

POLITECNICO DI TORINO

**Corso di Laurea Magistrale
in Ingegneria Informatica**

Tesi di Laurea Magistrale

**Ottimizzazione per Supply Chain nel
settore automobilistico**



Relatore
prof. Guido Perboli

firma del relatore

.....

Candidato
Marco Donnarumma

firma del candidato

.....

A.A. 2017/2018

Indice

Introduzione	5
1 Problema della pianificazione	11
1.1 Il gruppo FCA e il CRF	11
1.2 Il progetto PNet	13
1.2.1 Il problema da risolvere	13
1.2.2 I dati e la loro raccolta	14
1.2.3 L'interfaccia utente	15
1.2.4 Dati in output e relative specifiche	21
2 Teoria di ottimizzazione	25
2.1 Problema di ottimizzazione	25
2.1.1 Programmazione Lineare	26
2.1.2 Programmazione Lineare Intera, Intera Booleana e Mista Intera .	26
2.2 Metodi risolutivi: algoritmo Branch and Bound e simili	27
2.2.1 Algoritmo base	27
2.2.2 Branch and Cut	29
2.2.3 Confronto dei due algoritmi	30
2.3 La modellazione del problema	30
2.3.1 I dati di input e le variabili decisionali	31
2.3.2 I vincoli	33
2.3.3 Il modello finale	36
3 Ottimizzazione ed enumerazione	39
3.1 Approcci risolutivi	39

3.1.1	Primo approccio risolutivo	40
3.1.2	Secondo approccio risolutivo	41
3.1.3	Terzo approccio risolutivo	42
3.1.4	Quarto approccio risolutivo	44
3.2	Software usati	46
3.2.1	LINGO	46
3.2.2	CPLEX	47
4	Prove computazionali	51
4.1	Descrizione della metodologia adottata	51
4.1.1	Approccio risolutivo adottato	52
4.1.2	Misure delle prestazioni in LINGO	52
4.1.3	Misure delle prestazioni in CPLEX	53
4.2	Risultati della prima istanza di dati	54
4.2.1	Tempi di esecuzione	54
4.2.2	Memoria	55
4.3	Risultati della seconda istanza di dati	56
4.3.1	Tempi di esecuzione	56
4.3.2	Memoria	57
4.4	Risultati della terza istanza di dati	58
4.4.1	Tempi di esecuzione	58
4.4.2	Memoria	59
4.5	Risultati della quarta istanza di dati	59
4.5.1	Tempi di esecuzione	60
4.5.2	Memoria	61
4.6	Precisione dei software	61
4.6.1	Risultati	62
4.7	Analisi dei risultati	65
4.7.1	Prima istanza	65
4.7.2	Seconda istanza	66
4.7.3	Terza istanza	66
4.7.4	Quarta istanza	67
4.7.5	Sintesi	68

<i>INDICE</i>	5
Conclusioni	70

Introduzione

Una Supply Chain, in Italiano catena di distribuzione, è un sistema di organizzazioni, persone, attività, informazioni e risorse coinvolte nel processo atto a trasferire o fornire un prodotto o un servizio dal fornitore al cliente.

Volendo fare un esempio generale(senza entrare nel dettaglio) nel settore degli autoveicoli, tale processo inizia con il recupero delle materie prime(come ferro, carbone, alluminio ecc), sottoposte a una serie lavorazioni per ottenere delle componenti(portiere, albero motore ed altro) da montare insieme per ottenere i diversi veicoli che devono essere distribuiti tra le varie concessionarie.

Com'è possibile intuire, l'intero iter è diviso in vari step, e in ogni step sono coinvolte diverse figure professionali. Una delle risorse cardine della Supply Chain, è il Supply Chain Manager(SCM), cioè la persona responsabile del coordinamento delle varie fasi della catena in modo da migliorare le prestazioni e l'efficienza dell'intero flusso di approvvigionamento risorse e stoccaggio di prodotti finiti. Le attività previste per questa mansione sono:

- elaborare, coordinare e controllare gli aspetti strategici e operativi della catena di distribuzione;
- gestire le tempistiche nella distribuzione delle risorse e dei prodotti;
- curare i costi di approvvigionamento, produzione, scorte, trasporto, tasse;
- operare trasversalmente in maniera coordinata ed integrata tra le aree della allocazione geografica, produzione, gestione inventari, trasporti e distribuzione;
- sovrintendere alle fasi che accompagnano i beni ed i servizi dall'origine al consumo finale.

Ritornando all'esempio citato sopra, si può osservare che la Supply Chain per la vendita e distribuzione delle automobili risulterà di per sé molto complessa, in quanto verranno coinvolti diversi attori ognuno dei quali può ricoprire sia il ruolo di fornitore che di cliente: un impianto che produce portiere ha bisogno di essere rifornito di tutti quei materiali necessari per la produzione delle porte da altri impianti preposti per lo scopo, in modo da poter rifornire le linee di assemblaggio dei veicoli col numero di porte richiesto. Se le diverse organizzazioni appartengono allo stesso gruppo, sarà compito del SCM programmare la produzione e le attività dei vari stabilimenti, oltre a occuparsi del coordinamento logistico per il trasporto dei beni prodotti in ciascuna fase.

Uno dei problemi fondamentali nella gestione della Supply Chain, è appunto la pianificazione della produzione del prodotto per poter soddisfare la domanda del mercato. Per pianificazione della produzione s'intende il processo che stabilisce ed impegna l'ammontare delle risorse (manodopera, attrezzature, macchinari, servizi generali d'impianto ed altro ancora) di cui l'azienda avrà bisogno per le sue attività produttive future e allocazione di tali risorse per ottenere il prodotto desiderato nelle quantità stimate, nel tempo giusto, al posto giusto ed al minore costo totale possibile.

Al giorno d'oggi ricorrendo alle analisi di mercato, la previsione della domanda che si riesce a effettuare è solo una stima di tipo statistico con un ampio margine di incertezza, avente un orizzonte temporale che va da tre a dodici mesi. Riuscire ad individuarla in maniera esatta è impossibile, mentre averne una stima con bassissima incertezza è molto difficile (è quasi divinazione). I bisogni dei consumatori si modificano, difatti, con crescente intensità, inducendo elementi di varietà e variabilità nelle produzioni che difficilmente si conciliano con sistemi industriali, benché efficienti, ma caratterizzati da scarse flessibilità e capacità di reazione. Da sempre il governo dei sistemi produttivi si è trovato a dover coniugare l'inerzia di certe strutture con il dinamismo del mercato. Data l'imprevedibilità della domanda, il problema della pianificazione può essere risolto attraverso la versatilità e la tempestività della risposta, che sia la migliore possibile.

Il SCM avrà bisogno, quindi, di strumenti capaci di trattare tutte le informazioni che coinvolgono l'attività della pianificazione, al fine di fornire un supporto alle decisioni durante questa fase molto delicata. Per assistere un SCM, si ricorre:

- alla Business Intelligence che è l'insieme dei processi e delle tecnologie capaci di raccogliere ed analizzare le informazioni strategiche, al fine di fornire una conoscenza precisa, aggiornata e significativa del contesto di riferimento;

- alla Ricerca Operativa che "è l'applicazione dei metodi scientifici per la soluzione di problemi complessi che nascono nella direzione e gestione di grandi sistemi di uomini, macchine, materiali e denaro nell'industria, affari, amministrazione e difesa.[...] Lo scopo consiste nell'aiutare i decisori a determinare scientificamente la loro politica e loro azioni"[1].

Le tecnologie di Business Intelligence usate per la manipolazione di grandi moli di dati (da raccogliere o già raccolti), la relativa normalizzazione ed analisi, che si hanno a disposizione sono le seguenti: strumenti ETL(Extract Transform and Load); Datawarehouse; sistemi OLAP(Online Analytical Processing); sistemi di reportistica; Data Mining; Text Mining; sistemi GIS(Geographic Information Systems) ed altro ancora. Un Decision Support System (DSS) è un sistema software che integra una o più di queste tecnologie di Business Intelligence, permettendo di aumentare l'efficacia dell'analisi.

Una volta ottenuti i dati di sintesi descrittivi del contesto, bisogna utilizzarli al fine di prendere una decisione. In scenari molto complessi, nonostante si abbiano tutti i dati, un manager ha difficoltà a prendere una decisione operativa rapidamente, in quanto si possono avere tantissime soluzioni alternative, e potrebbe scegliere delle soluzioni non ottimali. Per questo motivo interviene la modellazione della Ricerca Operativa, in cui dato un sistema reale se ne formula il relativo modello matematico, che prende in considerazione solo alcuni aspetti del problema considerati importanti. Successivamente su questo modello si costruiscono le soluzioni proprie del modello, attraverso l'uso di tecniche risolutive, cioè particolari algoritmi implementati in un programma di calcolo. Ottenute le soluzioni bisogna passare all'interpretazione di queste confrontandole col sistema reale. Se le soluzioni fornite risultano coerenti col problema che si sta affrontando, allora il modello è ben definito, altrimenti se le soluzioni non risultano applicabili, vuol dire che alcuni aspetti del problema non sono stati presi in considerazione, rendendo invalidi i risultati così ottenuti. Questa obbliga all'esperto a ripetere ciclicamente tutto il processo di modellazione fin quando le soluzioni non risultano soddisfacenti.

Facendo una sintesi, un SCM, durante la pianificazione della produzione, utilizzerà i sistemi DSS per ottenere i dati del contesto ed avere così una stima della domanda, del budget a disposizione, delle capacità produttive degli impianti e dei diversi costi logistici e di produzione, per poi usare a parte un sistema software di ottimizzazione che ha al suo interno il modello matematico e il modulo software risolutore, che gli consente di ottenere le migliori soluzioni possibili.

Scopo della tesi è illustrare l'attività di tirocinio per lo sviluppo di un software di ottimizzazione a supporto delle decisioni da usare nella fase di pianificazione delle allocazioni produttive degli impianti di proprietà del gruppo Fiat Chrysler Automobiles (FCA). Tale software fornirà soluzioni che minimizzino i costi totali sia per il trasporto, sia per la produzione delle componenti necessari per l'assemblaggio dei veicoli. L'attività di tesi è stata svolta all'interno del progetto Production allocation Network (PNet) del Centro Ricerche Fiat S.c.p.a (CRF).

La tesi è strutturata come segue:

- nel capitolo 1 vengono forniti maggiori dettagli relativi all'azienda, al progetto e alle caratteristiche del problema;
- nel capitolo 2 sono presentati gli strumenti teorici della modellazione matematica necessari per la costruzione del modello usato;
- nel capitolo 3 vengono illustrati gli approcci risolutivi al fine di fornire soluzioni utili per l'utente del software;
- nel capitolo 4 sono mostrati tutti i risultati relativi ai vari casi di test computazionali per determinare le prestazioni del software ed effettuare una valutazione del calcolatore necessario per elaborare in maniera performante;
- nel capitolo delle conclusioni viene riportata una sintesi dell'attività di tesi, con una panoramica degli approcci e sviluppi futuri per affrontare una problematica complessa da risolvere: l'incertezza della domanda nella pianificazione.

Questo lavoro di tesi ha permesso di implementare una metodologia di risoluzione dei problemi combinando la Ricerca Operativa con algoritmi che modificano dinamicamente il modello base, fornendo soluzioni sempre diverse e consentendo al SCM di esplorarle avendo un'idea precisa dello scenario corrente che deve affrontare. Il software sviluppato durante il tirocinio ha mostrato grandi prestazioni nei tempi di risoluzione e ottimi risultati negli scenari reali di pianificazione.

Capitolo 1

Problema della pianificazione

In questo capitolo vengono presentati: il gruppo FCA, CRF e il progetto PNet. Vengono forniti i dettagli sulla tipologia di dati trattati, l'interfaccia in cui vengono inseriti, i vincoli del problema e le prime assunzioni del modello

1.1 Il gruppo FCA e il CRF

L'11 luglio 1899 viene firmato l'atto di nascita della "Società anonima Fabbrica Italiana di Automobili - Torino", da subito nota come Fiat : la prima automobile prodotta è la 4 HP, una vettura a motore bicilindrico raffreddato ad acqua che consumava 8 litri di carburante ogni 100 km, ma la produzione vera e propria inizia solo l'anno dopo con "ben" 24 vetture che escono dal primo stabilimento situato nei pressi di Corso Dante a Torino. Da allora è passato più di un secolo, durante il quale Fiat è stata a più riprese presente in modo significativo nella storia italiana e globale: sempre impegnata nelle innovazioni e nella ricerca dell'eccellenza, Fiat si è spesso distinta per essere riuscita a superare i limiti dei tempi, costruendo motori sempre più performanti e vetture sempre più apprezzate in ogni parte del mondo. Tra le protagoniste assolute del miracolo economico degli anni '50 e '60, l'azienda si è distinta anche per essere stata una delle prime ad aver introdotto il modello di catena di montaggio di matrice fordiana all'interno dei propri stabilimenti. Impegnata nella progettazione e costruzione anche di motori navali e per aeroplani, il gruppo ha ottenuto anche diversi riconoscimenti sportivi (da ricordare i titoli costruttori col marchio Ferrari nella Formula 1 e le vittorie nei campionati rallistici con Lancia, entrambe

controllate da Fiat) e svariati premi Auto dell'Anno con modelli quali la 127 (1972), la Uno (1984), la Tipo (1988) e più recentemente la nuova Fiat 500 (2008), vero e proprio successo in campo commerciale. Non è da dimenticare l'interesse per uno sviluppo che fosse allo stesso tempo compatibile con le norme ambientali: negli ultimi sei anni, infatti, il marchio si è imposto come leader per le più basse emissioni di CO₂ in Europa, con una media di appena 119 kg/km.



Figura 1.1: La 4HP, la prima autovettura prodotta dal gruppo FIAT

Una delle realtà più significative all'interno del gruppo è senz'altro rappresentata dal CRF – Centro Ricerche Fiat: con 81 milioni di euro dedicati nell'anno 2016 e 833 dipendenti impiegati in attività di Ricerca e Sviluppo, i 7 centri sparsi in Italia collaborano da oltre 35 anni con università [2], centri di ricerca pubblici e partner industriali, nazionali ed internazionali, progettando e sperimentando ogni giorno soluzioni per anticipare e soddisfare le necessità dei clienti e trasformare idee creative in prodotti e servizi innovativi, che possano contribuire a rendere la mobilità sempre più sostenibile. Numerosi sono i traguardi ottenuti grazie agli studi del centro: nella riduzione delle emissioni (dalle motorizzazioni Fire al Common Rail, dal Multijet e sistema TetraFuel alla gamma Natural Power fino al sistema Stop&Start), nel garantire la massima sicurezza dei veicoli nella guida di tutti i giorni e i massimi punteggi nei rating (dai dispositivi per la stabilità e il

controllo della dinamica del veicolo all'impiego estensivo di acciai a elevatissima resistenza) e tanto altro ancora. Complessivamente il gruppo detiene oltre 8000 brevetti attivi in portafoglio, a sottolineare ancora una volta il ruolo cardine che gioca la ricerca in un reparto con una continua necessità di innovazioni come quello automobilistico.

La data che ha segnato la storia del gruppo è il 10 Giugno 2009 che segna l'inizio dell'alleanza strategica globale col Chrysler Group LLC. Questa sinergia tra i due gruppi ha portato alla loro fusione nell'anno 2014 con la riorganizzazione in un unico grande gruppo industriale denominato Fiat Chrysler Automobiles s.p.a (FCA).

Ad oggi, FCA è uno dei più grandi gruppi industriali internazionali: con un fatturato 2016 dichiarato di quasi 111 miliardi di euro e oltre 4,74 milioni di veicoli prodotti nel 2017 (che ne fanno l'ottavo costruttore al mondo), esso comprende 16 brand commerciali che vanno dai marchi generalisti Fiat, Lancia, Alfa Romeo, Jeep, Chrysler, Dodge al settore delle auto sportive e di lusso con Ferrari e Maserati fino al settore di produzione e vendita di componenti e sistemi di produzione per il settore automotive (Magneti Marelli, Teksid, Comau). Oltre 231000 dipendenti lavorano nei 24 stabilimenti del gruppo, divisi su quattro aree operative regionali: Stati Uniti, Canada e Messico (NAFTA), America Latina (LATAM), Asia e stati del Pacifico (APAC), Europa, Medio Oriente e Africa (EMEA).

1.2 Il progetto PNet

Il progetto Production allocation Network (PNet) nasce dalla collaborazione tra diversi enti:

- il dipartimento di Manufacturing Engineering (ME) di FCA Italia;
- il team World Class Manufacturing Research and Innovation (WCMR&I) del CRF;
- centro ICT for City logistic and Enterprises Laboratory (ICE Lab) del Politecnico di Torino;
- Center for Automotive Research and Sustainable mobility (CARS) del Politecnico di Torino.

Tale progetto mira alla creazione di uno strumento software, che deve essere usato dai manager dell'ufficio ME, utile per risolvere il problema di ottimizzazione della fornitura delle componenti agli impianti di montaggio dei veicoli.

1.2.1 Il problema da risolvere

I manager del dipartimento ME ricevono da esperti di marketing delle analisi di mercato relative alla stima della domanda dei diversi veicoli in ciascuna regione in cui il gruppo vende automobili. I manager sanno che ciascun impianto di assemblaggio dei veicoli è preposto alla produzione di una sola tipologia di veicolo che richiede certe tipologie di componenti (porte, paraurti ecc). Uno o più tipi di componenti possono essere assemblati a loro volta in specifici stabilimenti sparsi nel mondo. Ciò che interessa ai manager è riuscire a individuare i giusti volumi produttivi di pezzi negli impianti fornitori, che consentano di rifornire in maniera adeguata gli impianti clienti per poter soddisfare la domanda stimata. Per “giusti volumi” e “rifornire in maniera adeguata” s’intende che la produzione e la distribuzione delle componenti debba comportare il minor costo possibile per quanto riguarda sia la produzione, sia per il loro trasporto verso gli stabilimenti clienti, senza avere dei pezzi in eccesso depositati nei magazzini.

