bursa, Turkey	E609	FCA italy	1000901413	Pratola Serra	6099.101.121	6000.503.227
044511056460W	DS	939W	Bursa, Turkey	E609	FPT industrial	1000901594	Foggia	6099.101.121	6000.514.862
0281004455129	DS	020W	Rutesheim, Geramny	E609	SATA	1000902447	Melfi	6099.100.058	6099.736.109
0281004455129	DS	020W	Rutesheim, Geramny	E609	FCA italy	1000901340	Piedimonte S.Germano	6099.100.058	6099.736.109
0281004455129	DS	020W	Rutesheim, Geramny	E609	SEVEL	1000013012	Atessa	6099.100.058	6099.736.109
028100306568F	DS	905W	Penang, Malaysia	E609	SATA	1000902447	Melfi	6099.100.058	6000.321.836
028100306568F	DS	905W	Penang, Malaysia	E609	FCA italy	1000901340	Piedimonte S.Germano	6099.100.058	6000.321.836
028100306568F	DS	905W	Penang, Malaysia	E609	FCA italy	1000901312	Pomigliano d'Arco	6099.100.058	6000.321.836

0281006287129	DS	816W	Eisenach, Germany	E609	SATA	1000902447	Melfi	6099.100.058	6099.736.477
0281006287129	DS	816W	Eisenach, Germany	E609	FCA italy	1000901340	Piedimonte S.Germano	6099.100.058	6099.736.477
0281006287129	DS	816W	Eisenach, Germany	E609	SEVEL	1000013012	Atessa	6099.100.058	6099.736.477
0281006287129	DS	816W	Eisenach, Germany	E609	FCA italy	1000901312	Pomigliano d'Arco	6099.100.058	6099.736.477
0281004473129	DS	020W	Rutesheim, Geramny	E609	SATA	1000902447	Melfi	6099.100.058	6099.736.109
0281004473129	DS	020W	Rutesheim, Geramny	E609	FCA italy	1000901340	Piedimonte S.Germano	6099.100.058	6099.736.109
33970216452Y1	ED	300W	Miskolc, Hungary	E609	SATA	1000902447	Melfi	6000.100.903	\
33970217886TN	ED	287W	Pecinci, Serbia	E609	SATA	1000902447	Melfi	6000.402.532	\
33981233125PZ	ED	917W	Tienen, Belgium	E609	SATA	1000902447	Melfi	6000.100.147	\
F000BL07782Y1	SG	300W	Miskolc, Hungary	E609	FPT industrial	1000901589	Foggia	6000.121.360	\
F000BL07822Y1	SG	300W	Miskolc, Hungary	E609	FPT industrial	1000901589	Foggia	6000.121.360	\
33981233135PZ	ED	917W	Tienen, Belgium	E609	SATA	1000902447	Melfi	6000.100.147	\
33981233885PZ	ED	917W	Tienen, Belgium	E609	SATA	1000902447	Melfi	6000.100.147	\
33981233895PZ	ED	917W	Tienen, Belgium	E609	SATA	1000902447	Melfi	6000.100.147	\
33981351752Y1	ED	300W	Miskolc, Hungary	E609	SATA	1000902447	Melfi	6099.100.046	6000.747.663
33981351762Y1	ED	300W	Miskolc, Hungary	E609	SATA	1000902447	Melfi	6099.100.046	6000.747.663
33970216451GF	ED	300W	Miskolc, Hungary	E609	SATA	1000902447	Melfi	6000.402.532	\
33981351751GF	ED	300W	Miskolc, Hungary	E609	SATA	1000902447	Melfi	6000.101.352	6099.506.280
33981351761GF	ED	300W	Miskolc, Hungary	E609	SATA	1000902447	Melfi	6000.101.352	6099.506.280
33970216462Y1	ED	300W	Miskolc, Hungary	E609	SATA	1000902447	Melfi	6000.100.903	\
33981232755PZ	ED	917W	Tienen, Belgium	E609	SATA	1000902447	Melfi	6000.100.147	\
33981232765PZ	ED	917W	Tienen, Belgium	E609	SATA	1000902447	Melfi	6000.100.147	\
33981351771GF	ED	300W	Miskolc, Hungary	E609	SATA	1000902447	Melfi	6000.101.352	6099.506.280
33981351781GF	ED	300W	Miskolc, Hungary	E609	SATA	1000902447	Melfi	6000.101.352	6099.506.280
33981353396TN	ED	287W	Pecinci, Serbia	E609	SATA	1000902447	Melfi	6000.101.746	6000.404.959
33981353406TN	ED	287W	Pecinci, Serbia	E609	SATA	1000902447	Melfi	6000.101.746	6099.506.280
33970217896TN	ED	287W	Pecinci, Serbia	E609	SATA	1000902447	Melfi	6000.402.532	\
0281006055129	DS	816W	Eisenach, Germany	E609	SATA	1000902447	Melfi	6099.100.058	6099.736.477
0281006055129	DS	816W	Eisenach, Germany	E609	FCA italy	1000901312	Pomigliano d'Arco	6099.100.058	6099.736.477
0281006055129	DS	816W	Eisenach, Germany	E609	FCA italy	1000901340	Piedimonte S.Germano	6099.100.058	6099.736.477
0281006055129	DS	816W	Eisenach, Germany	E609	SEVEL	1000013012	Atessa	6099.100.058	6099.736.477
33981233175PZ	ED	917W	Tienen, Belgium	E609	SATA	1000902447	Melfi	6000.100.147	\
33981233185PZ	ED	917W	Tienen, Belgium	E609	SATA	1000902447	Melfi	6000.100.147	\
33981353416TN	ED	287W	Pecinci, Serbia	E609	SATA	1000902447	Melfi	6000.101.746	6000.404.959
33981353426TN	ED	287W	Pecinci, Serbia	E609	SATA	1000902447	Melfi	6000.101.746	6099.506.280
0203300140129	AE	078W	Reutlingen, Germany	E609	SATA	1000902447	Melfi	6099.739.075	6000.930.377
02033002311VS	AE	078W	Reutlingen, Germany	E609	SATA	1000902447	Melfi	6099.739.075	6000.930.377
0445110524V13	DS	060W	Bamberg, Germany	E609	FCA italy	1000901413	Pratola Serra	6099.101.121	6000.503.227
0445110682129	DS	060W	Bamberg, Germany	E609	FCA italy	1000901413	Pratola Serra	6099.101.121	6000.503.227