1.2.2 I dati e la loro raccolta

I manager ME hanno a disposizione i seguenti dati:

- gli scaglioni produttivi che corrispondono alla combinazione di volumi di componenti delle diverse tipologie che lo stabilimento può produrre (esempio: 10000 porte lato guidatore e 20000 porte per il bagagliaio);
- gli investimenti usati per l’apertura dell’impianto con uno specifico scaglione produttivo;
- Present Value Added (PVA) che corrisponde al valore attuale aggiunto ed è un costo unitario per ciascun tipo di componente prodotto con un certo scaglione produttivo;
- costi logistici outbound, cioè costi di trasporto unitario delle componenti che vanno da un impianto fornitore di pezzi verso l’impianto cliente produttore di veicoli;
- dazi espressi in percentuale rispetto al costo totale di produzione e trasporto delle componenti che vanno dal fornitore al cliente;
- costi logistici inbound, cioè costi di trasporto unitario all’interno del paese in cui si trova l’impianto fornitore, che varia a secondo del tipo di componente;

- Direct Material Cost (DMC) costo unitario per la produzione dei pezzi legato al tipo di componente da fornire;
- Markup è il rapporto tra il costo di produzione del bene nello stabilimento fornitore e il suo prezzo di vendita allo stabilimento cliente, espresso in percentuale.

I dati relativi ai costi di produzione e trasporto(espressi in Euro) vengono forniti dai manager ME del gruppo, dislocati nelle varie sedi, ai manager ME Italia. I manager che consegnano questi dati hanno i loro sistemi informatici di raccolta e sintesi dati, e i manager italiani usano questi dati per sintetizzarli ulteriormente ottenendo così dei valori medi per ciascuna tipologia di costo. Costoro decidono di avere dei dati in estrema sintesi, per poter fare una stima iniziale e approssimata delle varie componenti di costo quando si passa alla pianificazione della produzione e distribuzione delle componenti. Il loro interesse è quello di ottenere una pianificazione veloce, i cui costi più dettagliati saranno determinati in un secondo momento.

1.2.3 L'interfaccia utente

L'interfaccia utente è stata sviluppata in un foglio di calcolo Microsoft Excel 2010. La scelta di tale soluzione è stata influenzata da diversi fattori: all'inizio dell'attività di tirocinio, il team di sviluppo aveva creato un foglio di calcolo con tantissime tabelle relative ai diversi costi, e sviluppava delle macro per preparare i dati al solver, il programma eseguibile responsabile della generazione delle soluzioni... pertanto si è deciso di continuare in questo ambiente per poter testare e verificare al meglio l'output del solver, consentendo di controllare rapidamente la validità del modello matematico adottato; i manager di ME si trovano a proprio agio nell'uso dei fogli di calcolo, e fornirne uno con una funzionalità aggiuntiva, consente loro di avere un primo approccio nell'uso di quest'ultima, per poi dare indicazioni ulteriori dei dati da gestire e inserire.

All'interno del file Excel si hanno diversi fogli di calcolo, ognuno contenente le tabelle dove inserire le diverse tipologie di costo. I fogli in questione sono: "Interface", "Production Activity", "Production Costs", "Transportation Cost", "Intermediate Calculation", "Intermediate 2", "Results". I fogli "Intermediate Calculation" e "Intermediate 2" vengono usati da una macro in Visual Basic per creare delle matrici da dare in input al solver, in modo che se cambiano il numero di impianti attivati, o di scaglioni, o di impianti destinazione, il solver riuscirà sempre a prendere le diverse matrici delle dimensioni necessarie

per l'istanza di dati attuale che gli viene fornita. Si fa presente che i dati riportati nelle tabelle sono fittizi e semi-casuali in quanto all'interno del CRF vige la restrizione sul rilascio dei dati sensibili. L'utente inizia a compilare i fogli nel seguente ordine: "Interface", "Production Activity", "Production Costs", "Transportation Cost". Di seguito si hanno le descrizioni delle varie tabelle mostrate nelle figure del foglio "Interface".

Nelle tabelle in figura 1.2 l'utente inserisce:

Objective: Providing powertrain units to FGA plants through a production network that minimizes total costs (logistic + production)

STEP 1: FGA plants de Meifi

Please define total demand of complete products (CBU) and its partition for each vehicle FGA plant

TYPE	I	II	III	IV
Total demand TYPE I	400000			
Total demand TYPE II	0			
Total demand TYPE III	0			
Total demand TYPE IV	0			

Required quantity

Plant Name	TYPE I	TYPE II	TYPE III	TYPE IV
EMEA1_V	0	0	0	0
EMEA2_V	0	0	0	0
EMEA3_V	100000	0	0	0
LATAM1_V	0	0	0	0
LATAM2_V	0	0	0	0
LATAM3_V	0	0	0	0
APAC1_V	200000	0	0	0
APAC2_V	100000	0	0	0
APAC3_V	0	0	0	0

Legend

Component	Code
EMEA1_C	C1
EMEA2_C	C2
EMEA3_C	C3
LATAM1_C	C4
LATAM2_C	C5
LATAM3_C	C6
APAC1_C	C7
APAC2_C	C8
APAC3_C	C9

Depreciation periods definition

Years: 10

Figura 1.2: Tabelle di definizione degli impianti, delle domande e delle componenti

- i nomi degli impianti di assemblaggio dei veicoli (saranno denotati con VP che sta per Vehicle Plant);
- i nomi degli stabilimenti di assemblaggio delle componenti (denotati con CP che sta per Components Plant);
- i nomi di ciascuna tipologia dei pezzi (fino ad un massimo di 4 componenti)
- il numero di componenti richieste da ciascun VP;
- la domanda totale delle componenti per tipologia;
- il numero di anni in cui effettuare l'ammortamento degli investimenti.

Nella tabella in figura 1.3 l'utente inserisce i costi DMC che vengono sommati ai costi logistici inbound per gli impianti CP. L'utente ha fatto questa scelta di unire i costi delle due tipologie poiché sono entrambi costi unitari e dipendono dal tipo di componente.

I tre tasti mostrati nella figura 1.4, vengono premuti dall'utente una volta che ha finito

	A	B	C	D	E	F	G	H	I	J
19										
20	DMC costs									
21										
22	Allocation maps									
23	EMEA			LATAM/NAFTA			APAC			
24	TYPE	C1	C2	C3	C4	C5	C6	C7	C8	C9
25	I	83,0	0,0	156,1	0,0	0,0	0,0	209,0	0,0	0,0
26	II	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
27	III	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
28	IV	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
29										

Figura 1.3: Tabelle di definizione dei costi DMC

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R													
43	STEP 10: Identification of solutions portfolio with least total cost																														
44																															
45																															
46	Please insert maximum number of solutions to generate										80																				
47																															
48	Start optimization with LINGO					Start optimization with CPLEX					Prepare Data for Solvers																				
49																															
50																															
51																															

Figura 1.4: Pulsanti per le varie operazioni

di compilare tutte le tabelle dei costi nei diversi fogli. Ovviamente “Prepare Data for Solvers” ha la precedenza rispetto agli altri due bottoni, in modo che la macro associata possa creare le tabelle ridotte contenenti le informazioni da passare ai solver. Si hanno due tasti, ognuno per attivare un solver diverso in quanto, all’inizio dell’attività di tesi, lo sviluppo del software di ottimizzazione è iniziato con l’uso di Lingo della Lindo Systems, e successivamente sono state effettuate delle prove con CPLEX di proprietà di IBM. Nel capitolo 4 vengono mostrati i test di confronto dei due solver.

Quanto segue è la descrizione delle tabelle del foglio “Production Activity”. La tabella in figura 1.5 viene usata dall’utente per inserire uno scaglione produttivo disponibile per ciascun CP. Successivamente, l’utente preme il pulsante “Insert Steps” mostrato nella figura 1.6 per inserire uno o più scaglioni nella tabella in figura 1.7, i quali non devono corrispondere allo scaglione nullo, in cui non si produce alcun tipo di pezzo. Il pulsante “Clear Input” azzerà i valori nella tabella in figura 1.5, mentre “Reset steps” elimina il contenuto nella tabella in figura 1.7. “Check step” verifica che l’utente abbia inserito degli scaglioni i cui volumi produttivi siano un multiplo intero del valore inserito nella cella in figura 1.6, e che non superi la quantità massima delle componenti prodotte definita sempre dall’utente e mostrato nella medesima figura. L’ultimo pulsante “Create tables” attiva una macro che definisce le righe in tutte le tabelle nel foglio “Production costs”, in cui ciascuna riga corrisponde a uno dei possibili scaglioni attivabili.

	A	B	C	D	E	F	G	H
1								
2			Insert here the production step for each plant					
3								
4			Components Plants	N3	N4	T3	T4	
5			C1	0	0	0	0	
6			C2	0	0	0	0	
7			C3	300000	0	0	0	
8			C4	0	0	0	0	
9			C5	0	0	0	0	
10			C6	0	0	0	0	
11			C7	0	0	0	0	
12			C8	0	0	0	0	
13			C9	0	0	0	0	
14								

Figura 1.5: Tabella per l'inserimento degli scaglioni

	J	K	L	M	N	O	P	Q	R	S	T
1											
2											
3											
4		Insert the step production displacement:		50000							
5		Insert the maximum amount of pieces for each type:		950000							
6											
7											
8											
9											
10											
11											
12											

Check step

Insert steps

Clear input

Reset steps

Create Tables

Figura 1.6: Pulsanti per controlli e verifiche

Le tabelle nelle figure 1.8, 1.9, 1.10 e 1.11 sono usate per specificare i costi produttivi, e sono presenti nel foglio "Production Costs". Nella prima figura s'inseriscono i costi di PVA di ciascun scaglione che il CP può attivare. Le celle a zero di questa tabella indicano anche che lo scaglione in quella riga non è attivabile per l'impianto in quella specifica colonna. Nella tabella in figura 1.9 sono definiti gli investimenti (in milioni di euro) necessari per l'apertura dello stabilimento con il relativo scaglione produttivo. Le altre due tabelle nelle figure 1.10 e 1.11 hanno all'interno delle celle le formule che effettuano rispettivamente calcoli del costo dell'ammortamento medio per componente prodotto e del DMC medio. La formula per il calcolo dell'ammortamento medio è la seguente

$$\mu_a = \frac{i \times 1000000}{p \times t} \quad (1.1)$$

	A	B	C	D	E	F	G	H
16								
17			Available Production Steps and Plants					
18			Components Plants	N3	N4	T3	T4	
19			C7	100000	0	0	0	
20			C1	200000	0	0	0	
21			C3	200000	0	0	0	
22			C7	200000	0	0	0	
23			C3	300000	0	0	0	
24								

Figura 1.7: Lista degli scaglioni attivabili in ogni impianto

	A	B	C	D	E	F	G	H	I	J	K	L	M
1													
2			PVA Costs										
3													
4					PLANTS								
5					EMEA			LATAM/NAFTA			APAC		
6	Type1	Type2	Type3	Type4	C1	C2	C3	C4	C5	C6	C7	C8	C9
7	100000	0	0	0	0	0	0	0	0	0	12,0	0	0
8	200000	0	0	0	15,0	0	33,0	0	0	0	11,0	0	0
9	300000	0	0	0	0	0	94,0	0	0	0	0	0	0
10													

Figura 1.8: Tabella dei costi PVA per ciascun scaglione

dove μ_a corrisponde all'ammortamento medio, i è l'investimento, p il periodo di ammortamento, espresso in numero di anni, definito dall'utente nella cella in figura 1.2, t il totale delle componenti prodotti con lo scaglione associato alla riga in cui si trova la cella. Per il calcolo del DMC medio dello scaglione produttivo, si usa la formula che segue

$$\mu_{DMC} = \frac{DMC_1 \times q_1 + DMC_2 \times q_2 + DMC_3 \times q_3 + DMC_4 \times q_4}{q_1 + q_2 + q_3 + q_4} \quad (1.2)$$

dove μ_{DMC} corrisponde alla media pesata dei DMC delle diverse tipologie i delle componenti prodotte, pesate in base alle quantità q_i prodotta. In questo modo si ottiene un costo DMC unitario che dipende dallo stabilimento e il relativo scaglione attivati.

Le tabelle nelle figure 1.12, 1.13 e 1.14 sono usate per specificare i costi logistici, e sono presenti nel foglio "Transportation Costs". I costi di Markup (presente in figura 1.12), sono fissati dall'utente e determinati dalla seguente formula:

$$M = \frac{v}{c} - 1 \quad (1.3)$$

	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA
1													
2			Investments										
3													
4					PLANTS								
5					EMEA			LATAM/NAFTA			APAC		
6	Type1	Type2	Type3	Type4	C1	C2	C3	C4	C5	C6	C7	C8	C9
7	100000	0	0	0	0	0	0	0	0	0	4,0	0	0
8	200000	0	0	0	13,0	0	39,0	0	0	0	17,0	0	0
9	300000	0	0	0	0	0	38,0	0	0	0	0	0	0
10													

Figura 1.9: Tabella degli investimenti(espressi in milioni di euro) per l'attivazione degli impianti con specifici volumi produttivi

	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO
1													
2			Depreciations										
3													
4					PLANTS								
5					EMEA			LATAM/NAFTA			APAC		
6	Type1	Type2	Type3	Type4	C1	C2	C3	C4	C5	C6	C7	C8	C9
7	100000	0	0	0	0	0	0	0	0	0	4	0	0
8	200000	0	0	0	6,5	0	19,5	0	0	0	8,5	0	0
9	300000	0	0	0	0	0	12,66667	0	0	0	0	0	0
10													

Figura 1.10: Tabella dei costi di ammortamento medio per componente prodotto per ogni scaglione

dove M è la percentuale di Markup, v il prezzo di vendita e c il costo di produzione. Il lettore potrebbe chiedersi per quale ragione nel calcolo dei costi si tiene conto anche di questo fattore, dato che è un sovrapprezzo del costo di produzione. Lo scopo è quello di garantire una copertura della spesa di produzione dei componenti che vengono prodotte e distribuite globalmente, per incrementare il profitto nella vendita dei veicoli una volta che gli saranno montate.

I dazi che vengono pagati quando le componenti vengono trasportate via mare verso le diverse destinazioni (definiti nella tabella in figura 1.14) sono anch'essi espressi in percentuale e dipendono fortemente dalla legge del paese importatore e dalla tipologia di componente esportato. I costi di trasporto unitario per pezzo via mare sono definiti nella tabella in figura 1.13. Si osservi che per certe tratte sono stati inseriti di proposito valori esorbitanti in modo che il software risolutore eviti il trasporto per tratte non abilitate.

	AQ	AR	AS	AT	AU	AV	AW	AX	AY	AZ	BA	BB	BC
1													
2			DMC										
3													
4					PLANTS								
5					EMEA			LATAM/NAFTA			APAC		
6	Type1	Type2	Type3	Type4	C1	C2	C3	C4	C5	C6	C7	C8	C9
7	100000	0	0	0	83	0	156,1	0	0	0	209	0	0
8	200000	0	0	0	83	0	156,1	0	0	0	209	0	0
9	300000	0	0	0	83	0	156,1	0	0	0	209	0	0
10													

Figura 1.11: Tabella dei DMC medi pesati relativi agli scaglioni e impianti attivati

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Mkp settings												
2													
3					Starting plant								
4					EMEA			LATAM/NAFTA			APAC		
5													
6													
7													
8													
9													
10													
11													
12													
13													
14													
15													

Figura 1.12: Tabella dei costi di Markup per ciascuna tratta

1.2.4 Dati in output e relative specifiche

Gli utenti del sistema una volta che avranno inseriti tutti i dati nei fogli del file Excel, premeranno il bottone “Prepare data for solvers”, la cui macro associata effettuerà un lavoro di pre-elaborazione dei dati di input, in modo che vengano create delle matrici ridotte con i dati essenziali nei fogli “Intermediate calculation” e “Intermediate 2”, associando a ciascuna matrice un nome(che nell’ambiente Excel identifica un gruppo univoco di celle), in modo che il solver possa prendere quello che serve. Gli utenti quando avranno avviato l’esecuzione del solver, si aspettano i seguenti dati in output nel foglio “Results”:

1. costo medio per componente prodotta e distribuita a livello globale
2. gli impianti attivati con i relativi volumi produttivi
3. le quantità delle componenti trasportate per ciascuna tipologia, per le varie tratte da fornitori a clienti

	A	B	C	D	E	F	G	H	I	J	K	L	M
19													
20	Shipping costs settings												
21													
22	Shipping		From:	Starting plant									
EMEA				LATAM/NAFTA			APAC						
23	To:		PLANT	C1	C2	C3	C4	C5	C6	C7	C8	C9	
24	Destination plant	EMEA	V1	1,3	0,0	1000000,0	1000000,0	87,0	1000000,0	1000000,0	1000000,0	1000000,0	
25			V2	0,0	1000000,0	25,0	1000000,0	71,0	1000000,0	1000000,0	1000000,0	1000000,0	
26			V3	10,0	1000000,0	5,0	1000000,0	77,0	1000000,0	100,0	1000000,0	1000000,0	
27		LATAM/N AFTA	V4	30,0	1000000,0	4,9	1000000,0	75,0	1000000,0	108,0	1000000,0	1000000,0	
28			V5	1000000,0	0,0	1000000,0	1000000,0	0,0	1000000,0	1000000,0	1000000,0	1000000,0	
29			V6	1000000,0	0,0	1000000,0	1000000,0	1000000,0	0,0	1000000,0	1000000,0	1000000,0	
30		APAC	V7	20,0	1000000,0	40,0	1000000,0	59,0	70,0	125,0	1000000,0	1000000,0	
31			V8	30,0	1000000,0	50,0	9,0	77,0	15,0	115,0	1000000,0	1000000,0	
32			V9	22,0	22,0	1000000,0	1000000,0	74,0	47,0	1000000,0	1000000,0	1000000,0	
33													
34													

Figura 1.13: Tabella dei costi di trasporto via mare delle componenti per ciascuna tratta

	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK
19																							
20																							
21																							
22	Duty		From:	Starting plant																			
23				C1				C2				C3				C4				C5			
24	To:		PLANT	N3	N4	T3	T4	N3	N4	T3	T4	N3	N4	T3	T4	N3	N4	T3	T4	N3	N4	T3	T4
25	Destination plant	EMEA	V1	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	4,20%	2,70%	2,70%	4,20%
26			V2	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	4,20%	0,00%	0,00%	0,00%
27			V3	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	4,20%	0,00%	0,00%	0,00%
28		LATAM/N AFTA	V4	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	4,20%	0,00%	0,00%	0,00%
29			V5	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
30			V6	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	18,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
31		APAC	V7	2,00%	0,00%	2,70%	18,00%	0,00%	0,00%	0,00%	0,00%	2,00%	0,00%	2,70%	18,00%	10,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
32			V8	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
33			V9	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	4,20%	2,70%	2,70%	4,20%

Figura 1.14: Tabella dei dazi da pagare per ciascuna destinazione

per tutte le migliori soluzioni trovate dal solver per la medesima istanza di dati. Gli utenti non vogliono avere a disposizione solo la soluzione ottima del problema, cioè quella con il minor costo medio in assoluto, ma anche le soluzioni subottime che vengono dopo la migliore. Si deve fornire all'utente uno strumento capace di esplorare lo spazio delle soluzioni migliori per il problema di pianificazione corrente da affrontare, in modo da avere un'idea di ciascuna soluzione e i relativi vantaggi e svantaggi che dovrà valutare.

Il costo medio per componente è dato dalla somma delle diverse componenti medie

$$\mu_m = \mu_{pg} + \mu_{ag} + \mu_{lg} + \mu_{dg} + \mu_{Mg} \quad (1.4)$$

dove: μ_m è il costo medio; μ_{pg} sta per costo medio di produzione; μ_{ag} indica l'ammortamento medio per gli investimenti fatti nell'attivazione dei CP con i relativi scaglioni scelti; μ_{lg} rappresenta il costo medio per il trasporto tra due stabilimenti; μ_{dg} è il costo medio dei dazi applicati sulle diverse tratte; μ_{Mg} indica il costo medio del Markup applicato per

ciascun collegamento tra due stabilimenti. La lettera g , presente ai pedici delle diverse componenti medie di costo, sta per ‘globale’ in quanto queste medie vengono calcolate in base a tutti le componenti prodotte nei diversi stabilimenti e trasportati sulle diverse tratte.

Le soluzioni che vengono fornite dal solver, devono essere soggette a queste ulteriori specifiche:

1. μ_{pg} deve tenere conto dei costi PVA e DMC (i DMC sono sommati ai costi di logistica inbound) dei diversi stabilimenti attivati con i volumi produttivi scelti;
2. il costo dei dazi applicati per il trasporto da un impianto fornitore verso l’impianto cliente deve rispettare la seguente formula

$$Cd_{ijm} = (Cp_{ijm} + Cl_{ijm} + Ca_{ijm} + Cmkp_{ijm}) \cdot d_{ijm} \quad (1.5)$$

dove: i rappresenta il CP; j è il VP; m indica la tipologia di componente; Cd_{ijm} rappresenta il costo totale del dazio(espresso in Euro) applicato a tutti i pezzi m trasportati lungo la tratta dal fornitore i al cliente j ; Cp_{ijm} è il costo di produzione totale rispetto a una certa quantità di pezzi m trasportati sulla tratta ij ; Cl_{ijm} è il totale dei costi logistici da affrontare per le componenti m trasportati sulla tratta ij ; Ca_{ijm} rappresenta il costo di ammortamento totale delle componenti m trasportate sulla tratta ij ; CM_{ijm} è il costo totale dei Markup applicati ai pezzi m trasportati sulla tratta ij .

3. il costo totale di Markup(espresso in Euro) delle componenti di una certa tipologia trasportati su una tratta tra due stabilimenti è dato dalla formula

$$Cmkp_{ijm} = Mkp_{ij} \cdot (Cp_{ijm} + Ca_{ijm}) \quad (1.6)$$

dove i termini CM_{ijm} , Cp_{ijm} e Ca_{ijm} hanno gli stessi significati espressi in precedenza, mentre $Mkp(i, j)$ rappresenta la percentuale di Markup applicata sulla tratta ij

4. possono essere attivati solo gli scaglioni produttivi forniti dall’utente per ciascun stabilimento;
5. si deve attivare per ciascun CP al più un solo scaglione produttivo;

6. se viene aperto un CP, tutti le componenti prodotte di una certa tipologia devono andare a soddisfare, pienamente o in parte, le domande delle componenti di quella medesima tipologia di uno o più VP;
7. le domande di pezzi delle diverse tipologie degli VP devono essere pienamente soddisfatte;
8. non si possono avere sovrapproduzioni delle componenti;
9. le quantità delle componenti trasportate nelle diverse tratte da CP a VP, devono essere un multiplo intero di un certo numero di componenti, definito dall'utente nella figura 1.6;

Dopo aver illustrato tutti i requisiti del problema e i dati da elaborare, si può iniziare a mostrare la teoria della modellazione matematica, adottata per la risoluzione, nel Capitolo 2.

Capitolo 2

Teoria di ottimizzazione

Nel capitolo precedente sono stati forniti i dettagli del problema da risolvere, relativi ai dati e alle specifiche che le soluzioni devono rispettare. In questo capitolo viene illustrata la teoria usata per ottenere il modello matematico secondo la Ricerca Operativa: Programmazione Lineare e Programmazione Lineare Intera. Sarà descritto il metodo risolutivo che viene normalmente applicato a quest'ultima tipologia di modelli, e indicate le tecniche con cui le specifiche del problema vengono tradotte in vincoli del modello finale

2.1 Problema di ottimizzazione

Un problema di ottimizzazione può essere rappresentato matematicamente nella seguente forma:

$$\begin{aligned} & \text{minimizza } f_0(x) \\ & \text{soggetta a } f_i(x) \leq b_i \quad i = 1, \dots, m. \end{aligned} \tag{2.1}$$

oppure:

$$\begin{aligned} & \text{massimizza } f_0(x) \\ & \text{soggetta a } f_i(x) \leq b_i \quad i = 1, \dots, m. \end{aligned} \tag{2.2}$$

dove: $x = (x_1, x_2, \dots, x_n)^T$ è un vettore di n variabili decisionali del problema; la funzione $f_0(x) : \mathbf{R}^n \rightarrow \mathbf{R}$ è denominata *funzione obiettivo*; le funzioni $f_i(x) : \mathbf{R}^n \rightarrow \mathbf{R}$, $i = 1, \dots, m$

corrispondono alle *funzioni vincolari* e le costanti b_1, \dots, b_m sono i limiti di questi vincoli. Un vettore x^* è detto *ottimo* se gli viene associato un valore della funzione obiettivo che corrisponde al minimo assoluto (o al massimo assoluto) tra tutti i vettori che soddisfano i vincoli: $\forall z \in \mathbf{R}^n$ tale che $f_1(z) \leq b_1, \dots, f_m(z) \leq b_m$ si ha $f_0(z) \geq f_0(x^*)$ (o $f_0(z) \leq f_0(x^*)$). Questa rappresentazione matematica è un'astrazione generale del problema che permette di effettuare la scelta migliore in \mathbf{R}^n dei valori delle variabili decisionali tra un insieme di possibili vettori candidati alias *soluzioni*.

2.1.1 Programmazione Lineare

La *Programmazione Lineare* (PL) è una classe di problemi di ottimizzazione in cui la funzione obiettivo e i vincoli sono definiti come combinazioni lineari delle variabili decisionali

$$\begin{aligned} &\text{minimizza } c^T x \\ &\text{soggetta a } a_i^T x \leq b_i \quad i = 1, \dots, m. \end{aligned} \tag{2.3}$$

dove i vettori $c, a_1, \dots, a_m \in \mathbf{R}^n$ e gli scalari b_1, \dots, b_m sono parametri del problema che specificano la funzione obiettivo e le funzioni vincolari.

Non esiste una formula analitica semplice per la risoluzione di un problema lineare, ma esistono una serie di metodi efficaci per la risoluzione, tra cui: *metodo del simplesso*, *metodo dei punti interni* e l'algoritmo *Branch and Bound* con le sue varianti. In questo capitolo verrà esposto l'algoritmo Branch and Bound, ma prima si devono introdurre ulteriori classi di problemi di ottimizzazione.

2.1.2 Programmazione Lineare Intera, Intera Booleana e Mista Intera

La *Programmazione Lineare Intera* (PLI) è un caso speciale di PL in quanto tutte le variabili decisionali anziché assumere valori reali assumono valori interi ($x \in \mathbf{Z}^n$). Un'ulteriore sotto-classe di PL è la *Programmazione Lineare Intera Booleana* (PLIB) dove le variabili decisionali assumono solo due valori interi: 0 o 1 ($x_i \in \{0, 1\}, i = 1, \dots, n$). L'ultima sotto-classe è la *Programmazione Lineare Mista Intera* (PLMI) dove le variabili decisionali non appartengono allo stesso insieme di numeri, ma si hanno alcune variabili intere, altre variabili reali, e/o altre ancora booleane.

2.2 Metodi risolutivi: algoritmo Branch and Bound e simili

L'algoritmo Branch and Bound (B&B) è uno degli algoritmi più usati per la risoluzione dei problemi PLI, PLIB e PLMI. È un metodo che consente di esplorare sia alcuni sottoinsiemi di soluzioni ammissibili raggiungendo la soluzione ottima globale, e allo stesso tempo permette di escluderne altri evitando così di enumerare tutte le soluzioni del problema.

2.2.1 Algoritmo base

Sia dato un problema base P_B di PLI (di minimo o di massimo) espresso nella *forma standard* come segue

$$\begin{aligned} &\text{minimizza } c^T x \\ &\text{soggetta a } a_i^T x \leq b_i \quad i = 1, \dots, m \\ &x \in \mathbf{Z}^n \end{aligned} \tag{2.4}$$

Viene applicato al modello un rilassamento di uno o più vincoli: ad esempio viene eliminato il vincolo di interezza delle variabili decisionali ottenendo così un problema PLC (Programmazione Lineare Continua) che lo si denota con PL_B . Utilizzando un metodo di risoluzione dei problemi PLC, come l'algoritmo del simplesso, si ottiene una soluzione x_{LB}^* ottima per il modello PL_B . A questo punto si effettua la valutazione della soluzione:

- se la soluzione x_{LB}^* è intera allora è stata trovata una soluzione per P_B e l'algoritmo termina;
- altrimenti si sceglie il valore di una variabile x_f^* che appartiene alla soluzione correntemente trovata, e sarà usata per costruire due sottoproblemi:

$$\begin{aligned} &\text{minimizza } c^T x \\ &\text{soggetta a } a_i^T x \leq b_i \quad i = 1, \dots, m. \\ &x \in \mathbf{Z}^n \\ &x_f \leq \lfloor x_f^* \rfloor \end{aligned} \tag{2.5}$$

$$\begin{aligned}
& \text{minimizza } c^T x \\
& \text{soggetta a } a_i^T x \leq b_i \quad i = 1, \dots, m. \\
& x \in \mathbf{Z}^n \\
& x_f \leq \lceil x_f^* \rceil
\end{aligned} \tag{2.6}$$

dove $\lceil x_f^* \rceil$ e $\lfloor x_f^* \rfloor$ corrispondono rispettivamente all'arrotondamento per difetto e per eccesso della variabile x_f^* . Tale operazione viene detta di *branch* in quanto partiziona l'insieme di ricerca delle soluzioni ammissibili in due sottoinsiemi dati dai problemi 2.5 e 2.6 che saranno indicati con P_1 e P_2 , e la variabile x_f di separazione sarà nominata *variabile di branch*. Tali problemi generati vengono detti *figli* di P_B . La sequenza di operazioni *rilassamento - risoluzione - valutazione - branching* viene ripetuta ciclicamente per ogni sotto-problema generato fino a quando non si trova più una variabile decisionale su cui effettuare il branch. L'algoritmo, in questo modo, costruisce un *albero* dei sotto-problemi generati a partire dal problema base, dove ciascun nodo dell'albero N è un problema che è sottoposto ai vincoli base del nodo radice P_B e da tutti i vincoli di branch che si incontrano a partire dalla radice lungo il percorso dell'albero fino a giungere al nodo N .

Se l'algoritmo operasse effettuando sempre il branching, allora non risulta essere un metodo risolutivo per l'ottimizzazione, ma un *metodo enumerativo* delle soluzioni ammissibili di un problema PLI. Per evitare la generazione dell'intero albero, si adottano i seguenti tre criteri per la valutazione di ciascun nodo generato.

1. **Inammissibilità della soluzione:** il nuovo nodo generato N viene chiuso se i vincoli di branch relativi al percorso dell'albero dalla radice fino al nodo corrente risultano incompatibili con i vincoli del problema base P_B .
2. **Soluzione intera:** sia N il nodo corrente esplorato dell'albero, associato al problema rilassato PL_N che ha una soluzione ottima intera x_N^* . Allora si valuta il valore della funzione obiettivo ad esso associato $f_0(x_N^*)$ con $f_0(x_B^*)$ dove x_B^* è la soluzione migliore che è stata trovata fino a quel momento. Se risulta che $f_0(x_N^*) < f_0(x_B^*)$ allora (x_B^*) viene aggiornato con (x_N^*) e il nodo N viene chiuso.
3. **Assenza di soluzione migliorante:** sia N il nodo corrente esplorato dell'albero, associato al problema rilassato PL_N che ha una soluzione ottima non intera x_N^* . Se il valore associato della funzione obiettivo soddisfa la seguente condizione $f_0(x_N^*) \geq f_0(x_B^*)$ allora il nodo N verrà chiuso. Si effettua quest'azione in quanto

tutti i sottoproblemi che vengono generati a partire dal nodo corrente non riusciranno ad ottenere un valore della funzione obiettivo inferiore all'estremo inferiore dato da $f_0(x_N^*)$.

2.2.2 Branch and Cut

Tale algoritmo nasce come unione tra l'algoritmo Branch and Bound e la generazione dei *piani di taglio*, e viene normalmente applicato ai problemi PLI. Per spiegare un piano di taglio abbiamo bisogno della definizione di un *insieme convesso*.

Insieme convesso. Un insieme C viene detto *convesso* se ciascuna *combinazione convessa* dei punti $x_i \in C$, $i = 1, \dots, k$ rimane in C . Una combinazione convessa viene espressa con la seguente formula

$$\begin{aligned} c &= \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_k x_k, \quad \theta_i \geq 0, \\ \sum_{i=1}^k \theta_i &= 1, \quad i = 1, \dots, k \end{aligned} \tag{2.7}$$

I vincoli di un problema PL (indipendentemente se le variabili sono intere o continue) definiscono un insieme convesso.

Taglio. Dato un insieme convesso C , un taglio è un vettore $x \in C$ che soddisfa un vincolo lineare di tipo $\alpha^T x = \beta$ dove α^T è un vettore di coefficienti, mentre β è uno scalare non negativo.

Algoritmo B&C

- 0 Porre $z = \min\{c^T x : x \in X\}$ dove c^T è il vettore dei coefficienti della funzione obiettivo, x è il vettore delle variabili decisionali e X è l'insieme convesso delle soluzioni che soddisfano i vincoli del problema. z^* che corrisponde al valore della funzione obiettivo della migliore soluzione corrente x^* , viene posto a $+\infty$, mentre x^* viene lasciato vuoto. Tale definizione del problema base P_B viene inserito nella lista dei nodi dell'albero da esplorare.
- 1 Se la lista dei nodi è vuota andare al passo 7, altrimenti prelevare un nodo i , rimuoverlo dalla lista, porre $k = 1$ e andare al passo successivo.

- 2 Risolvere il problema $P^{i,k}$ con il rilassamento continuo $P_L^{i,k}$.
- 3 Se esiste una soluzione $x_L^{i,k}$ andare al passo 4, altrimenti si ritorna al passo 1.
- 4 Generare dei tagli che escludono $x_L^{i,k}$, aggiungerli al problema $P^{i,k}$ ottenendo $P^{i,k+1}$, incrementare k di 1 e ripetere il passo 2. Altrimenti se non esistono tagli si va al passo 5.
- 5 Se $x_L^{i,k} \in X$ e il corrispondente valore della funzione obiettivo $z_L^{i,k} < z^*$, porre $x^* = x_L^{i,k}$ e $z^* = z_L^{i,k}$ e ritornare al passo 1. In caso contrario andare al passo successivo.
- 6 Creare due o più sottoinsiemi X_b^i associati alle rispettive formulazioni P_b^i che vengono aggiunte alla lista dei nodi da esplorare.
- 7 La soluzione x^* è ottima con valore della funzione obiettivo z^* .

2.2.3 Confronto dei due algoritmi

Il normale algoritmo Branch and Bound con la fase di branch cerca di separare lo spazio di ammissibilità del problema in più sottospazi possibili aumentando la dimensione dell'albero dei sottoproblemi generati fin quando non trova una soluzione ottima intera dopo un rilassamento lineare. Il Branch and Cut invece per ogni nodo dell'albero, crea dei vincoli aggiuntivi sempre più stringenti finquando non trova una soluzione intera, altrimenti separa lo spazio di ricerca in due sottospazi per poter esprimere ulteriori tagli. In questo modo si riesce a convergere alla soluzione riducendo la crescita di dimensione dell'albero di ricerca, ma comporta un incremento di tempi di calcolo speso sui singoli nodi. Un vantaggio che introduce il B&C è l'aumento di efficienza dell'algoritmo rispetto alla versione base.

2.3 La modellazione del problema

Il problema affrontato, per come viene posto, è un tipico caso di *Capacited Facility Location Problem* (CFLP) e *Transportation Problem* (TP) in quanto bisogna aprire dei siti produttivi per poter soddisfare la domanda di siti clienti, e si devono minimizzare sia i

costi di apertura che i costi di trasporto . In letteratura viene riportato un modello generico (inserire riferimento) per ciascun problema e quindi si prestano ad essere facilmente estesi. In seguito vengono introdotti tutte le variabili e i dati coinvolti nel modello.