Allegato 2: Matrici di analisi

	ASTRA Piacenza	DENSO Poirino	FCA ITALY Torino	FCA ITALY Villanova	FCA ITALY - VM Cento	FPT INDUSTRIAL Torino	FPT INDUSTRIAL Pregnana	FPT INDUSTRIAL Villanova	IVECO Villanova	IVECO Brescia	IVECO Suzzara	LAMBORGHINI Sant'Agata	LAMBORGHINI Castel Maggiore	NEW HOLLAND S.Mauro	FCA ITALY Pratola	FCA ITALY Pomigliano	FCA ITALY Piedimonte	FPT INDUSTRIAL Foggia	SATA Melfi	SEVEL Atessa	TOTALI	
Ansbach												1										1
Bamberg															3			1				4
Blaichach																				2		2
Blaj						1		2														3
Braga									1	1												2
Bursa					5	1									1			1				8
Castellet												1										1
Eisenach			5	7							1	1				2	2		2	2		22
Feuerbach						1												2				3
Hallein	2						10		4	4				2								22
Hatvan												4	1									5
Jihlava					3													1				4
Llica																				2		2
Madrid			1																			1
Miskolc		4																2	9			15
Pecinci																				6		6
Penang			1	3												1	1			1		7
Reutlingen				3																4		7
Rutesheim				3													2		2	1		8
Tienen				2								4								8		14
TOTALI	2	4	7	18	8	3	10	2	5	5	1	11	1	2	4	3	5	7	34	5		

In azzurro sono evidenziati i flussi che transitano nell'EDL nord, in verde sono evidenziati i codici che transitano nell'EDL sud.

	ASTRA Piacenza	DENSO Poirino	FCA ITALY Torino	FCA ITALY Villanova	FCA ITALY - VM Cento	FPT INDUSTRIAL Torino	FPT INDUSTRIAL Pregnana	FPT INDUSTRIAL Villanova	IVECO Villanova	IVECO Brescia	IVECO Suzzara	LAMBORGHINI Sant'Agata	LAMBORGHINI Castel Maggiore	NEW HOLLAND S.Mauro	FCA ITALY Pratola	FCA ITALY Pomigliano	FCA ITALY Piedimonte	FPT INDUSTRIAL Foggia	SATA Melfi	SEVEL Atessa	TOTALI	
Ansbach												1764									1764	
Bamberg															333240			554400				887640
Blaichach																				104640		104640
Blaj						21240		2880														24120
Braga									64	3972												4036
Bursa					388656	4800									81864			50400				525720
Castellet												3900										3900
Eisenach			70118	508540							36848	5544				86908	60656		100812	138716		1008142
Feuerbach						7232												13056				20288
Hallein	516						25172		4096	19028				212								49024
Hatvan												6254	2									6256
Jihlava					132960													90760				223720
Llica																				77580		77580
Madrid			60588																			60588
Miskolc		115243																26051	93300			234594
Pecinci																				1650		1650
Penang			300	25320												120	30		23550			49320
Reutlingen				5129																15070		20199
Rutesheim				9360													180		35062	0		44602
Tienen				12000								2580							73520			88100
TOTALI	516	115243	131006	560349	521616	33272	25172	2880	4160	23000	36848	20042	2	212	415104	87028	60866	734667	420544	243356		

In azzurro sono evidenziati i flussi che transitano nell'EDL nord, in verde sono evidenziati i codici che transitano nell'EDL sud.

Allegato 4: Milk run di raccolta presso i clienti

ROUTE	PERCORSO	km
ROUTE 108	SETTIMO T.SE	90
	Poirino	
	Villanova	
	S. Mauro	
	Torino	
	SETTIMO T.SE	
ROUTE 208	SETTIMO T.SE	766
	Piacenza	
	Sant'Agata	
	Castel maggiore	
	Cento	
	Suzzara	
	SETTIMO T.SE	
ROUTE 308	SETTIMO T.SE	450
	Brescia	
	Pregnana	
	SETTIMO T.SE	
ROUTE 109	MADDALONI	392
	Melfi	
	Foggia	
	MADDALONI	
ROUTE 209	MADDALONI	389
	Piedimonte	
	Atessa	
	MADDALONI	
ROUTE 309	MADDALONI	146
	Pratola	
	Pomigliano	
	MADDALONI	

Allegato 5: Milk run di consegna presso gli stabilimenti Bosch

ROUTE	PERCORSO	km da E608
ROVEST 1	Castellet	4048
	Llica	
	Madrid	
	Braga	
RCENT 1	Hallein	2157
	Jihlava	
RCENT 2	Reutlingen	2202
	Rutesheim	
	Feuerbach	
	Tienen	
RCENT 3	Ansbach	1862
	Bamberg	
	Eisenach	
	Blaichach	
REST1	Hatvan	2562
	Miskolc	
REST 2	Pecinci	3173
	Blaj	