2.3.1 I dati di input e le variabili decisionali

Quando l'utente avrà finito di inserire i dati nell'interfaccia e avrà mandato in esecuzione il software si avranno in input per il modello i seguenti dati:

- f impianti produttori indicati con i ;
- r impianti clienti indicati con j ;
- s scaglioni produttivi distinti indicizzati con k ;
- e tipi di componenti da produrre e trasportare indicati con m ;
- $slot$ indica quanti pezzi alla volta possono essere trasportati tra una coppia di stabilimenti produttore e cliente.
- mm il massimo numero di componenti per ciascun tipo che possono essere prodotti;
- tot il numero totale di componenti prodotte per lo scenario corrente;
- l_{km} rappresenta la capacità produttiva per il pezzo di tipo m quando viene attivato lo scaglione di indice k ;
- d_{jm} è la domanda di componenti m per l'impianto j ;
- cl_{ij} costo logistico di trasporto outbound;
- Mkp_{ij} costo di Markup tra fornitore e cliente;
- pva_{ik} costo di PVA per la produzione di pezzi in un stabilimento i con uno scaglione produttivo k ;
- \bar{v}_{ik} indica se lo scaglione produttivo k è attivabile per l'impianto i con il valore 1, in caso contrario con il valore 0;

- amm_{ik} l'ammortamento degli investimenti per l'apertura dello stabilimento i con lo scaglione k ;
- da_{ijm} il dazio pagato per il trasporto dei pezzi di tipo m da un sito i al sito j ;

Si introducono le seguenti variabili decisionali di base del problema:

- q_{ijm} il numero intero di componenti di tipo m trasportati dal produttori i verso il richiedente j ;
- v_{ik} assume valore 0 se l'impianto produttore i non attiva lo scaglione di indice k , mentre vale 1 in caso contrario;

Al modello vengono aggiunte ulteriori variabili decisionali a supporto della traduzione di alcune specifiche:

- cp_{im} indica la capacità produttiva espressa in numero intero di pezzi m dell'impianto i ;
- qd_{ijm} il numero intero di *slot* di componenti di tipo m trasportate dal produttore i verso il richiedente j ;
- qt_{ij} il numero totale di componenti trasportati dal produttori i verso il richiedente j ;
- uq_{ijkm} indica il numero di pezzi di tipo m prodotti dal fornitore i per il richiedente j attivando lo scaglione k ;
- $Cpva_{ijm}$ costo totale di PVA per la produzione delle componenti di tipo m affrontati dallo stabilimento i per soddisfare la domanda dello stabilimento j ;
- Cam_{ijm} costo totale di ammortamento per la produzione dei pezzi di tipo m affrontati dall'impianto i per soddisfare la domanda dell'impianto j ;
- $Cmkp_{ijm}$ costo totale di Markup per la produzione delle componenti di tipo m affrontati dal sito i per soddisfare la domanda del sito j ;
- μ_{pg} sta per costo medio di produzione;
- μ_{ag} indica l'ammortamento medio per gli investimenti fatti nell'attivazione dei siti produttivi;

- μ_{lg} rappresenta il costo medio per il trasporto;
- μ_{dg} è il costo medio dei dazi applicati sulle diverse tratte;
- μ_{Mg} indica il costo medio del Markup applicato per ciascun collegamento tra due stabilimenti.

2.3.2 I vincoli

Per ciascuna specifica sono mostrati i vincoli associati del modello finale. La specifica 4 a pagina 20 viene tradotta con la seguente disequazione

$$v_{ik} \leq \bar{v}_{ik} \quad i = 1, \dots, f, k = 1, \dots, s \quad (2.8)$$

in quanto non è possibile attivare un impianto con volumi produttivi per cui non è predisposto. La specifica 5 della stessa pagina viene tradotta con

$$\sum_{k=1}^s v_{i,k} \leq 1 \quad i = 1, \dots, f \quad (2.9)$$

ed essendo le v_{ik} binarie verrà attivato al più uno scaglione per lo stabilimento i . Per determinare la capacità produttiva dell'impianto per una certa tipologia di componente, associato allo scaglione produttivo attivato, si ha quanto segue

$$cp_{im} = \sum_{k=1}^s v_{ik} l_{km} \quad i = 1, \dots, f, m = 1, \dots, e \quad (2.10)$$

dato che sarà attivato un solo scaglione, solo una riga della matrice l viene usata per la determinazione dei volumi produttivi di ciascun tipo di pezzo. Le specifiche 6, 7 e 8 a

pagina 20 sono tradotte con i vincoli che seguono

$$\begin{aligned}
 cp_{im} &= \sum_{j=1}^r q_{ijm} & i = 1, \dots, f, m = 1, \dots, e \\
 d_{jm} &= \sum_{i=1}^f q_{ijm} & j = 1, \dots, r, m = 1, \dots, e \\
 \sum_{i=1}^f cp_{im} &= \sum_{j=1}^r d_{jm} & m = 1, \dots, e
 \end{aligned} \tag{2.11}$$

Il terzo vincolo di 2.11 evita la sovrapproduzione globale rispetto alla domanda. La specifica 9 viene tradotta banalmente con

$$qd_{ijm}slot = q_{ijm} \quad i = 1, \dots, f, j = 1, \dots, r, m = 1, \dots, e \tag{2.12}$$

Vengono mostrati i vincoli aggiuntivi necessari per la correttezza logica dei dati di output e la definizione di certe variabili decisionali utili nel calcolo della funzione obiettivo. I vincoli seguenti

$$\begin{aligned}
 qt_{ij} &= \sum_{m=1}^e q_{ijm} & i = 1, \dots, f, j = 1, \dots, r \\
 \sum_{k=1}^s v_{ik} &\leq \sum_{j=1}^r qt_{ij} & i = 1, \dots, r,
 \end{aligned} \tag{2.13}$$

servono per evitare l'attivazione dell'impianto quando non viene trasportato alcun componente verso i vari clienti.

$$\begin{aligned}
 q_{ijm} - uq_{ijkm} &\leq (1 - v_{ik})mm & i = 1, \dots, f, j = 1, \dots, r, k = 1, \dots, s, m = 1, \dots, e, \\
 \sum_{i=1}^f \sum_{j=1}^r \sum_{k=1}^s uq_{ijkm} &= \sum_{j=1}^f d_{jm} & m = 1, \dots, e, \\
 \sum_{i=1}^f \sum_{j=1}^r \sum_{k=1}^s \sum_{m=1}^e uq_{ijkm} &= tot
 \end{aligned} \tag{2.14}$$

Per il primo vincolo di 2.14 si cerca di definire il numero di componenti m che sono stati prodotti con lo scaglione k nell'impianto i per soddisfare le necessità del cliente j . Il

secondo e il terzo vincolo servono per evitare che il totale dei pezzi nella matrice uq non superi il totale di pezzi da produrre sia globalmente che per ciascuna tipologia.

La nostra funzione obiettivo è data dalla equazione 1.4, pertanto per calcolare le varie componenti di costo medio si ricorrono ai seguenti vincoli

$$\begin{aligned}
 Cpva_{ijm} &= \sum_{k=1}^s pva_{ik} uq_{ijkm} & i = 1, \dots, f, j = 1, \dots, r, m = 1, \dots, e, \\
 Camm_{ijm} &= \sum_{k=1}^s amm_{ik} uq_{ijkm} & i = 1, \dots, f, j = 1, \dots, r, m = 1, \dots, e, \\
 Cmkp_{ijm} &= Mkp_{ij}(Camm_{ijm} + Cpva_{ijm}) & i = 1, \dots, f, j = 1, \dots, r, m = 1, \dots, e, \\
 \mu_{lg} &= \frac{1}{tot} \sum_{i=1}^f \sum_{j=1}^r cl_{ij} qt_{ij} \\
 \mu_{pg} &= \frac{1}{tot} \sum_{i=1}^f \sum_{j=1}^r \sum_{m=1}^e Cpva_{ijm} \\
 \mu_{dg} &= \frac{1}{tot} \sum_{i=1}^f \sum_{j=1}^r \sum_{m=1}^e (Cpva_{ijm} + cl_{ij} q_{ijm} + Camm_{ijm} + Cmkp_{ijm}) da_{ijm} \\
 \mu_{Mg} &= \frac{1}{tot} \sum_{i=1}^f \sum_{j=1}^r \sum_{m=1}^e Cmkp_{ijm} \\
 \mu_{ag} &= \frac{1}{tot} \sum_{i=1}^f \sum_{j=1}^r \sum_{m=1}^e Camm_{ijm}
 \end{aligned} \tag{2.15}$$

dove ciascuna μ è un componente di costo medio pesato.

2.3.3 Il modello finale

In sintesi viene riportato l'intero modello matematico usato per la risoluzione del problema.

$$\begin{aligned}
& \min \mu_{pg} + \mu_{ag} + \mu_{lg} + \mu_{dg} + \mu_{Mg} \\
& \text{soggetto a } v_{ik} \leq \bar{v}_{ik} \quad i = 1, \dots, f, \quad k = 1, \dots, s \\
& \sum_{k=1}^s v_{ik} \leq 1 \quad i = 1, \dots, f \\
& cp_{im} = \sum_{k=1}^s v_{ik} l_{km} \quad i = 1, \dots, f, \quad m = 1, \dots, e \\
& cp_{im} = \sum_{j=1}^r q_{ijm} \quad i = 1, \dots, f, \quad m = 1, \dots, e \\
& d_{jm} = \sum_{i=1}^f q_{ijm} \quad j = 1, \dots, r, \quad m = 1, \dots, e \\
& \sum_{i=1}^f cp_{im} = \sum_{j=1}^r d_{jm} \quad m = 1, \dots, e \\
& qd_{ijm} \text{slot} = q_{ijm} \quad i = 1, \dots, f, \quad j = 1, \dots, r, \quad m = 1, \dots, e \\
& qt_{ij} = \sum_{m=1}^e q_{ijm} \quad i = 1, \dots, f, \quad j = 1, \dots, r \\
& \sum_{k=1}^s v_{ik} \leq \sum_{j=1}^r qt_{ij} \quad i = 1, \dots, r, \\
& q_{ijm} - uq_{ijk} \leq (1 - v_{ik})mm \quad i = 1, \dots, f, \quad j = 1, \dots, r, \quad k = 1, \dots, s, \quad m = 1, \dots, e, \\
& \sum_{i=1}^f \sum_{j=1}^r \sum_{k=1}^s uq_{ijk} = \sum_{j=1}^f d_{jm} \quad m = 1, \dots, e, \\
& \sum_{i=1}^f \sum_{j=1}^r \sum_{k=1}^s \sum_{m=1}^e uq_{ijk} = tot \\
& Cpva_{ijm} = \sum_{k=1}^s pva_{ik} uq_{ijk} \quad i = 1, \dots, f, \quad j = 1, \dots, r, \quad m = 1, \dots, e, \\
& Camm_{ijm} = \sum_{k=1}^s amm_{ik} uq_{ijk} \quad i = 1, \dots, f, \quad j = 1, \dots, r, \quad m = 1, \dots, e, \\
& Cmkp_{ijm} = Mkp_{ij}(Camm_{ijm} + Cpva_{ijm}) \quad i = 1, \dots, f, \quad j = 1, \dots, r, \quad m = 1, \dots, e, \\
& \mu_{lg} = \frac{1}{tot} \sum_{i=1}^f \sum_{j=1}^r cl_{ij} qt_{ij}
\end{aligned} \tag{2.16}$$

$$\begin{aligned}
\mu_{pg} &= \frac{1}{tot} \sum_{i=1}^f \sum_{j=1}^r \sum_{m=1}^e Cpva_{ijm} \\
\mu_{dg} &= \frac{1}{tot} \sum_{i=1}^f \sum_{j=1}^r \sum_{m=1}^e (Cpva_{ijm} + cl_{ij}q_{ijm} + Camm_{ijm} + Cmkp_{ijm}) da_{ijm} \\
\mu_{Mg} &= \frac{1}{tot} \sum_{i=1}^f \sum_{j=1}^r \sum_{m=1}^e Cmkp_{ijm} \\
\mu_{ag} &= \frac{1}{tot} \sum_{i=1}^f \sum_{j=1}^r \sum_{m=1}^e Camm_{ijm}
\end{aligned} \tag{2.17}$$

Capitolo 3

Ottimizzazione ed enumerazione

Nel capitolo precedente sono stati forniti gli strumenti teorici e risolutivi base per affrontare i problemi di ottimizzazione PLMI. In questo capitolo vengono illustrati i quattro approcci risolutivi con cui si soddisfa il requisito di mostrare le n migliori soluzioni, con n definito dall'utente. Saranno introdotti i due solver usati durante il lavoro di tesi per la risoluzione del problema.

3.1 Approcci risolutivi

La specifica di ottenere le n migliori soluzioni comporta risolvere i problemi di ottimizzazione e di *enumerazione* delle soluzioni allo stesso tempo. Esistono in letteratura differenti algoritmi per l'enumerazione, ma risultano avere una complessità computazionale e di memoria NP, aumentando così i tempi e le risorse necessarie nella risoluzione. Gli approcci risolutivi che sono mostrati in seguito, consistono nel risolvere iterativamente il problema usando però dei vincoli aggiuntivi che consentano di modificare il modello alla singola iterazione ed ottenere così la soluzione migliore che segue quella ottenuta all'iterazione precedente. La risoluzione del modello nella singola iterazione, consiste nell'uso di un solver che usa un algoritmo di Branch and Bound o di Branch and Cut.

3.1.1 Primo approccio risolutivo

Un primo modo per generare le soluzioni subottimali che seguono quella ottima, consiste nella riformulazione del modello base con l'aggiunta del seguente vincolo

$$\mu_{pg} + \mu_{ag} + \mu_{lg} + \mu_{dg} + \mu_{Mg} \geq pb + \varepsilon \quad (3.1)$$

e fa sì che il valore della funzione obiettivo deve essere minimizzato, ma non deve coincidere con il valore della funzione obiettivo della soluzione precedentemente trovata, indicato con pb . ε è un valore definito a priori prima dell'esecuzione.

Quando l'utente avrà mandato in esecuzione il software, il sistema richiamerà il solver che fornirà la soluzione ottima alla prima esecuzione, e successivamente il software modificherà il modello base aggiungendo questo vincolo avendo noto il valore della funzione obiettivo e avendo ε definito costante all'interno del codice, oppure regolato in input dall'utente. Tale riformulazione rende la soluzione ottima *non fattibile* (*infeasible*), mentre tutte le altre soluzioni sono *fattibili* (*feasible*), e quindi il solver restituirà la seconda migliore soluzione. Il vincolo 3.1 verrà ridefinito a tutte le iterazioni successive fino a quando o il solver non indica il modello *infeasible* perché il vincolo ad una certa iterazione rende il problema irrisolvibile, oppure quando avrà restituito l'ennesima soluzione. Questa tecnica ha l'enorme vantaggio di introdurre solo una modifica per il modello base e ciò evita che il solver al suo interno non usi strutture dati aggiuntive, o non incrementi le dimensioni delle struttura dati che dovranno essere usate dagli algoritmi di Branch and Bound. Risparmia sia tempo che memoria per l'elaborazione. Tale metodologia introduce degli svantaggi: il primo svantaggio è legato al fatto che l' ε deve essere tarato e deve essere il più piccolo possibile per evitare l'esclusione di certe soluzioni che differiscono molto meno di ε ; il secondo svantaggio consiste nell'esclusione delle *soluzioni multiple equivalenti*. Data la definizione del problema base (il modello), nello spazio delle soluzioni è possibile che alcune configurazioni di impianti aperti e/o di flussi logistici possano avere lo stesso costo medio per componente. Introduzione di altre configurazioni. Queste soluzioni sono molto importanti per gli utenti per poter far fronte alle incertezze della domanda delle componenti sia nell'apertura degli impianti e sia nella variazione dei flussi logistici. Questo primo approccio non consente di avere soluzioni con lo stesso valore di funzione obiettivo, quindi non risulta molto efficace per la risoluzione degli scenari da

affrontare.

3.1.2 Secondo approccio risolutivo

Il secondo approccio consiste sempre nell'aggiunta di un vincolo, in questo caso non lineare, al modello base ad ogni iterazione di ricerca di una soluzione subottima. Il vincolo in questione

$$\sum_{i=1}^f \sum_{j=1}^r \sum_{m=1}^e |q_{ijm} - \bar{q}_{ijm}^{h-1}| \geq slot \quad h = 2, \dots, n \quad (3.2)$$

richiede che la matrice dei pezzi trasportati nelle diverse tratte della soluzione corrente $(q_{i,j,m})$ deve essere differente da quella delle quantità trasportate nella soluzione trovata all'iterazione $h - 1$ $(\bar{q}_{i,j,m}^{h-1})$ per almeno uno *slot* di componenti di una tipologia m in una tratta. L'aggiunta di un vincolo non lineare fa sì che il problema diventi PNLMI (*Programmazione Non Lineare Mista Intera*) che non può essere risolto con un algoritmo di BB, e richiede algoritmi basati sul calcolo del gradiente nello spazio di ricerca. Per evitare questa complicazione è stata usata una tecnica di linearizzazione del modulo assoluto nota in letteratura [3]. Il vincolo 3.2 viene riformulato con i seguenti vincoli

$$\begin{aligned} \Delta_{ijm}^h - q_{ijm} + \bar{q}_{ijm}^{h-1} &\geq 0 & i = 1, \dots, f, j = 1, \dots, r, m = 1, \dots, e, h = 2, \dots, n \\ \Delta_{ijm}^h - q_{ijm} + \bar{q}_{ijm}^{h-1} &\leq mm(1 - \alpha_{ijm}^h) & i = 1, \dots, f, j = 1, \dots, r, m = 1, \dots, e, h = 2, \dots, n \\ \Delta_{ijm}^h - \bar{q}_{ijm}^{h-1} + q_{ijm} &\geq 0 & i = 1, \dots, f, j = 1, \dots, r, m = 1, \dots, e, h = 2, \dots, n \\ \Delta_{ijm}^h - \bar{q}_{ijm}^{h-1} + q_{ijm} &\leq mm\alpha_{ijm}^h & i = 1, \dots, f, j = 1, \dots, r, m = 1, \dots, e, h = 2, \dots, n \\ \sum_{i=1}^f \sum_{j=1}^r \sum_{m=1}^e \Delta_{ijm}^h &\geq slot & h = 2, \dots, n \end{aligned} \quad (3.3)$$

dove: Δ_{ijm}^h è una variabile decisionale aggiuntiva che deve corrispondere al valore assoluto della differenza tra \bar{q}_{ijm}^{h-1} e q_{ijm} ; α_{ijm}^h è una variabile decisionale binaria usata per rendere Δ_{ijm}^h o equivalente a $q_{ijm} - \bar{q}_{ijm}^{h-1}$ oppure a $-q_{ijm} + \bar{q}_{ijm}^{h-1}$. Il quinto vincolo permette di avere gli stessi effetti di 3.2.