Allegato 6: Risultati test algoritmo

Instance	Optimal value	npop	ngen	Best found genetic	Inserted items	Computation time[s]	Error wrt opt[%]
A1-n100-R100	3720	100	50	3715	56	1,989	0,13
		500	100	3715	56	20,607	0,13
A1-n100-R1000	39718	100	50	39625	57	2,015	0,23
		500	100	39698	56	20,655	0,05
A1-n1000-R100	39549	100	50	39451	582	215,776	0,25
		500	100	39450	582	2166,668	0,25
A1-n1000-R1000	392539	100	50	391380	581	209,983	0,30
		500	100	391671	584	2126,502	0,22
A2-n100-R100	2702	100	50	2625	47	2,187	2,85
		500	100	2636	48	21,731	2,44
A2-n100-R1000	28385	100	50	27415	50	2,2	3,42
		500	100	27773	50	21,2	2,16
A2-n1000-R100	27608	100	50	26157	436	222,509	5,26
		500	100	26240	445	2222,859	4,96
A2-n1000-R1000	277398	100	50	262085	453	219,285	5,52
		500	100	262854	464	2191,769	5,24
A3-n100-R100	2702	100	50	2689	51	2,158	0,48
		500	100	2694	51	21,418	0,30
A3-n100-R1000	28385	100	50	28271	52	2,096	0,40
		500	100	28243	52	21,185	0,50
A3-n1000-R100	27606	100	50	27413	522	221,843	0,70
		500	100	27418	526	2251,687	0,68
A3-n1000-R1000	277565	100	50	275933	533	217,763	0,59
		500	100	275995	537	2187,978	0,57

Allegato 7: Codice algoritmo genetico

main.c

```
#include<stdio.h>
#include<stdlib.h>
#include<malloc.h>
#include<math.h>
#include<time.h>

#include"lettura.h"
#include"gen_casuale.h"
#include"mutazione.h"
#include"crossover.h"
#include"residui.h"
#include"improve.h"
#include"selezione.h"

    int n, C, D, npop, ngen, maxprof; //n°oggetti da inserire, capacità di peso, capacità di
volume, numerosità popolazione, numero generazioni genetico e massimo profitto ottenibile
dalla specifica istanza
    int *p, *w, *v, **s, *C_res, *D_res, **sm, **sc, *C_resm, *D_resm, *C_resc, *D_resc,
**st; //array dei profitti, pesi e volumi, matrice delle soluzioni scritte per riga, capacità residue di
peso e volume, matrice delle soluzioni mutate e matrice delle soluzioni figli ottenute da
crossover
    int *score; //punteggio per la selezione

    int lettura(); //funzione per la lettura del file di input
    int gen_casuale (int n, int npop); //funzione per la generazione casuale di soluzioni
    int mutazione (int n, int npop, int **s, int l); //funzione per la mutazione
    int crossover (int n, int npop, int **s, int **sm); //fuznione per il crossover
    int residui (int n, int npop, int C, int D, int *w, int *v, int **s, int **sm, int **sc);
//funzione per il calcolo delle capacità residue
    int improve (int n, int npop, int C, int D, int *C_res, int *D_res, int *C_resm, int
*D_resm, int *C_resc, int *D_resc, int **s, int **sm, int **sc, int *p, int *w, int *v);
    int selezione (int n, int npop, int **s, int **sm, int **sc, int *p, int maxprof); //funzione
per la selezione

int main()
{
    int i; //indice cicli for stampa/lettura, indice di righe
    int j; //indice cicli for stampa/lettura, indice di colonne
    int l; //indice generazioni genetico

    /* chiamata alla funzione che legge e stampa le informazioni contenute nel file di input */
    lettura();
```

```

/* assegna ad ogni elemento di una soluzione un valore intero random in [0,1] */
gen_casuale(n, npop);

for (l = 0; l < ngen; l++){

    //printf("Generazione %d\n", l+1);

    /* mutazione */
    //printf("Mutazione\n");
    mutazione(n, npop, s, l);

    /* crossover */
    //printf("Crossover\n");
    crossover (n, npop, s, sm);

    /* calcola le capacità residue */
    //printf("Calcolo residui\n");
    residui(n, npop, C, D, w, v, s, sm, sc);

    /* ripristino ammissibilità e migliora il valore delle soluzioni cercando di inserire
items se possibile */
    //printf("Improve\n");
    improve (n, npop, C, D, C_res, D_res, C_resm, D_resm, C_resc, D_resc, s, sm, sc,
p, w, v);

    /* selezione */
    //printf("Selezione\n\n");
    selezione (n, npop, s, sm, sc, p, maxprof);
}

/* inizializza a 0 tutti gli score della popolazione finale */
for (i = 0; i < npop; i++){
    score[i] = 0;
}

/* calcola gli score delle soluzioni della popolazione finale */
for (i = 0; i < npop; i++){
    for (j = 0; j < n; j++){
        score[i] = score[i] + p[j] * s[i][j];
    }
}

/* printf("Score pop finale\n");
for (i = 0; i < (npop); i++){
    printf("%d\t", score[i]);
}

```

```

printf("\n\n");
*/
/* trova il massimo score tra le soluzioni della popolazione finale */
int max = 0; //variabile a cui è assegnato l'indice a cui corrisponde l'elemento massimo
dell'array score
for (i = 0; i < npop; i++){
    if (score[i] > score[max]){
        max = i;
    }
}

/* stampa la migliore soluzione trovata */
printf("Best found solution:\n");
for(j = 0; j < n; j++){
    printf("%d\t", s[max][j]);
}
printf("\n\n");

/* stampa il dettaglio degli elementi che vengono selezionati */
printf("Details of the selected items:\n");
for (j = 0; j < n; j++){
    if (s[max][j] == 1){
        printf("%d\t", j+1);
    }
}
printf("\n");

/* stampa la migliore soluzione trovata */
printf("Value of the best found solution:\n>>>>>>%d\n\n", score[max]);

/* stampa il numero di items che vengono selezionati dalla soluzione migliore */
int count = 0; //contatore items inseriti dalla soluzione migliore
printf("Number of selected items:\n");
for (j = 0; j < n; j++){
    if(s[max][j] == 1){
        count++;
    }
}
printf(">>>>>>%d\n\n", count);

return 0;

} //fine main

```

lettura.h

```
#include<stdio.h>
#include<stdlib.h>
#include<malloc.h>
#include<math.h>

extern int *p, *w, *v, **s, *C_res, *D_res, *C_resm, *D_resm, *C_resc, *D_resc,
maxprof, **sm, **sc, **st; //array dei profitti, pesi, volumi, soluzioni iniziali e capacità residue
di peso e volume
extern int *score;

int lettura ()
{
    FILE *fp_input; //puntatore al file input
    FILE *fp_parametri; //puntatore al file di input dei parametri
    int i, j; //indice elementi degli array nei cicli for di lettura e stampa, indice di righe e di
colonne
    extern int n, C, D; //n°oggetti da inserire, capacità di peso, capacità di volume,
    extern int npop, ngen; //numerosità popolazione iniziale, numero di generaz del genetico
sono parametri dell'algoritmo

    /* apre il file "input.txt" in lettura */
    fp_input = fopen("input.txt", "r");

    /* apre il file "parametri.txt" in lettura */
    fp_parametri = fopen("parametri.txt", "r");

    /* gestisce un errore in fase di apertura del file "input.txt" */
    if((fp_input) == NULL) {
        printf("Errore in apertura del file input.txt.\n");
        return 1;
    }

    /* gestisce un errore in fase di apertura del file "parametri.txt" */
    if((fp_parametri) == NULL) {
        printf("Errore in apertura del file parametri.txt.\n");
        return 1;
    }

    /* legge il file dei parametri: numerosità popolazione iniziale e generazioni genetico */
    if (!(fp_parametri) == NULL){

        /* legge numerosità popolazione iniziale e generazioni genetico */
        fscanf(fp_parametri, "%d\n %d\n", &npop, &ngen);
    }
}
```

```

/* alloca la memoria per gli array delle capacità residue e per gli score */
C_res = (int *)malloc(npop * sizeof(int));
D_res = (int *)malloc(npop * sizeof(int));

C_resm = (int *)malloc(npop * sizeof(int));
D_resm = (int *)malloc(npop * sizeof(int));

C_resc = (int *)malloc(npop * sizeof(int));
D_resc = (int *)malloc(npop * sizeof(int));

score = (int *)malloc((3 * npop) * sizeof(int));
}

/* legge il file di input: n°items e capacità contenitore in termini di peso e volume e gli
array p, w, v */
if (!(fp_input) == NULL){

    /* legge i valori di: n°items, capacità di peso, capacità di volume,
    numerosità popolazione iniziale */
    fscanf(fp_input, "%d\n %d\n %d\n", &n, &C, &D);

    /* alloca la memoria per gli array di p, w, v */
    p = (int *)malloc((n+1) * sizeof(int));
    w = (int *)malloc((n+1) * sizeof(int));
    v = (int *)malloc((n+1) * sizeof(int));

    /* alloca la memoria per la matrice delle soluzioni iniziali */
    s = (int **)malloc(npop * sizeof(int *));
    for (i = 0; i < npop; i++){
        s[i] = (int *)malloc(n * sizeof(int *));
    }

    /* alloca la memoria per la matrice delle soluzioni mutate */
    sm = (int **)malloc(npop * sizeof(int *));
    for (i = 0; i < npop; i++){
        sm[i] = (int *)malloc(n * sizeof(int *));
    }

    /* alloca la memoria per la matrice delle soluzioni figli */
    sc = (int **)malloc(npop * sizeof(int *));
    for (i = 0; i < npop; i++){
        sc[i] = (int *)malloc(n * sizeof(int *));
    }
}

```

```

/* alloca la memoria per la matrice della popolazione di soluzioni per la
selezione */
st = (int **)malloc((3 * npop) * sizeof(int *));
for (i = 0; i < (3 * npop); i++){
    st[i] = (int *)malloc(n * sizeof(int *));
}

/* legge gli array di: profitti, pesi e volumi */
for (i = 0; i < n; i++){
    fscanf(fp_input, "%d", &p[i]);
}

for (i = 0; i < n; i++){
    fscanf(fp_input, "%d", &w[i]);
}

for (i = 0; i < n; i++){
    fscanf(fp_input, "%d", &v[i]);
}
}

/* chiude il file aperto in lettura */
fclose (fp_input);

/* stampa n, C, D, npop, ngen per verificare che la funzione lettura abbia restituito
correttamente tutti i parametri */
printf("Numero di items: %d\n\nCapacita di peso: %d\n\nCapacita di volume: %d\n\n", n,
C, D);
printf("Numerosita popolazione iniziale: %d\n\nGenerazioni genetico: %d\n\n", npop,
ngen);

/* stampa p, w, v per verificare che la funzione lettura abbia restituito correttamente tutti i
parametri */
for (i = 0; i < n; i++){
    printf("item_%d p_%d: %5d\tw_%d: %5d\tv_%d: %5d\n", i+1, i+1, p[i], i+1,
w[i], i+1, v[i]);
}
printf("\n\n");

/* calcola il massimo profitto possibile per la specifica istanza, ottenuto inserendo tutti gli
oggetti nel contenitore */
for (j = 0; j < n; j++){
    maxprof += p[j];
}