Questo approccio permette quindi di escludere in maniera puntuale all'iterazione h ciascuna delle $h - 1$ soluzioni precedenti in base ai flussi logistici ad essi associati. Non

facendo alcuna assunzione sul valore della funzione obiettivo per la soluzione successiva, si riesce a trovare anche le soluzioni multiple equivalenti. Questo approccio risulta molto potente dal punto di vista formale, ma presenta un enorme svantaggio: la crescita del grado di complessità del problema per la ricerca delle soluzioni successive. Dopo aver ottenuto la prima soluzione dal solver, si aggiungono i vincoli 3.3 al modello, che saranno in totale $(4 \times f \times r \times e) + 1$. Alla fine della seconda ricerca questi vincoli non vengono cancellati ma mantenuti, e se ne aggiungono altri $(4 \times f \times r \times e) + 1$. Alla fine della ricerca $n - 1$, il modello per individuare l'ultima soluzione desiderata avrà oltre i vincoli del modello base anche $(n - 1) \times [(4 \times f \times r \times e) + 1]$ vincoli di 3.3. Non aumentano solo il numero di vincoli ma anche le variabili decisionali, dato che vengono aggiunte $2 \times f \times r \times e$. Aggiungere così tanti vincoli e variabili decisionali al modello non solo fa aumentare le dimensioni della struttura dati all'interno dell'algoritmo B&B ma anche il tempo di elaborazione di ciascun nodo generato nell'albero. Tale metodologia è l'esatto opposto della prima in termini di tempo e memoria di calcolo richiesti, ma risolve il problema delle esclusioni di alcune soluzioni subottimali.

3.1.3 Terzo approccio risolutivo

Questo approccio è la combinazione dei primi due, in quanto cerca di mantenere una sorta di stato della ricerca come nella prima metodologia, e cerca di escludere puntualmente delle soluzioni come nella seconda. Quindi dopo aver ottenuto la prima soluzione, dalla seconda ricerca in poi si aggiungono questi vincoli

$$\mu_{pg} + \mu_{ag} + \mu_{lg} + \mu_{dg} + \mu_{Mg} \geq pb \quad (3.4)$$

$$\begin{aligned} \Delta_{ijm}^p - q_{ijm} + \bar{q}_{ijm}^p &\geq 0 & i = 1, \dots, f, j = 1, \dots, r, m = 1, \dots, e \\ \Delta_{ijm}^p - q_{ijm} + \bar{q}_{ijm}^p &\leq mm(1 - \alpha_{ijm}^p) & i = 1, \dots, f, j = 1, \dots, r, m = 1, \dots, e \\ \Delta_{ijm}^p - \bar{q}_{ijm}^p + q_{ijm} &\geq 0 & i = 1, \dots, f, j = 1, \dots, r, m = 1, \dots, e \\ \Delta_{ijm}^p - \bar{q}_{ijm}^p + q_{ijm} &\leq mm\alpha_{ijm}^p & i = 1, \dots, f, j = 1, \dots, r, m = 1, \dots, e \\ \sum_{i=1}^f \sum_{j=1}^r \sum_{m=1}^e \Delta_{ijm}^p &\geq slot \end{aligned} \quad (3.5)$$

dove all'apice delle variabili Δ e α si ha p , che sta a indicare una differenza in modulo

rispetto alla soluzione precedente. Per comprendere al meglio come funziona questo approccio viene illustrato l'algoritmo che parte alla fine della ricerca (e salvataggio) della prima soluzione:

- 0** Si instanzia un $k = 1$.
- 1** Viene prelevato il valore della funzione obiettivo della soluzione appena trovata $f_0(x^k)$ (dove x^k è il vettore delle variabili decisionali all'iterazione k), si pone $pb = f_0(x^k)$, si pone $p = 1$, e si aggiunge al modello base il vincolo 3.4.
- 2** Vengono prelevati i valori delle variabili decisionali $q_{i,j,m}^k$ della soluzione appena trovata, si pone $\bar{q}_{ijm}^p = q_{ijm}^k$, si instanziano nuove variabili decisionali Δ_{ijm}^p e α_{ijm}^p , e si aggiungono al modello riformulato i vincoli riportati in 3.5.
- 3** Si esegue un algoritmo di B&B o di B&C sul modello appena costruito, si salva la soluzione k e si valuta k : se coincide con n si va al passo **5**; altrimenti si prosegue al passo **4**.
- 4** Viene valutato il valore della funzione obiettivo della nuova soluzione: se corrisponde a pb allora si deve incrementare p e k , e si ritorna al passo **2**; se risulta $f_0(x^k) > pb$ si incrementa solo k e si prosegue col passo **1**.
- 5** Alla fine dell'algoritmo sono state generate e salvate le n migliori soluzioni desiderate dall'utente.

In questo modo il vincolo 3.4 effettua un taglio dello spazio di ricerca che esclude tutte le $k - 1$ soluzioni trovate fino a quel momento, esclusa la k -esima. Quindi si aggiungono i vincoli 3.5 che la escludono puntualmente, e il modello così riformulato o trova una soluzione ottima equivalente alla k -esima soluzione per valore di funzione obiettivo ($f_0(x^{k+1}) = f_0(x^k)$), oppure trova una soluzione con valore $f_0(x^{k+1})$ maggiore. Nel primo caso si devono aggiungere al modello dell'iterazione precedente i vincoli 3.5 per escludere la nuova soluzione equivalente, nel secondo caso devono essere riformulati entrambi i vincoli 3.4 e 3.5. I modelli riformulati ad ogni iterazione avranno bisogno di poche variabili decisionali e vincoli aggiuntivi per assicurare la ricerca delle soluzioni multiple equivalenti e delle soluzioni subottime, evitando di aumentare le dimensioni della struttura dati dell'algoritmo B&B e di avere un tempo di elaborazione maggiore per i nodi dell'albero.

3.1.4 Quarto approccio risolutivo

Durante l'attività di tirocinio i ME Manager, dopo aver visualizzato le soluzioni ottenute con la terza metodologia appena mostrata, hanno fornito un ulteriore requisito precedentemente non previsto. Dato che le soluzioni che venivano generate differivano l'una dall'altra per ammontare di pezzi trasportati per delle stesse tratte, allora è stato richiesto di visualizzare le migliori c configurazioni di apertura degli stabilimenti, detti *footprint*, e per ciascuna configurazione si desidera avere le migliori n soluzioni di flussi logistici.

Per soddisfare questo requisito si hanno due possibili strade: generare $2 \times c \times n$ soluzioni, ispezionarne la lista per inserire le soluzioni in sottoliste apposite per ciascun footprint; migliorare l'algoritmo del terzo approccio per far fronte a una ricerca vincolata dall'apertura di impianti specifici. È stata adottata la seconda strada e quindi oltre ad usare i vincoli 3.4 e 3.5, sono stati introdotti anche

$$v_{ik} = v_{ik}^t \iff v_{i,k}^t = 1 \quad \text{per un } t \text{ fissato} \quad (3.6)$$

$$\sum_{(ik) \in NPA^t} v_{ik} \leq |NPA^t| - 1 \quad NPA^t = \{(i,k) : v_{ik}^t = 1\} \quad (3.7)$$

dove: t è l'indice relativo alla ricerca di soluzioni per il t -esimo footprint; v_{ik}^t è il valore che assume la variabile binaria v_{ik} di apertura dell'impianto i con lo scaglione k , quando è stato trovato il t -esimo footprint; NPA^t è l'insieme degli impianti aperti con i relativi scaglioni produttivi col footprint t . Dopo aver trovato la soluzione ottima per il modello base, viene eseguito questo algoritmo per ricercare le soluzioni successive per footprint:

- 0 Porre $t = 1$ e $w = 1$, dove t varia in $\{1, \dots, c\}$ mentre w in $\{1, \dots, s\}$.
- 1 Vengono prelevati i valori di v_{ik} della soluzione appena trovata e li si assegna a v_{ik}^t .
- 2 Per tutti i $(i,k) \in F \times S$ (F insieme degli impianti fornitori e S insieme degli scaglioni) si verifica il valore di v_{ik}^t e se pari a 1 allora (i,k) viene aggiunto a NPA^t .
- 3 Viene prelevato il valore della funzione obiettivo della soluzione appena trovata $f_0(x^{tw})$ (dove x^{tw} è il vettore delle variabili decisionali col footprint t e iterazione w), si pone $pb = f_0(x^{tw})$, si pone $p = 1$, e si aggiunge al modello base il vincolo 3.4.

- 4 Si aggiungono i vincoli riportati in 3.6.
- 5 Vengono prelevati i valori delle variabili decisionali q_{ijm}^{tw} della soluzione appena trovata, si pone $\bar{q}_{ijm}^p = q_{ijm}^{tw}$, si istanziano nuove variabili decisionali Δ_{ijm}^p e α_{ijm}^p , e si aggiungono al modello riformulato i vincoli riportati in 3.5.
- 6 Si esegue un algoritmo di B&B o di B&C sul modello appena costruito, si salva la soluzione w per il footprint t e si valuta w : se coincide con n si va al passo 8; altrimenti si prosegue al passo 7.
- 7 Viene valutato il valore della funzione obiettivo della nuova soluzione: se corrisponde a pb allora si deve incrementare p e w , e si ritorna al passo 5; se risulta $f_0(x^{tw}) > pb$ si incrementa solo w e si prosegue col passo 3.
- 8 Si verifica il valore di t : se corrisponde a c si va al passo 9, altrimenti si prende il modello base e viene aggiunto iterativamente il vincolo 3.7 per ciascun footprint t trovato fino all'istante corrente. In tal caso si esegue il solver su questo modello, si incrementa t di 1, si pone $w = 1$, e si va al passo 1.
- 9 Sono state trovati i migliori c footprint e le migliori n soluzioni per flussi logistici di ciascun footprint.

Con l'uso del vincolo 3.6 si limita la ricerca in un sottospazio molto ridotto per la t -esima configurazione, in quanto il solver sarà costretto a cercare soluzioni che usano quegli impianti specifici, soddisfacendo sempre i vincoli dei flussi logistici diversi e sul valore della funzione obiettivo (3.4 e 3.5). Uno o più vincoli 3.7 saranno usati alla fine della ricerca di n soluzioni per il footprint corrente t , per poter passare alla ricerca del miglior footprint successivo $t + 1$ senza fare alcune assunzioni sui valori di $f_0(x)$. In questo modo si hanno più tagli dello spazio di ricerca ma maggiore libertà nel trovare il sottospazio successivo in cui continuare l'esplorazione delle soluzioni. Non vengono aumentate di troppo il numero di vincoli usati per la ricerca delle soluzioni successive, mantenendo ancora una volta una struttura dati usata dal B&B a dimensioni ridotte per la risoluzione del problema all'iterazione corrente ed altrettanto ridotto sarà il tempo di elaborazione dei nodi.

3.2 Software usati

Durante l'attività di tirocinio sono stati usati 2 software per poter risolvere il problema. In un primo momento è stato adottato **LINGO** della *Lindo Systems Inc.*, ed è un software di ottimizzazione che permette di modellare i modelli con un linguaggio di scripting proprietario di alto livello, e dispone al suo interno differenti moduli solver dedicati alla risoluzione dei problemi Lineari, Non Lineari, Quadratici, Quadratici Vincolati, Semi-definiti, Coni del second'ordine e Stocastici.

In un secondo momento si è incominciato ad usare il software **CPLEX** della *IBM* che fornisce sempre un linguaggio di scripting proprietario per la modellizzazione del problema, noto come *OPL* (Optimization Programming Language), e di differenti moduli solver dedicati alla risoluzione di problemi Lineari, Lineari Misti Interi, Quadratici e Quadratici vincolati. Nei paragrafi che seguono saranno spiegati i meccanismi interni di alto livello di ciascun software usato.

3.2.1 LINGO

Il software dispone di quattro tipi di solver al suo interno:

- diretto;
- lineare;
- non lineare;
- gestore del Branch and Bound.

Quando viene dato a LINGO un file in formato *LG4* contenente il modello di alto livello, questo costruisce internamente una matrice dove le righe sono associati ai vincoli, mentre le colonne sono associate alle variabili. Tale matrice viene mantenuta nella memoria interna del programma Lingo, e viene richiamato inizialmente il solver diretto. Tale solver cerca di assegnare dei valori a quante più variabili decisionali possibili, cercando di usare solamente i vincoli di uguaglianza stretta. Il solver diretto finisce la sua elaborazione, se ha trovato una soluzione per il problema oppure non trova altri vincoli di uguaglianza. In questo caso, LINGO determina quale solver usare in base alla natura dei vincoli del modello matematico. Nel nostro caso verrà richiamato il gestore del B&B che al suo

interno per eseguire la risoluzione del sottoproblema rilassato, userà o il solver lineare o quello non lineare. Il solver lineare di LINGO usa l'algoritmo del simplesso rivisitato con forma del prodotto inversa, mentre il solver non lineare usa o l'algoritmo di linearizzazione successiva *SLP* (Successive Linear Programming) oppure l'algoritmo del gradiente generalizzato ridotto *GRG* (Generalized Reduced Gradient).

Il software LINGO mette a disposizione delle librerie nei diversi linguaggi di programmazione per integrare uno dei solver in applicazioni costruite ad-hoc, e per definire a volo il modello matematico. I linguaggi supportati sono i seguenti:

- Visual C/C++;
- Visual Basic;
- Fortran;
- ASP .NET;
- C# .NET;
- VB .NET;
- Delphi;
- Java.

3.2.2 CPLEX

CPLEX dispone al suo interno più algoritmi per la risoluzione dei problemi lineari:

- simplesso;
- simplesso duale;
- algoritmo del simplesso per le reti;
- metodo dei punti interni (noto anche come metodo delle barriere);
- algoritmo di sifting.

Per la risoluzione di problemi PLI e PLMI si usa il B&C con il semplice duale, mentre per i problemi quadratici, a seconda della loro complessità, viene usato o il metodo delle barriere oppure il B&C. CPLEX, come LINGO, definisce il modello del problema all'interno della memoria come se fosse una matrice le cui righe corrispondono ai vincoli mentre le colonne coincidono con le variabili decisionali. Questo software in base al modello passato, determina automaticamente il tipo di problema e manda in esecuzione fino a 8 thread in contemporanea, ognuno dei quali esegue uno degli algoritmi menzionati in precedenza. Ciò permette di individuare la soluzione ottima in tempi rapidi. CPLEX permette di lavorare su tre livelli di astrazione differenti:

- OPL per la prima fase di analisi e modellazione del problema;
- Concert Technology è un insieme di librerie implementate in Java, C# e C++ che forniscono la possibilità di costruire il modello dentro al codice di un applicativo che usa uno dei solver di CPLEX;
- Callable Library è una libreria scritta in C che permette di costruire il modello all'interno dell'applicazione e di gestire l'allocazione dinamica della memoria per le varie matrici e vettori utilizzati dai solver;

L'utilizzo di uno di questi strumenti comporta vantaggi e svantaggi. L'OPL permette di effettuare rapidamente lo sviluppo del modello matematico ma non è predisposto all'integrazione in un'applicazione. La Concert Technology consente l'integrazione rapida del modello nel codice applicativo con una grande facilità d'utilizzo, a scapito della gestione della memoria e dei tempi di elaborazione. La Callable Library colma la lacuna dell'uso efficiente della memoria ma richiede un certo sforzo nella programmazione in quanto si deve lavorare a basso livello rispetto alle librerie nei linguaggi OOP(Object Oriented Programming). Il programmatore nello sviluppo dell'applicazione di ottimizzazione deve trovare un giusto compromesso tra tempi di sviluppo ed efficienza del programma in esecuzione.

CPLEX JAVA/NET API (from oplall.jar and oplall.dll)		
CPLEX C++ API	CPLEX JAVA/.NET API (from cplex.jar and ILOG.CPLEX.dll)	Other CPLEX APIs and connectors
CPLEX Callable Libraries		

Figura 3.1: Librerie di Cplex che operano su differenti livelli di linguaggio

Capitolo 4

Prove computazionali

In quest'ultimo capitolo vengono mostrati i risultati relativi alle varie prove di esecuzione dei due software per poter determinare le prestazioni in termini di tempo di esecuzione e di memoria richiesta per affrontare i diversi problemi. Saranno mostrati anche i risultati relativi alla capacità dei programmi di trovare le soluzioni corrette per ciascun problema.

4.1 Descrizione della metodologia adottata

Dati i software LINGO e CPLEX, rispettivamente nelle versioni 17 e 12.7.1, bisogna determinare quale dei due sistemi risulta migliore per lo sviluppo di un'applicazione che possa essere rilasciata ai manager, in modo tale da poter condurre le loro attività di pianificazione in totale autonomia. Per misurare le prestazioni si hanno a disposizione quattro istanze di dati, dove ciascuna istanza è l'insieme di tutti i dati di input che vengono normalmente inseriti nell'interfaccia utente che caratterizzano un problema da affrontare. Le prime tre istanze di dati sono state inventate appositamente in modo che ciascuna disponga di un numero limitato di soluzioni, che sono rispettivamente 33, 52 e 118. La quarta istanza di dati appartiene ad un caso reale, la cui complessità prevede un numero elevato di soluzioni. Pertanto per questa istanza si è deciso di ricercare solo le 100 migliori soluzioni. Le prove sono state condotte su un notebook Asus a 64 bit, che dispone di un processore Intel Core i7-4710HQ a 2.5 GHz quando opera normalmente, e con la tecnologia Turbo Boost raggiunge i 3.45 Ghz. La memoria RAM a disposizione è di 7.89 GB con

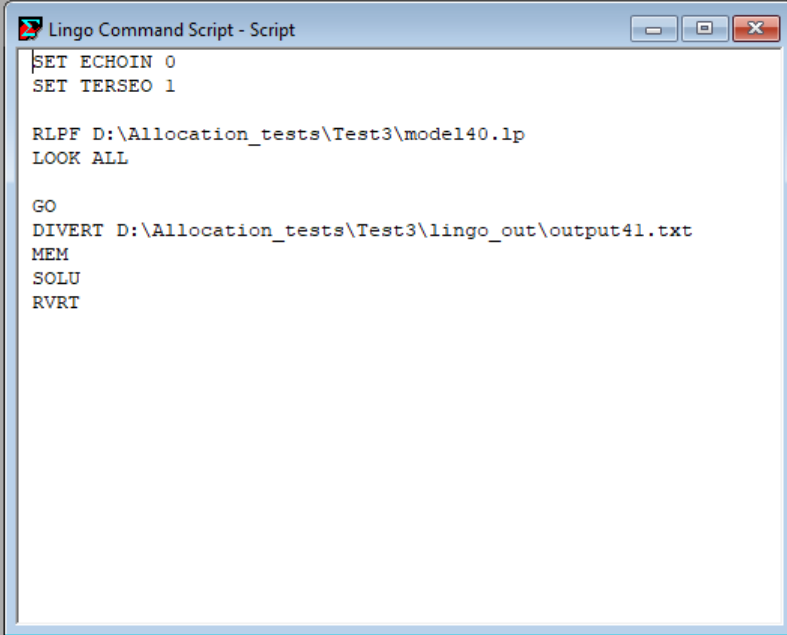
frequenza operativa di 1600 MHz. Prima di avviare le prove, il processore operava a 3.45 GHz ed era usato al 20% dai processi del Sistema Operativo Windows 10, con memoria RAM occupata a 3.3 GB. Non erano attivi altri processi utente e alcuna connessione a Internet per evitare che qualche processo di rete potesse interferire con l'uso della CPU.