```

```

    /* stampa il massimo profitto possibile per la specifica istanza */
    printf("Il massimo profitto (UB) ottenibile per la specifica istanza e: %d\n\n", maxprof);

    /* ritorna alla funzione main i parametri */
    return (n, C, D, npop, ngen, *p, *w, *v, maxprof);
}

```

gen_casuale.h

```

#include<stdio.h>
#include<stdlib.h>
#include<malloc.h>
#include<math.h>

```

```

int gen_casuale(int n, int npop)

```

```

{
    int i, j; //indice di righe e di colonne

    /* assegna ad ogni elemento di una soluzione un valore intero random in [0,1] */
    for (i = 0; i < npop; i++) {
        srand(i);
        for (j = 0; j < n; j++){
            s[i][j] = rand()%2;
        }
    }

    /* stampa la popolazione iniziale generata random, ogni riga identifica una soluzione */
    /* printf("Popolazione iniziale generata random\n\n");
    for (i = 0; i < npop; i++) {
        printf("Soluzione %d\n", i+1);
        for (j = 0; j < n; j++){
            printf("%d\t", s[i][j]);
        }
        printf("\n\n");
    }
    printf("\n\n");
    */
    return(**s);
}

```


mutazione.h

```
#include<stdio.h>
#include<stdlib.h>
#include<malloc.h>
#include<math.h>

int mutazione(int n, int npop, int **s, int l)
{
    int i, j; //indici di riga e di colonna
    int *m = (int *)malloc((0.20 * n) * sizeof(int));

    /* inizializza sm a 0 */
    for (i = 0; i < npop; i++){
        for (j = 0; j < n; j++){
            sm[i][j] = 0;
        }
    }

    /* mutazione */
    for (i = 0; i < npop; i++) {
        srand((l + 1) * ((i + 1)));
        for (j = 0; j < (0.20 * n); j++){
            m[j] = rand()%n;
            if (s[i][m[j]] == 0){
                sm[i][m[j]] = 1;
            }
            if (s[i][m[j]] == 1){
                sm[i][m[j]] = 0;
            }
        }
    }

    /* stampa la popolazione mutata */
    /* printf("Popolazione iniziale mutata\n\n");
    for (i = 0; i < npop; i++) {
        printf("Soluzione %d\n", i+1);
        for (j = 0; j < n; j++){
            printf("%d\t", sm[i][j]);
        }
        printf("\n\n");
    }
    printf("\n\n");
*/
    return (**sm);
}
```

crossover.h

```
#include<stdio.h>
#include<stdlib.h>
#include<malloc.h>
#include<math.h>

int crossover (int n, int npop, int **s, int **sm)
{
    int i, j;

    /* crossover 1 cut point */
    for (i = 0; i < npop; i++){
        for (j = 0; j < n; j++){
            if (j <= (n/5)){
                sc[i][j] = sm[i][j]; //il secondo terzo è = alla soluzione della pop
iniziale
            }
            if (j > (n/5)){
                sc[i][j] = s[i][j]; //il terzo terzo è = alla soluzione mutata
            }
        }
    }

    /* stampa la matrice delle soluzioni figli */
    /* printf("Popolazione di figli\n\n");
    for (i = 0; i < npop; i++){
        printf("Figlio %d\n", i+1);
        for (j = 0; j < n; j++){
            printf("%d\t", sc[i][j]);
        }
        printf("\n\n");
    }
    printf("\n\n");
    */
    return (**sc);
}
```

residui.h

```
#include<stdio.h>
#include<stdlib.h>
#include<malloc.h>
#include<math.h>
```

```
extern int *C_res, *D_res, *C_resm, *D_resm, *C_resc, *D_resc;
```

```

int residui (int n, int npop, int C, int D, int *w, int *v, int **s, int **sm, int **sc)
{
    int i, j; //indici di riga e di colonna

    /* inizializza a C e D le capacità residue */
    for (i = 0; i < npop; i++){
        C_res[i] = C;
        D_res[i] = D;

        C_resm[i] = C;
        D_resm[i] = D;

        C_resc[i] = C;
        D_resc[i] = D;
    }

    /* sottrae le corrispondenti capacità se nelle soluzioni s il valore di un item è 1 */
    for (i = 0; i < npop; i++){
        for (j = 0; j < n; j++){
            if (s[i][j] == 1){
                C_res[i] = C_res[i] - w[j];
                D_res[i] = D_res[i] - v[j];
            }
        }
    }

    /* sottrae le corrispondenti capacità se nelle soluzioni sm il valore di un item è 1 */
    for (i = 0; i < npop; i++){
        for (j = 0; j < n; j++){
            if (sm[i][j] == 1){
                C_resm[i] = C_resm[i] - w[j];
                D_resm[i] = D_resm[i] - v[j];
            }
        }
    }

    /* sottrae le corrispondenti capacità se nelle soluzioni sc il valore di un item è 1 */
    for (i = 0; i < npop; i++){
        for (j = 0; j < n; j++){
            if (sc[i][j] == 1){
                C_resc[i] = C_resc[i] - w[j];
                D_resc[i] = D_resc[i] - v[j];
            }
        }
    }
}

```

```

    return (*C_res, *D_res, *C_resm, *D_resm, *C_resc, *D_resc);
}

```

improve.h

```

#include<stdio.h>
#include<stdlib.h>
#include<malloc.h>
#include<math.h>