4.1.1 Approccio risolutivo adottato

Dopo aver discusso nel Capitolo 3 i vantaggi e gli svantaggi di ciascun approccio, si è deciso di utilizzare il terzo approccio risolutivo, che consente una ricerca globale delle migliori n soluzioni, avendo una memoria costante per la creazione del modello preposto per la ricerca della soluzione successiva (tranne nei casi in cui sono state trovate soluzioni multiple equivalenti). Il modello matematico presente alla fine del Capitolo 2 è stato tradotto nel linguaggio di scripting OPL, e tramite la libreria di CPLEX in C# è stato importato il modello OPL per poi esportarlo in un file col formato standard LP. I file in formato LP sono uno standard industriale per l'esportazione dei modelli matematici di Programmazione Lineare, per effettuare le risoluzioni dei problemi su solver differenti. Utilizzando un programma implementato in C# che usa l'algoritmo del terzo approccio e le librerie di CPLEX, vengono generati i modelli che servono per la ricerca dell' n -esima soluzione e ciascuno di questi modelli sono salvati sempre nei file in formato LP. Questi modelli verranno dati singolarmente ai due solver per poter misurare la memoria e il tempo di risoluzione impiegati per ciascun modello.

4.1.2 Misure delle prestazioni in LINGO

Per determinare i tempi e la memoria di elaborazione impiegati dal solver di LINGO per un singolo modello, è stato usato lo script in linguaggio proprietario nella Figura 4.1. Il comando SET ECHOIN visualizza o meno all'interno della finestra di comando del software, tutto l'input che viene passato (incluso i file), mentre SET TERSEO permette di avere un *solution report* (un rapporto sull'esecuzione e sulla soluzione trovata) breve o dettagliato. Il comando RLPF serve per aprire il file LP e caricarlo all'interno del software. GO avvia il solver per la risoluzione del modello caricato, DIVERT permette di trasferire tutti i dati di output generati dal solver verso un file di testo, MEM e SOLU stampano in output la memoria usata dal solver per quella esecuzione e il *solution report*. Il comando RVRT chiude i file di testo che è stato aperto. Nei dettagli della memoria usata, si ha a



```
Lingo Command Script - Script
SET ECHOIN 0
SET TERSEO 1

RLPF D:\Allocation_tests\Test3\model40.lp
LOOK ALL

GO
DIVERT D:\Allocation_tests\Test3\lingo_out\output41.txt
MEM
SOLU
RVRT
```

Figura 4.1: File col script LINGO per il salvataggio del *solution report* e della soluzione in un file di testo apposito

disposizione il picco di memoria generata durante l'esecuzione del solver, e la memoria effettivamente usata per il modello.

4.1.3 Misure delle prestazioni in CPLEX

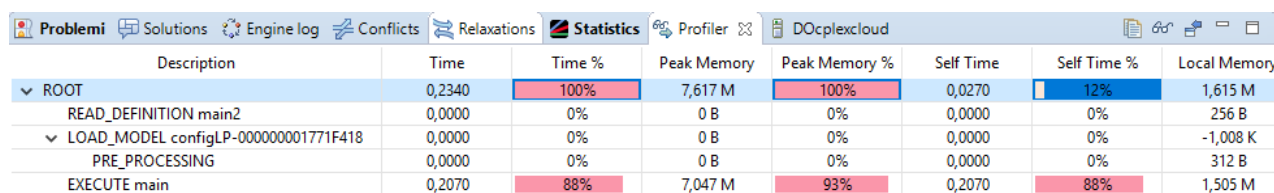
Per CPLEX analogamente si usa uno script OPL che preleva il modello e ne restituisce la soluzione ottima all'interno dell'ambiente CPLEX Studio IDE. In questo ambiente di sviluppo dei modelli, c'è una sezione apposita per il profiler, come riportato nella Figura 4.3, che restituisce tutti i dettagli relativi all'uso della memoria e al tempo di esecuzione impiegati per la risoluzione del problema. *Peak Memory* sta a indicare la memoria massima generata per la risoluzione del problema che risulta molto variabile per l'esecuzione di uno stesso modello, mentre *Local Memory* misura quanta memoria è stata effettivamente usata per il modello.

```

1  /*****
2  * OPL 12.7.1.0 Model
3  * Author: Marco
4  * Creation Date: 18/feb/2018 at 17:33:56
5  *****/
6  main{
7      cplex.importModel("D:\\Allocation_tests\\Test4\\model99.lp");
8      cplex.solve();
9      writeln(cplex.getObjValue());
10
11 }
12

```

Figura 4.2: File col script OPL per la risoluzione del modello



Description	Time	Time %	Peak Memory	Peak Memory %	Self Time	Self Time %	Local Memory
ROOT	0,2340	100%	7,617 M	100%	0,0270	12%	1,615 M
READ_DEFINITION main2	0,0000	0%	0 B	0%	0,0000	0%	256 B
LOAD_MODEL configLP-000000001771F418	0,0000	0%	0 B	0%	0,0000	0%	-1,008 K
PRE_PROCESSING	0,0000	0%	0 B	0%	0,0000	0%	312 B
EXECUTE main	0,2070	88%	7,047 M	93%	0,2070	88%	1,505 M

Figura 4.3: Profiler dell'ambiente CPLEX Studio IDE

4.2 Risultati della prima istanza di dati

Il primo scenario ha le seguenti caratteristiche: si hanno tre impianti produttori e tre stabilimenti clienti, dove i clienti richiedono una sola tipologia di componente. Si hanno tre scaglioni produttivi distinti e delle nove possibili combinazioni di scaglioni produttivi e fornitori, l'utente permette l'attivazione solo di cinque combinazioni. Queste caratteristiche strutturali implicano che il problema dispone di 33 possibili soluzioni in termini di apertura di impianti e flussi logistici.

4.2.1 Tempi di esecuzione

Come si può osservare nella Figura 4.4, CPLEX risulta mediamente migliore per la velocità di elaborazione di ogni singolo problema, tranne per due esecuzioni. LINGO risulta quasi sempre più lento. Sono stati registrati per questo caso dei tempi medi di risoluzione, rispettivamente per CPLEX e per LINGO, di 88 ms e di 206 ms. Le rispettive deviazioni standard sono di 77 ms e di 29 ms, a conferma del fatto che i tempi di esecuzione di CPLEX sono fortemente variabili rispetto a LINGO. I tempi di massimo e di minimo di

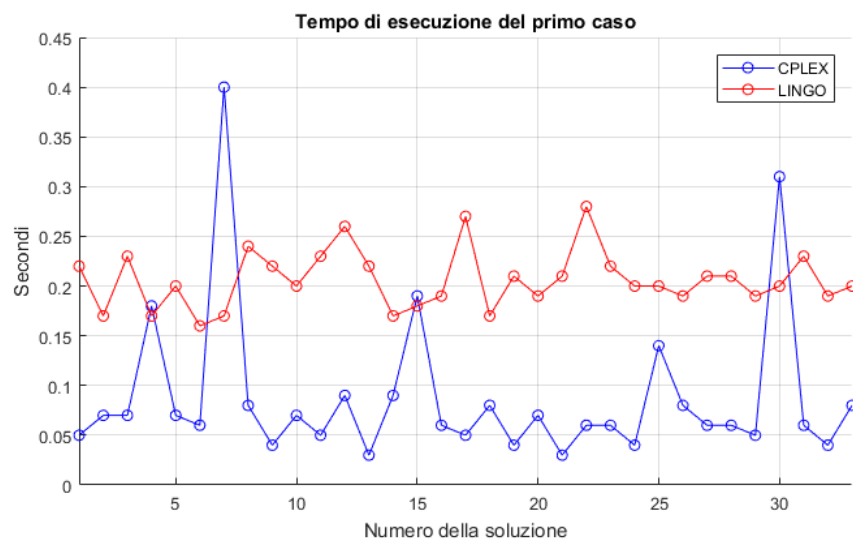


Figura 4.4: Tempi di esecuzione della prima istanza per ciascuna soluzione ricercata

CPLEX sono di 400 ms e 30 ms. Invece per LINGO si ha un massimo di 280 ms e un minimo di 160 ms.

4.2.2 Memoria

Osservando la Figura 4.5, si nota che entrambi i software usano poca memoria per il modello iniziale passato per la ricerca dell'ottimo, e successivamente grazie all'aggiunta dei vincoli di esclusione delle soluzioni precedenti, si ha un solo incremento della memoria necessaria, dopodiché rimane costante dalla seconda esecuzione per tutte quelle successive. La memoria usata alla prima esecuzione, rispettivamente per CPLEX e LINGO, è di 212.094 KB e di 408.5 KB. Dalla seconda esecuzione in poi CPLEX mantiene una memoria costante di 266.32 KB, mentre per LINGO si ha una memoria allocata poco variabile che mediamente corrisponde a 510.108 KB con una deviazione standard di 0.482 KB. Durante l'esecuzione di questi modelli sono stati registrati dei picchi di allocazione di memoria di 6.32 MB per CPLEX e di 716.469 KB per LINGO.

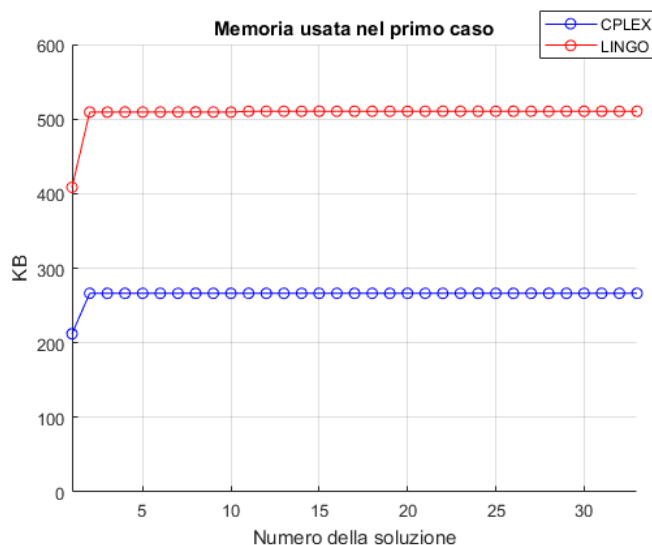


Figura 4.5: Memoria usata per la risoluzione delle diverse soluzioni della prima istanza

4.3 Risultati della seconda istanza di dati

Nel secondo scenario si hanno quattro impianti produttori e due stabilimenti clienti, e questi richiedono sempre una sola tipologia di componente. Si hanno quattro scaglioni produttivi distinti e delle sedici possibili combinazioni di scaglioni produttivi e fornitori, l'utente permette l'attivazione di otto combinazioni. Date le caratteristiche strutturali del problema, si hanno a disposizione 52 possibili soluzioni in termini di apertura di impianti e flussi logistici.

4.3.1 Tempi di esecuzione

Come si può osservare nella Figura 4.6, CPLEX presenta per questo scenario un numero maggiore di picchi di tempi di esecuzione rispetto al caso precedente confermando che tempi di risoluzione del software sono fortemenete variabili. Anche in questo caso LINGO risulta mediamente più lento. Sono stati registrati i seguenti tempi medi di risoluzione, rispettivamente per CPLEX e per LINGO, di 122 ms e di 264 ms. Le rispettive deviazioni standard sono di 108 ms e di 80 ms, e ciò implica che LINGO risulta più stabile nell'esecuzione. I tempi di massimo e di minimo di CPLEX sono di 510 ms e 30 ms. Invece per LINGO si ha un massimo di 560 ms e un minimo di 160 ms.

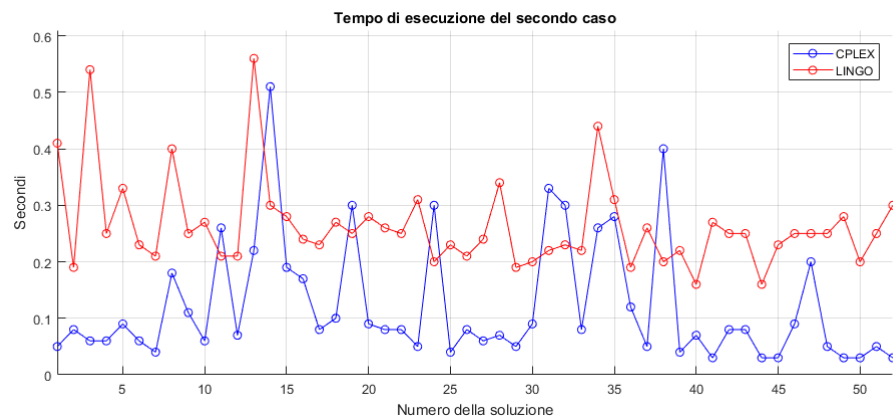


Figura 4.6: Tempi di esecuzione della seconda istanza per ciascuna soluzione ricercata

4.3.2 Memoria

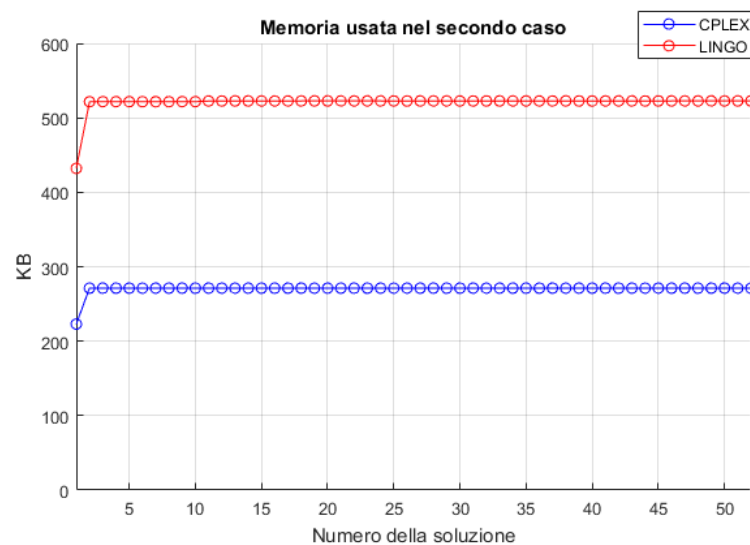


Figura 4.7: Memoria usata per la risoluzione delle diverse soluzioni della seconda istanza

Anche nella Figura 4.7 si nota che entrambi i software usano poca memoria per il modello iniziale, successivamente si ha un incremento della memoria, dopodiché rimane costante dalla seconda esecuzione per tutte quelle successive. La memoria usata alla prima esecuzione, per CPLEX e LINGO, è di 222.953 KB e di 432.109 KB. Dalla seconda esecuzione in poi CPLEX mantiene una memoria costante di 271.211 KB, mentre per LINGO si ha

una memoria allocata poco variabile che mediamente corrisponde a 522.5 KB con una deviazione standard di 0.372 KB. Durante l'esecuzione di questi modelli sono stati registrati dei picchi di allocazione di memoria di 6.148 MB per CPLEX e di 723.758 KB per LINGO.

4.4 Risultati della terza istanza di dati

Nella terza istanza si hanno quattro impianti produttori e tre stabilimenti clienti con richieste di una sola tipologia di componente. Si hanno tre scaglioni produttivi distinti e delle dodici possibili combinazioni di scaglioni produttivi e fornitori, l'utente consente l'attivazione di sette combinazioni. Date le caratteristiche strutturali del problema, si hanno a disposizione 118 possibili soluzioni.

4.4.1 Tempi di esecuzione

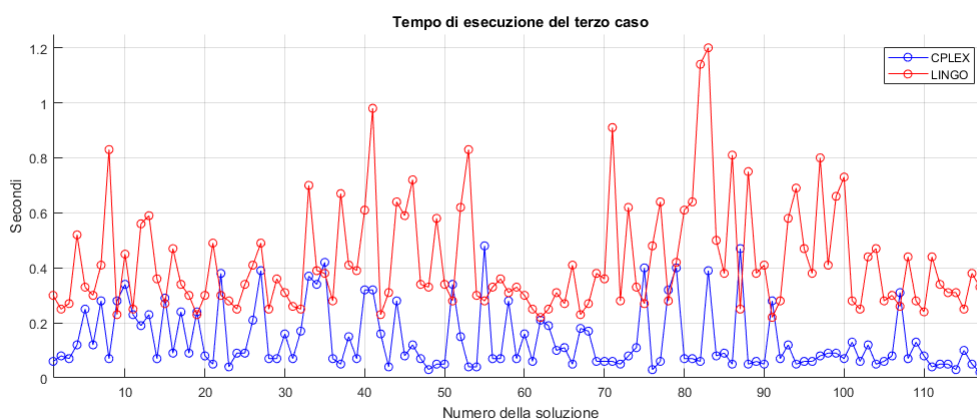


Figura 4.8: Tempi di esecuzione della terza istanza per ciascuna soluzione ricercata

Per questo scenario, osservando la Figura 4.8, CPLEX presenta sempre un grande numero di picchi per i tempi di esecuzione, ma questa volta molto di questi sono inferiori rispetto ai tempi di risoluzione di LINGO. Per questo caso LINGO risulta mediamente molto più lento e infatti è stato registrato un tempo medio di risoluzione di 421 ms con una deviazione standard di 197 ms. CPLEX invece presenta una media di 142 ms e una deviazione standard di 115 ms. Dati questi risultati, si può affermare che LINGO non riesce

a mantenersi stabile per i tempi di risoluzione quando aumenta cardinalità dell'insieme di soluzioni ammissibili.

4.4.2 Memoria

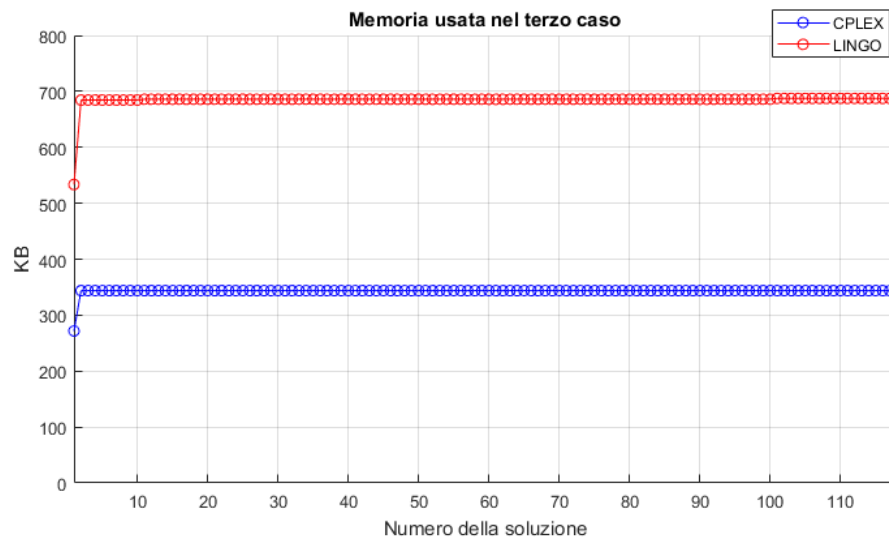


Figura 4.9: Memoria usata per la risoluzione delle diverse soluzioni della terza istanza

Nella Figura 4.9 si nota che il divario di memoria richiesta tra CPLEX e LINGO aumenta notevolmente rispetto alle Figure 4.5 e 4.7. La memoria usata alla prima esecuzione, per CPLEX e LINGO, è di 271.695 KB e di 533.523 KB. Dalla seconda esecuzione in poi CPLEX mantiene una memoria costante di 343.828 KB, mentre per LINGO si ha una memoria allocata poco variabile che mediamente corrisponde a 686.066 KB con una deviazione standard di 0.700 KB. Durante l'esecuzione di questi modelli sono stati registrati dei picchi di allocazione di memoria di 6.84 MB per CPLEX e di 913.336 KB per LINGO.

4.5 Risultati della quarta istanza di dati

Il quarto caso è un problema reale in cui si hanno cinque impianti produttori e nove stabilimenti clienti. Si hanno sette scaglioni produttivi distinti per la produzione di due tipi di componenti e delle trentacinque possibili combinazioni di scaglioni produttivi e

fornitori, l'utente consente l'attivazione di quindici combinazioni. Ciascun cliente inoltre effettua la richiesta di un solo tipo di componente, e pertanto quattro clienti richiedono pezzi del primo tipo, mentre gli altri cinque richiedono pezzi del secondo tipo. Nei casi precedenti tutte le tratte tra fornitori e clienti erano abilitate, mentre in questo scenario si hanno quattordici tratte inibite per il trasporto delle quarantacinque possibili. Questo problema risulta molto più complesso di quelli precedenti, e per stimare il numero delle soluzioni ammissibili sarebbe richiesto un algoritmo apposito di enumerazione. Pertanto si è deciso di studiare il comportamento dei due software solo sulle prime cento soluzioni.

4.5.1 Tempi di esecuzione

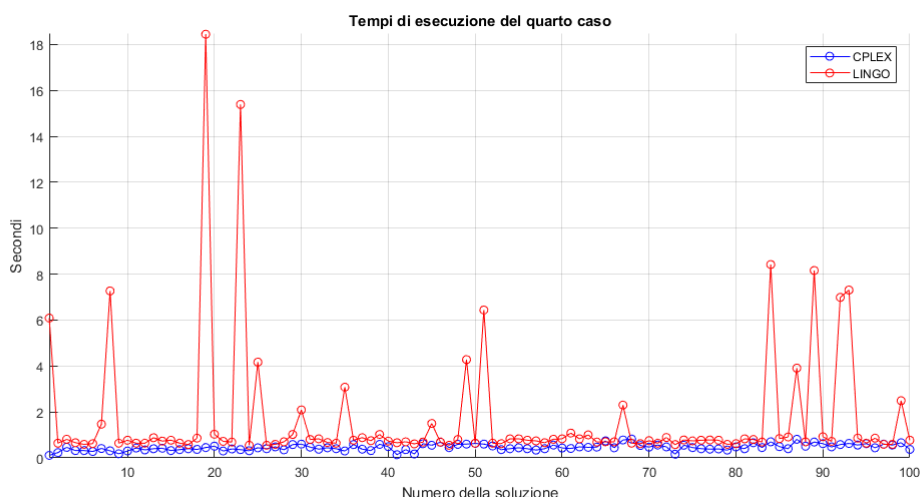


Figura 4.10: Tempi di esecuzione della quarta istanza per ciascuna soluzione ricercata

Nella Figura 4.10 si nota che LINGO presenta dei picchi nei tempi di elaborazione, quando ricerca certe soluzioni, molto più alti rispetto a CPLEX che in questo scenario non presenta picchi considerevoli. In tale scenario LINGO risulta troppo lento rispetto a CPLEX e infatti è stato registrato un tempo medio di risoluzione di 1.701 s con una deviazione standard di 2.802 s. CPLEX invece presenta una media di 458 ms e una deviazione standard di 141 ms. Si può affermare che LINGO presenta difficoltà a risolvere problemi con questo grado di complessità in tempi brevi.

4.5.2 Memoria

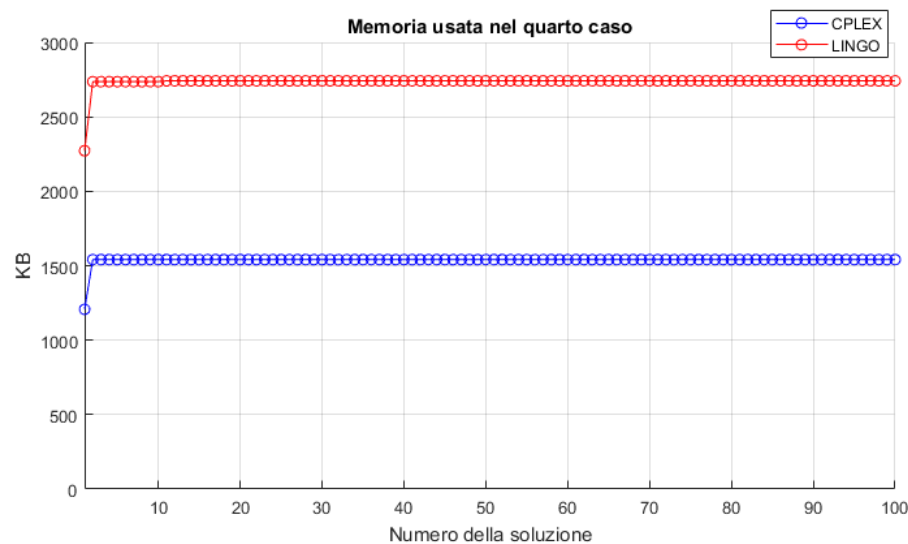


Figura 4.11: Memoria usata per la risoluzione delle diverse soluzioni della quarta istanza

Per l'andamento della memoria di questo scenario, mostrato nella Figura 4.11, il divario cresce ulteriormente tra questi due software, facendo di CPLEX il risolutore meno oneroso. La memoria usata alla prima esecuzione, per CPLEX e LINGO, è di 1.179 MB e di 2.219 MB. Dalla seconda esecuzione in poi CPLEX mantiene una memoria costante di 1.505 MB, mentre per LINGO si ha una memoria allocata poco variabile che mediamente corrisponde a 2.677 MB con una deviazione standard di 1.573 MB. Durante l'esecuzione di questi modelli sono stati registrati dei picchi di allocazione di memoria di 13.773 MB per CPLEX e di 3.57 MB per LINGO.

4.6 Precisione dei software

In questo paragrafo si mostra il grado di precisione dei due software, e in questo contesto s'intende la capacità di trovare la soluzione ottima non solo per il modello base, ma anche per tutte le versioni del modello modificato durante la ricerca delle soluzioni sub-ottime.

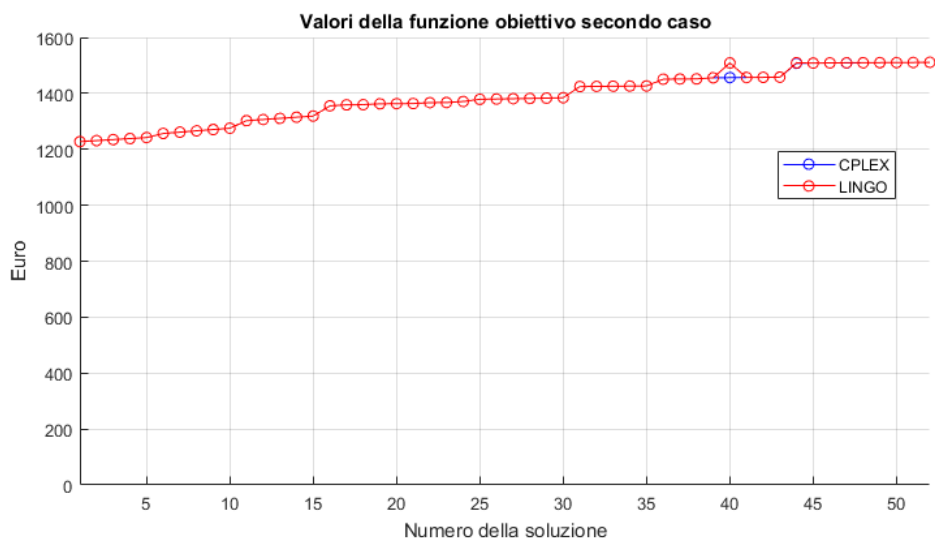


Figura 4.12: Sequenza delle soluzioni ammissibili del secondo caso.

4.6.1 Risultati

Per la prima istanza di dati entrambi i software sono in grado di trovare tutte le 33 soluzioni previste per quel problema. Per il secondo scenario invece delle 52 soluzioni possibili, LINGO non fornisce per tre esecuzioni le soluzioni aspettate a differenza di CPLEX. Difatti il valore della funzione obiettivo per la 40^a, 44^a e 47^a soluzione si ha una differenza rispettivamente di 52.544, 1.608 e 0.514. Questi due ultime differenza di risultati non sono percettibili nel grafico riportato in Figura 4.12.

Per quanto riguarda alla terza istanza LINGO fornisce una sola soluzione non accurata nella ricerca della 29^a soluzione che differisce da quella fornita da CPLEX di 3.136. Si può osservare l'andamento della funzione obiettivo per questo scenario nella Figura 4.13, anche se non è visibile l'errore.

Per l'ultimo caso analizzato, osservando la Figura 4.14, LINGO in molte esecuzioni non restituisce le stesse soluzioni ottimali di CPLEX, che sono 92 casi su 100. Restituisce le stesse soluzioni ottime di CPLEX solo nella 1^a, 8^a, 19^a, 23^a, 49^a, 51^a, 61^a e 84^a. In alcuni casi LINGO fornisce soluzioni il cui valore della funzione obiettivo raggiunge valori di 50000 euro, che sono inaccettabili in quanto si utilizzano delle tratte per soddisfare la domanda del cliente che sono state inibite ponendo costi logistici di 1 milione di euro per

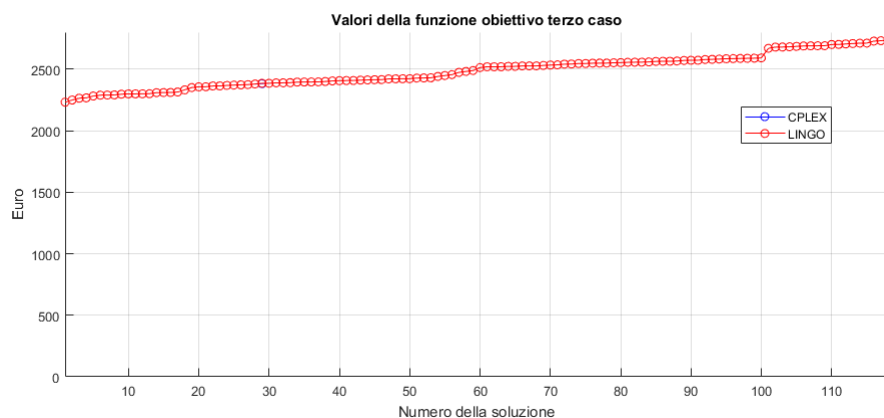


Figura 4.13: Sequenza delle soluzioni ammissibili del terzo scenario.

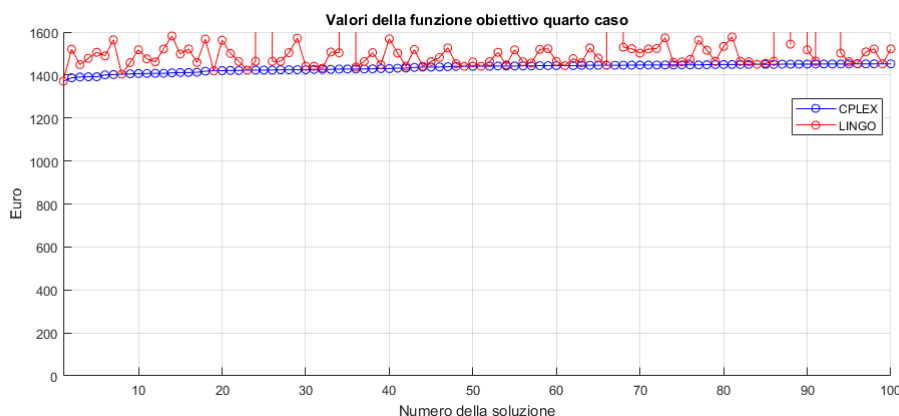


Figura 4.14: Sequenza delle soluzioni ammissibili per la quarta istanza.

pezzo.

Il motivo per cui LINGO non riesce a fornire le stesse soluzioni di CPLEX in alcuni casi, è legato alla taratura di alcuni parametri operativi del solver interno a LINGO. Osservando la Figura 4.15 per i problemi PLMI bisogna regolare opportunamente le tolleranze di fattibilità delle soluzioni (*Initial Linear Feasibility Tol.* e *Final Linear Feasibility Tol.*) e di ottimalità (*Linear Optimality Tol.*) per il solver lineare affinché, durante il rilassamento del problema nel B&B, possa trovare le soluzioni e l'ottimo nello spazio di ricerca e non scartarle per via di una violazione dei vincoli per valori maggiori alle tolleranze impostate. Analogamente per il gestore della parte intera del solver (*Integer Solver*) si devono regolare i parametri di integrità assoluta e relativa (*Absolute Integrality* e *Relative*

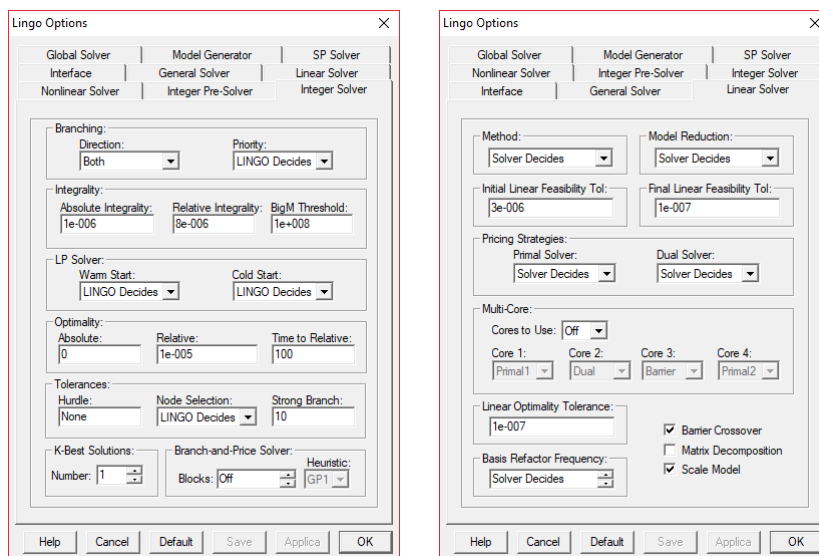


Figura 4.15: Finestre per tarare i parametri di LINGO

Integrity) che sono sempre delle tolleranze per accettare dei valori delle variabili decisionali come interi o meno. Al diminuire del valore di tutti questi parametri mostrati, si aumenta il tempo di ricerca e la probabilità che il solver scarti certe soluzioni perché sono state violate le tolleranze. Al contrario, aumentando questi valori si rischia di accettare delle soluzioni in cui i valori delle variabili decisionali intere risultino reali, o si ottengano delle soluzioni che in realtà sono infattibili. Chi usa il software LINGO deve regolare opportunamente questi parametri quando aumenta la complessità del problema, al fine di migliorare la precisione. Questa operazione risulta di per sé molto dispendiosa dal punto di vista del tempo impiegato per la taratura. Inoltre una volta trovati i valori corretti dei parametri per tutte le soluzioni di un problema, non è detto che siano adeguati anche per un altro problema che ha la stessa struttura del modello base, ma valori dei coefficienti completamente diversi all'interno dei vincoli. Se per un modello differente si ha un elevato numero di soluzioni fattibili, facendo diverse prove con parametri differenti, in alcune esecuzioni si trovano le soluzioni ricercate a certe iterazioni, e per altre iterazioni in cui si ottenevano certe soluzioni desiderate nelle prove precedenti, ci si ritrova risultati differenti.

CPLEX al contrario non ha bisogno di una taratura dei parametri delle tolleranze dei solver prima dell'esecuzione, in quanto internamente, quando acquisisce il modello e lo analizza, determina i valori corretti di tutti questi parametri, liberando il programmatore da

questo onere. Per la ricerca delle migliori soluzioni si desidera evitare la taratura del solver per ogni problema, in modo che i manager possano lavorare ad alto livello sul sistema in maniera autonoma senza dover conoscere questi dettagli e fare delle regolazioni.