```

```

int improve (int n, int npop, int C, int D, int *C_res, int *D_res, int *C_resm, int *D_resm, int
*C_resc, int *D_resc, int **s, int **sm, int **sc, int *p, int *w, int *v)

```

```

{
    int i, j; //indici di riga e di colonna
    int min, max, k; //variabili utilizzate nella ricerca del profitto minore e maggiore
    double *rate = (double *)malloc(n * sizeof(double));

    /* calcola il rate di ogni item */
    for (j = 0; j < n; j++){
        rate[j] = (((double)p[j]/(w[j] + v[j])));
    }

    /* rimuovi item da s finchè la capacità residua è negativa risettando a 0 */
    for (i = 0; i < npop; i++){
        if (C_res[i] < 0 || D_res[i] < 0){
            //cerca l'item con minimo profitto e lo mette a 0 se è a 1 in s
            while (C_res[i] < 0 || D_res[i] < 0){
                min = 0;
                for (k = 0; k < n; k++){
                    if (rate[k] <= rate[min] && s[i][k] == 1){
                        min = k;
                    }
                }
                rate[min] = 999999;
                if (s[i][min] == 1){
                    s[i][min] = 0;
                    C_res[i] = C_res[i] + w[min];
                    D_res[i] = D_res[i] + v[min];
                }
            }
        }
    }
    /* calcola il rate di ogni item */
    for (j = 0; j < n; j++){
        rate[j] = (((double)p[j]/(w[j] + v[j])));
    }
}

```

```

}

/* rimuovi item da sm finchè la capacità residua è negativa risettando a 0 */
for (i = 0; i < npop; i++){
    if (C_resm[i] < 0 || D_resm[i] < 0){
        //cerca l'item con minimo profitto e lo mette a 0 se è a 1 in s
        while (C_resm[i] < 0 || D_resm[i] < 0){
            min = 0;
            for (k = 0; k < n; k++){
                if (rate[k] <= rate[min] && sm[i][k] == 1){
                    min = k;
                }
            }
            rate[min] = 999999;
            if (sm[i][min] == 1){
                sm[i][min] = 0;
                C_resm[i] = C_resm[i] + w[min];
                D_resm[i] = D_resm[i] + v[min];
            }
        }
    }
}

/* calcola il rate di ogni item */
for (j = 0; j < n; j++){
    rate[j] = (((double)p[j]/(w[j] + v[j])));
}
}

```

```

/* rimuovi item da sc finchè la capacità residua è negativa risettando a 0 */
for (i = 0; i < npop; i++){
    if (C_resc[i] < 0 || D_resc[i] < 0){
        //cerca l'item con minimo profitto e lo mette a 0 se è a 1 in s
        while (C_resc[i] < 0 || D_resc[i] < 0){
            min = 0;
            for (k = 0; k < n; k++){
                if (rate[k] <= rate[min] && sc[i][k] == 1){
                    min = k;
                }
            }
            rate[min] = 999999;
            if (sc[i][min] == 1){
                sc[i][min] = 0;
                C_resc[i] = C_resc[i] + w[min];
                D_resc[i] = D_resc[i] + v[min];
            }
        }
    }
}

```

```

    }
    /* calcola il rate di ogni item */
    for (j = 0; j < n; j++){
        rate[j] = (((double)p[j]/(w[j] + v[j])));
    }
}

/* cerca di inserire item in s se ci stanno */
for (i = 0; i < npop; i++){
    for (j = 0; j < n; j++){
        //prova a inserire gli items con profitto maggiore finchè ci stanno
        max = 0;
        for (k = 0; k < n; k++){
            if (rate[k] >= rate[max] && s[i][k] == 0){
                max = k;
            }
        }
        rate[max] = 0;
        if (w[max] <= C_res[i] && v[max] <= D_res[i] && s[i][max] == 0){
            s[i][max] = 1;
            C_res[i] = C_res[i] - w[max];
            D_res[i] = D_res[i] - v[max];
        }
    }
    /* calcola il rate di ogni item */
    for (j = 0; j < n; j++){
        rate[j] = (((double)p[j]/(w[j] + v[j])));
    }
}

/* cerca di inserire item in sm se ci stanno */
for (i = 0; i < npop; i++){
    for (j = 0; j < n; j++){
        //prova a inserire gli items con profitto maggiore finchè ci stanno
        max = 0;
        for (k = 0; k < n; k++){
            if (rate[k] >= rate[max] && sm[i][k] == 0){
                max = k;
            }
        }
        rate[max] = 0;
        if (w[max] <= C_resm[i] && v[max] <= D_resm[i] && sm[i][max] ==
0){
            sm[i][max] = 1;
            C_resm[i] = C_resm[i] - w[max];

```

```

        D_resm[i] = D_resm[i] - v[max];
    }
}
/* calcola il rate di ogni item */
for (j = 0; j < n; j++){
    rate[j] = (((double)p[j]/(w[j] + v[j])));
}
}

/* cerca di inserire item in sc se ci stanno */
for (i = 0; i < npop; i++){
    for (j = 0; j < n; j++){
        //prova a inserire gli items con profitto maggiore finchè ci stanno
        max = 0;
        for (k = 0; k < n; k++){
            if (rate[k] >= rate[max] && sc[i][k] == 0){
                max = k;
            }
        }
        rate[max] = 0;
        if (w[max] <= C_resc[i] && v[max] <= D_resc[i] && sc[i][max] == 0){
            sc[i][max] = 1;
            C_resc[i] = C_resc[i] - w[max];
            D_resc[i] = D_resc[i] - v[max];
        }
    }
}
/* calcola il rate di ogni item */
for (j = 0; j < n; j++){
    rate[j] = (((double)p[j]/(w[j] + v[j])));
}
}

```

//INIZIO LOCAL SEARCH

/* cerca se ci sono elementi non inseriti in s che hanno profitto maggiore di quelli inseriti e pesi minori */

```

for (i = 0; i < npop; i++){
    for (j = 0; j < n; j++){
        if (s[i][j] == 0){
            for (k = 0; k < n; k++){
                if (s[i][k] == 1 && p[k] < p[j] && w[k] >= w[j] && v[k] >=
v[j]){
                    //if (w[j] > C_res[i] && v[j] > D_res[i]){
                        s[i][j] = 1;
                        s[i][k] = 0;
                        C_res[i] = C_res[i] - w[j] + w[k];
                    }
                }
            }
        }
    }
}

```

```

        D_res[i] = D_res[i] - v[j] + v[k];
    /*}
else if (w[j] <= C_res[i] && v[j] <= D_res[i]){
    s[i][j] = 1;
    C_res[i] = C_res[i] - w[j] + w[k];
    D_res[i] = D_res[i] - v[j] + v[k];
}*/
    }
}
}
}
}

/* cerca se ci sono elementi non inseriti in sm che hanno profitto maggiore di quelli
inseriti e pesi minori */
for (i = 0; i < npop; i++){
    for (j = 0; j < n; j++){
        if (sm[i][j] == 0){
            for (k = 0; k < n; k++){
                if(sm[i][k] == 1 && p[k] < p[j] && w[k] >= w[j] && v[k]
>= v[j]){
                    //if (w[j] > C_resm[i] && v[j] > D_resm[i]){
                        sm[i][j] = 1;
                        sm[i][k] = 0;
                        C_resm[i] = C_resm[i] - w[j] + w[k];
                        D_resm[i] = D_resm[i] - v[j] + v[k];
                    /*}
else if (w[j] <= C_resm[i] && v[j] <= D_resm[i]){
                        sm[i][j] = 1;
                        C_resm[i] = C_resm[i] - w[j];
                        D_resm[i] = D_resm[i] - v[j];
                    }*/
                }
            }
        }
    }
}

```

```

/* cerca se ci sono elementi non inseriti in sc che hanno profitto maggiore di quelli
inseriti e pesi minori */
for (i = 0; i < npop; i++){
    for (j = 0; j < n; j++){
        if (sc[i][j] == 0){
            for (k = 0; k < n; k++){

```



```

if(sc[i][k] == 1 && p[k] < p[j] && w[k] >= w[j] && v[k]
>= v[j]){
    //if (w[j] > C_resc[i] && v[j] > D_resc[i]){
        sc[i][j] = 1;
        sc[i][k] = 0;
        C_resc[i] = C_resc[i] - w[j] + w[k];
        D_resc[i] = D_resc[i] - v[j] + v[k];
    /*}
    else if (w[j] <= C_resc[i] && v[j] <= D_resc[i]){
        sc[i][j] = 1;
        sc[i][k] = 1;
        C_resc[i] = C_resc[i] - w[j];
        D_resc[i] = D_resc[i] - v[j];
    }*/
}
}
}
}
}
}
//FINE LOCAL SEARCH

/* cerca di inserire item in s se ci stanno */
for (i = 0; i < npop; i++){
    for (j = 0; j < n; j++){
        //prova a inserire gli items con profitto maggiore finchè ci stanno
        max = 0;
        for (k = 0; k < n; k++){
            if (rate[k] >= rate[max] && s[i][k] == 0){
                max = k;
            }
        }
        rate[max] = 0;
        if (w[max] <= C_res[i] && v[max] <= D_res[i] && s[i][max] == 0){
            s[i][max] = 1;
            C_res[i] = C_res[i] - w[max];
            D_res[i] = D_res[i] - v[max];
        }
    }
    /* calcola il rate di ogni item */
    for (j = 0; j < n; j++){
        rate[j] = (((double)p[j]/(w[j] + v[j])));
    }
}

/* cerca di inserire item in sm se ci stanno */

```

```

for (i = 0; i < npop; i++){
    for (j = 0; j < n; j++){
        //prova a inserire gli items con profitto maggiore finchè ci stanno
        max = 0;
        for (k = 0; k < n; k++){
            if (rate[k] >= rate[max] && sm[i][k] == 0){
                max = k;
            }
        }
        rate[max] = 0;
        if (w[max] <= C_resm[i] && v[max] <= D_resm[i] && sm[i][max] ==
0){
            sm[i][max] = 1;
            C_resm[i] = C_resm[i] - w[max];
            D_resm[i] = D_resm[i] - v[max];
        }
    }
    /* calcola il rate di ogni item */
    for (j = 0; j < n; j++){
        rate[j] = (((double)p[j]/(w[j] + v[j])));
    }
}

/* cerca di inserire item in sc se ci stanno */
for (i = 0; i < npop; i++){
    for (j = 0; j < n; j++){
        //prova a inserire gli items con profitto maggiore finchè ci stanno
        max = 0;
        for (k = 0; k < n; k++){
            if (rate[k] >= rate[max] && sc[i][k] == 0){
                max = k;
            }
        }
        rate[max] = 0;
        if (w[max] <= C_resc[i] && v[max] <= D_resc[i] && sc[i][max] == 0){
            sc[i][max] = 1;
            C_resc[i] = C_resc[i] - w[max];
            D_resc[i] = D_resc[i] - v[max];
        }
    }
    /* calcola il rate di ogni item */
    for (j = 0; j < n; j++){
        rate[j] = (((double)p[j]/(w[j] + v[j])));
    }
}

```

```

        return (**s, **sm, **sc, *C_res, *D_res, *C_resm, *D_resm, *C_resc, *D_resc);
    }

```

selezione.h

```

#include<stdio.h>
#include<stdlib.h>
#include<malloc.h>
#include<math.h>

```

```

extern int **st;
extern int *score;

```

```

int selezione (int n, int npop, int **s, int **sm, int **sc, int *p, int maxprof)

```

```

{
    int i, j, q; //indici di riga e di colonna

    /* inizializza a 0 tutti gli score di s */
    for (i = 0; i < (3 * npop); i++){
        score[i] = 0;
    }

    /* unisce le tre matrici, la matrice st contiene la popolazione per la selezione ottenuta
    unendo s, sm, sc */
    for (i = 0; i < (3 * npop); i++){
        for (j = 0; j < n; j++){
            if (i < npop){
                st[i][j] = s[i][j];
            }
            else if (i >= npop && i < (2 * npop)){
                st[i][j] = sm[i - npop][j];
            }
            else if (i >= (2 * npop) && i < (3 * npop)){
                st[i][j] = sc[i - (2 * npop)][j];
            }
        }
    }

    /* stampa la popolazione a cui viene applicata la selezione */
    printf("Popolazione per la selezione\n");
    for (i = 0; i < (3 * npop); i++){
        for (j = 0; j < n; j++){
            printf("%d\t", st[i][j]);
        }
    }
}

```

```

printf("\n\n");
*/
/* calcola i punteggi di tutte le soluzioni della popolazione in st */
for (i = 0; i < (3 * npop); i++){
    for (j = 0; j < n; j++){
        score[i] = score[i] + p[j] * st[i][j];
    }
}

/* stampa i punteggi delle soluzioni di st */
/* printf("Score di tutte le soluzioni\n");
for (i = 0; i < (3 * npop); i++){
    printf("%d\t", score[i]);
}
*/

/* individua le migliori npop soluzioni in base allo score e le fa diventare la popolazione
iniziale per la generazione successiva */
int max = 0;
for (i = 0; i < npop; i++){
    for (j = 0; j < (3 * npop); j++){
        if (score[j] > score[max]){
            max = j;
        }
    }
    //printf("\n");
    //printf("%d", score[max]);
    score[max] = 0;
    for (q = 0; q < n; q++){
        s[i][q] = st[max][q];
    }
}

/* printf ("\n\n");
printf("Popolazione iniziale per la successiva generazione\n");
for (i = 0; i < npop; i++){
    for (j = 0; j < n; j++){
        printf("%d\t", s[i][j]);
    }
}
printf("\n\n");
*/
return (**s);
}

```

BIBLIOGRAFIA E SITOGRAFIA

- [1] <http://bzo.bosch.com/>.
- [2] Chism, Camille, "Optimizing and Benchmarking Returnable Container Processes within an Automotive Distribution System" (2010). Thesis. Rochester Institute of Technology.
- [3] Mark Gardner, Mark Abassi, Debanjan Dutt, Yogesh Goswami. Automotive Supply Chain: Unlocking potential cost savings in automotive packaging.
- [4] European Parliament and Council Directive 94/62/EC, of 20. Dec. 1994 on packaging and packaging waste, Off. Journal of the European Communities, No. L 365/10 ff. Brüssel, 31.12.1994.
- [5] <http://www.automotivelogisticsmagazine.com/intelligence/to-own-or-not-to-own>.
- [6] Dario Scudellaro, 2010. Supply Network Analysis: Studio Dei Flussi Logistici Di Ricircolo. Master Thesis.
- [7] Stefano Allesina, Anna Azzi, Daria Battini, Alberto Regattieri. Performance Measurement in Supply Chain: New Network Analysis and Entropic Indexes.. International Journal of Production Research, Taylor & Francis: STM, Behavioural Science and Public Health Titles, 2010, 48 (08), pp.2297-2321.
- [8] Souza, R.d., Zice, S., Chaoyang, L., 2000. Supply chain dynamics and optimization. Journal of Integrated Manufacturing Systems 11 (5), 348-364.
- [9] Frizelle G. and Woodcock E., 1995. Measuring complexity as an aid to developing operational strategy, International Journal of the Operation and Production, 15/5, 26-39.
- [10] Suwignjo, P., Bititci, U.S., Carrie, A.S., 2000. Quantitative models for performance measurement system. International Journal of Production Economics 64, 231-241.
- [11] Albino V., Izzo C. and Kutzt S., 2002. Input-output models for the analysis of a local/global supply chain, International Journal of Production Economics, 78, 119-131.

- [12] Makui A. and Aryanezhad M. B., 2002. A new method for measuring the static complexity in manufacturing, *Journal of the Operational Research Society*, 54/5, 555-558.
- [13] Ulanowicz R.E, 2004. Quantitative methods for ecological network analysis, *Computational Biology Chemistry*, 28, 321-339.
- [14] Massimo Di Francesco, 2007. *New Optimization Model For Empty Container Management*. PhD Thesis.
- [15] Mohammed Marufuzzaman, Ahmed M. Deif (2010). A dynamic approach to determine the product flow nature in apparel supply chain network. *International Journal of Production Economics* 128, 484-495.
- [16] Romano, P., 2009. How can fluid dynamics help supply chain management? *International Journal of Production Economics* 118 463–472.
- [17] Alessandro Creazza, Fabrizio Dallari, 2010. *Speciale: la logistica automotive*.
- [18] Silvano Martello, Paolo Toth, (2003) An Exact Algorithm for the Two-Constraint 0-1 Knapsack Problem. *Operations Research* 51(5):826-835.