4.7 Analisi dei risultati

In questo paragrafo vengono analizzate le migliori soluzioni trovate, in termini di apertura degli impianti e impatto dei costi, per ciascuna problema affrontato.

4.7.1 Prima istanza

Nel primo problema si hanno tre impianti produttori, di cui due appartenenti alla stessa regione, che la denoteremo con α mentre l'altra con β . I tre impianti clienti invece sono localizzati in regioni differenti tra loro, di cui uno appartiene alla regione θ , un altro alla regione ϕ , e mentre l'ultimo appartiene ad α . Di conseguenza, nelle prime nove migliori soluzioni trovate, si aprono i due impianti della regione α per soddisfare le domande degli stabilimenti di assemblaggio, facendo variare tra una soluzione e un'altra l'ammontare di pezzi trasportati su ciascuna rotta. Nelle altre soluzioni, si vede l'apertura sempre di due stabilimenti di cui uno della regione α e l'altra della region β , per far sì che il primo possa soddisfare sempre la domanda dell'impianto cliente della stessa regione e insieme all'altro produttore va a soddisfare le richieste degli altri due clienti. Per quanto riguarda l'impatto dei costi, si può far riferimento alla tabella riassuntiva seguente

Tipo di costo	Perc. Min	Perc. Max
μ_{pg}	81.73 %	87.10 %
μ_{ag}	4.49 %	6.25 %
μ_{lg}	1.33 %	2.91 %
μ_{dg}	0.65 %	2.10 %
μ_{Mg}	4.67 %	8.77 %

Tabella 4.1: Percentuali minime e massime dei diversi costi medi rispetto al costo medio globale per il primo problema

e si può ovviamente concludere che il costo medio di produzione e dei materiali richiesti è particolarmente incisivo rispetto ai costi logistici e dei dazi.

4.7.2 Seconda istanza

Nel secondo problema si hanno quattro impianti produttori, di cui due appartenenti alla stessa regione α mentre gli altre due appartengono alla regione β . I due impianti clienti sono localizzati nella regione α . La migliore delle configurazioni di apertura impianti prevede l'apertura di un impianto α e di un impianto β per rifornire di componenti gli impianti di assemblaggio. Risulta come seconda migliore soluzione, l'apertura di tutti gli impianti nella regione α contrariamente all'aspettativa di migliore soluzione globale. Ciò è legato al fatto che i costi di produzione del migliore impianto in β scelto, sono decisamente più bassi. La terza soluzione di apertura risulta simile alla prima, ma viene cambiato lo stabilimento in β . La quarta migliore configurazione presenta l'apertura di un impianto nella regione α e di tutti gli impianti nella regione β . La quinta soluzione prevede l'apertura di un solo stabilimento nella regione β la cui capacità produttiva è tale da soddisfare le domande degli stabilimenti clienti in α . La sesta soluzione consiste nell'apertura dei soli impianti in β . Di seguito viene riportata l'incidenza delle diverse tipologie di costi medi sul costo medio globale.

Tipo di costo	Perc. Min	Perc. Max
μ_{pg}	76.07 %	81.42 %
μ_{ag}	5.74 %	10.87 %
μ_{lg}	2.62 %	3.03 %
μ_{dg}	1.39 %	1.44 %
μ_{Mg}	8.63 %	8.78 %

Tabella 4.2: Percentuali minime e massime dei diversi costi medi rispetto al costo medio globale per il secondo problema

Anche in questo scenario il costo medio di produzione e dei materiali richiesti incide molto sul costo medio globale. Si osserva che il costo medio di Markup non ha un ampio intervallo di variazione in percentuale.

4.7.3 Terza istanza

Nel terzo problema si hanno quattro impianti produttori, di cui due appartenenti alla stessa regione α mentre un impianto appartiene alla regione β , e un altro ancora alla regione γ . I tre stabilimenti di assemblaggio sono localizzati, rispettivamente, nelle regioni α , β e

ϕ . La migliore soluzione consiste nell'apertura di un impianto nella regione α e un altro nella regione β . La seconda soluzione consiste nell'apertura di entrambi gli stabilimenti produttori nella regione α . Nella terza soluzione si apre un altro impianto della regione α e lo stesso impianto della regione β . La quarta soluzione richiede tre impianti, ognuno di una regione diversa di produzione. La quinta soluzione consiste nell'apertura di uno stabilimento nella regione α e uno nella regione γ . La sesta soluzione è analoga, solo che cambia l'impianto della regione α . La settima soluzione è simile alla quarta ma cambia lo stabilimento della regione α . Le soluzioni individuate per la terza istanza di dati, fa sì che

Tipo di costo	Perc. Min	Perc. Max
μ_{pg}	80.81 %	84.06 %
μ_{ag}	4.44 %	4.92 %
μ_{lg}	2.02 %	4.00 %
μ_{dg}	0 %	2.08 %
μ_{Mg}	8.67 %	9.00 %

Tabella 4.3: Percentuali minime e massime dei diversi costi medi rispetto al costo medio globale per il terzo scenario

all'apertura di opportuni impianti produttori, si riesce ad avere un costo nullo per i dazi riducendone l'impatto sul costo totale. I costi di ammortamento degli investimenti solo in questo scenario hanno una bassa influenza rispetto ai casi precedenti. I costi logistici incidono di più dato che si hanno più rotte e in certi scenari un maggior numero di impianti produttori.

4.7.4 Quarta istanza

Per il quarto problema si hanno cinque impianti produttori, di cui due appartenenti alla stessa regione α mentre altri due appartengono alla regione β , e l'ultimo stabilimento è localizzato nella regione γ . I nove stabilimenti di assemblaggio sono ripartiti, rispettivamente, cinque nella regione α , tre nella regione β e un impianto nella regione δ . La migliore soluzione, in termini di apertura impianti, consiste nell'apertura di due impianti, uno nella regione α e un altro nella regione β . La seconda soluzione consiste nell'apertura di tre impianti, di cui due nella regione α , e uno nella regione β . La terza soluzione è simile alla seconda ma vengono cambiati i livelli produttivi. La quarta soluzione richiede

sempre tre impianti, di cui uno in α e due nella regione β . La quinta soluzione è analoga alla seconda ma gli scaglioni produttivi attivati risultano diversi. La sesta soluzione usa gli stessi impianti e prevede l'apertura di un ulteriore impianto in β . La settima soluzione è analoga alla precedente ma cambiano gli scaglioni produttivi attivati. Anche l'ottava usa gli stessi impianti, ma con livelli produttivi maggiori in un impianto localizzato nella regione β . Per il quarto problema, con l'opportuna apertura degli impianti produttori, si

Tipo di costo	Perc. Min	Perc. Max
μ_{pg}	82.63 %	84.47 %
μ_{ag}	5.35 %	5.73 %
μ_{lg}	1.49 %	2.12 %
μ_{dg}	0 %	4.20 %
μ_{Mg}	5.70 %	8.30 %

Tabella 4.4: Percentuali minime e massime dei diversi costi medi rispetto al costo medio globale per il quarto scenario

riesce ad avere un costo nullo per i dazi, ma l'intervallo di variazione risulta maggiore rispetto ai casi precedenti. Anche i costi di ammortamento degli investimenti presentano una bassa variabilità. Ad incidere in maniera significativa, dopo i costi di produzione, sono solo i costi di Markup.

4.7.5 Sintesi

A fronte delle quattro istanze di dati usate nelle prove, e in base ai valori delle diverse tipologie di costi forniti in ingresso, è possibile affermare che il risolutore cerca di restituire delle configurazioni di apertura di impianti che tendono a minimizzare i costi di produzione e dei materiali richiesti per la fabbricazione dei componenti. I costi logistici e dei dazi hanno un bassissimo impatto, in generale, nei diversi scenari, e ciò spiega il motivo per cui vengono escluse certe soluzioni che sembrano ovvie per i bassi costi di trasporto. Anche se ci sono degli stabilimenti con elevata capacità produttiva, che possono scalare facilmente il costo di produzione, la tendenza è quella di aprire almeno due impianti in modo da avere una spesa di produzione che in totale risulta inferiore a quella relativa all'apertura di un solo impianto. Ovviamente questa non è la regola, perché possono esistere nella realtà impianti che possono soddisfare pienamente la domanda dei pezzi a

bassissimo costo di produzione. Dipende fortemente dal problema che viene affrontato e quindi dai dati forniti. L'ammortamento degli investimenti e il Markup, sono due fattori che influenzano comunque nella scelta delle soluzioni anche se in maniera non decisiva.

Conclusioni

In questa attività di tesi è stato studiato un problema di apertura di impianti e trasporto merci che è stato opportunamente modellato secondo l'approccio di Programmazione Lineare della Ricerca Operativa. Per risolvere il problema di ricerca delle migliori n soluzioni definite dall'utente usando i software di ottimizzazione, sono stati sviluppati quattro approcci risolutivi, ognuno con dei propri vantaggi e svantaggi dal punto di vista della ricerca formale delle soluzioni o dal punto di vista del carico computazionale richiesto ai software. Il migliore approccio per un'enumerazione globale è il terzo approccio, mentre il quarto approccio è stato sviluppato per soddisfare ulteriori requisiti sulle soluzioni individuate. Sono stati presentati due software quali LINGO e CPLEX, ed ognuno usa un differente metodo risolutivo per l'ottimizzazione globale dei problemi di PLMI quali Branch and Bound per il primo e Branch and Cut per il secondo. Nell'ultimo capitolo sono stati mostrati i risultati delle prove computazionali effettuate sui due software per verificare i tempi di esecuzione e la memoria richiesti per ricercare le differenti soluzioni per quattro scenari differenti, ognuno con caratteristiche diverse per apertura degli impianti, flussi logistici ammessi e numero di tipologie di componenti trattate.

A fronte delle prove il terzo approccio risolutivo risulta migliore, come è stato spiegato nel Capitolo 3, in quanto viene mantenuta costante la memoria necessaria per la formulazione dei modelli per entrambi i software studiati, dove CPLEX in tutti e quattro i casi usa sempre meno memoria rispetto a LINGO. Per quanto riguarda i tempi di elaborazione questi dipendono fortemente dal software utilizzato, e in questo caso LINGO risulta generalmente più lento rispetto a CPLEX in tutti i quattro scenari, dove nei primi due casi si mantiene bassa la variabilità dei tempi di risoluzione rispetto al tempo medio, mentre negli ultimi due casi il software risulta fortemente variabile. Un altro aspetto che si deve tener conto è la capacità dei software di trovare le soluzioni ottime corrette a fronte della stessa formulazione del modello in un file in formato LP. Per i casi studiati LINGO non

trova poche soluzioni accurate nel secondo e terzo scenario, mentre nel quarto scenario per molte iterazioni della ricerca non fornisce le stesse soluzioni ottime di CPLEX, dato che utilizza i valori predefiniti dei parametri di ricerca del solver. CPLEX al contrario, mostra sempre le migliori soluzioni per ciascuna iterazione, dato che internamente utilizza un algoritmo capace di tarare opportunamente il solver in base al modello che viene dato in input.

Conclusa l'attività di tesi, rimangono aperti ancora alcuni aspetti importanti per il supporto dell'attività di pianificazione delle Supply Chain. Il primo aspetto è il miglioramento dell'interfaccia grafica utente, implementandone una nuova che risulta più comoda per l'utente anziché utilizzare un foglio di calcolo. Nell'attività di tesi i risultati dei software vengono salvati in file di testo e dato il grande numero di variabili decisionali usate, risulta poco facile da leggere. Pertanto il secondo aspetto da affrontare è la memorizzazione degli scenari di pianificazione affrontati e dei relativi risultati in una opportuna base dati, in maniera tale che l'utente possa confrontare i problemi futuri con quelli già affrontati in passato per individuare le analogie nei casi e nei risultati in modo da acquisire esperienza. Per creare questa sistema si sta valutando l'utilizzo dei software commerciali forniti da IBM quali il Decision Optimization Center versione sviluppatore e pianificatore, in modo da poter implementare e mettere in esercizio rapidamente un'interfaccia grafica, un risolutore e una base dati che i manager possano iniziare ad usare.

I problemi descritti sono relativi alle attività nel breve periodo, ma esistono dei problemi che devono essere affrontati nel medio e lungo periodo: *analisi di sensitività* e la *Programmazione Stocastica* (PS). L'analisi di sensitività nella Ricerca Operativa analizza le conseguenze sulla variazione dei parametri, presenti nei vincoli, sull'ottimalità e ammissibilità della soluzione trovata, espressa come insieme di variabili decisionali. Si vuole condurre questo tipo di analisi perché la domanda dei pezzi richiesti dai vari stabilimenti dipende fortemente dalla domanda del mercato dei veicoli, e questa risulta fortemente variabile e imprevedibile. Pertanto i manager ne fanno una sovrastima e usando l'analisi di sensitività sulle soluzioni trovate vedono quanto queste risultano robuste al cambiamento della domanda. Tali tecniche di analisi di sensitività vengono normalmente condotte sui problemi di Programmazione Lineare Continua, e non è possibile applicarle a quelli Lineari Miste Intere, soprattutto con la presenza di variabili decisionali binarie. Data l'impossibilità di condurre questo tipo di analisi, si passa a quella "*If-Then*", cioè un'analisi in cui a partire dalle domande dei pezzi stimate, si effettuano delle piccole variazioni per

ricavare dei modelli differenti su cui effettuare delle risoluzioni, in modo da controllare quanto le soluzioni trovate col problema base, risultino adeguate rispetto alle variazioni delle domande. Questo approccio è ovviamente possibile solo se l'utente fornisce al sistema non un numero esiguo di scaglioni produttivi attivabili per ciascun impianto, ma un numero molto elevato ottenendo una matrice degli scaglioni attivabili per impianto in input molto sparsa, aumentando di conseguenza il numero di vincoli, la complessità del modello da risolvere e infine il tempo e la memoria richiesti per la risoluzione. Il motivo per cui si deve fornire così tanti scaglioni è legato al fatto che i vincoli delle domande sulle capacità produttive sono di uguaglianza stretta e se si cambia la domanda deve essere possibile avere a disposizione degli scaglioni produttivi capaci di soddisfarla. Condurre questa analisi richiede inoltre molta memoria per la memorizzazione degli scenari alternativi e delle loro relative soluzioni.

La Programmazione Stocastica, invece, è un metodologia della Ricerca Operativa nella modellazione dei problemi che tengono conto dell'incertezza di alcuni parametri. L'approccio più usato è quello a *due fasi* dove il modello viene formulato come segue

$$\begin{aligned} & \text{minimizza } c^T x + \mathbb{E}[Q(x, \xi)] \\ & \text{soggetta a } a_i^T x \leq b_i \quad i = 1, \dots, m. \\ & x \geq 0 \end{aligned} \tag{4.1}$$

dove $Q(x, \xi)$ è il valore ottimo della seconda fase del problema cioè

$$\begin{aligned} & \text{minimizza } q(\xi)^T y \\ & \text{soggetta a } T(\xi)x + W(\xi)y = h(\xi) \\ & y \geq 0 \end{aligned} \tag{4.2}$$

dove $\xi = (q, h, T, W)$ contiene i dati della realizzazione della seconda fase. Con la PS bisogna prendere una decisione per le variabili decisionali $x \in \mathbf{R}^n$ nella prima fase, per le variabili decisionali $y \in \mathbf{R}^m$ nella seconda fase, prima che diventino noti i dati ξ nella seconda fase. ξ , quindi, è una variabile aleatoria che ha una certa distribuzione di probabilità opportunamente modellata. Per poter calcolare la soluzione ottima per un problema stocastico, esistono due approcci: *problema stocastico deterministico equivalente* e *approssimazione media del campione* (Sample Average Approximation - SAA). Il primo

approccio consiste nel costruire un modello lineare, molto grande, in questo modo

$$\begin{aligned}
 & \text{minimizza } c^T x + \sum_{k=1}^K p_k q_k^T y_k \\
 & \text{soggetta a } T_k x + W_k y_k = h_k \quad k = 1, \dots, K. \\
 & a_i^T x \leq b_i \quad i = 1, \dots, m. \\
 & x \geq 0 \quad y_k \geq 0 \quad k = 1, \dots, K.
 \end{aligned} \tag{4.3}$$

dove p_k è la probabilità associata alla realizzazione del k -esimo scenario nella seconda fase, e q_k , h_k , T_k , W_k sono tutti i dati che si hanno in ciascuno degli scenari. Con questa metodologia si individua la soluzione per le variabili della prima fase che risulta mediamente buona rispetto alle variabili decisionali nella seconda fase. Usare questa tecnica comporta la creazione di un modello lineare molto grande a seconda del numero di scenari realizzati nella seconda fase, e i tempi di calcolo stimati sono nell'ordine delle settimane, tempi non ammissibili per la pianificazione. L'approccio SAA, invece, prevede l'uso di N campioni di ξ , cioè $\xi_1, \xi_2, \dots, \xi_N$, per determinare $Q(x, \xi_k)$ per ciascun campione. Il problema 4.1 viene riformulato come

$$\begin{aligned}
 & \text{minimizza } c^T x + \frac{1}{N} \sum_{k=1}^N Q(x, \xi_k) \\
 & \text{soggetta a } a_i^T x \leq b_i \quad i = 1, \dots, m. \\
 & x \geq 0 \quad y_k \geq 0.
 \end{aligned} \tag{4.4}$$

Il modello stocastico riformulato come modello lineare in 4.4, risulta meno oneroso a livello di calcolo rispetto al modello 4.3. Per avere dei risultati soddisfacenti per la PS bisogna definire un numero di scenari N molto grande, e comunque comporta un certo onere computazionale nella generazione degli scenari della seconda fase, richiedendo l'uso di calcolatori ad alte prestazioni per ridurre i tempi di elaborazione. Queste tecniche stocastiche prevedono sempre l'uso di stimatori medi all'interno della funzione obiettivo, che per alcuni problemi reali non danno soluzioni sempre soddisfacenti. Pertanto, sarà necessario usare un differente stimatore all'interno della funzione obiettivo.

È preferibile usare la PS in modo da affrontare l'incertezza di diversi parametri all'interno del problema tra cui: la domanda per massimizzare i profitti; i dazi per ridurre quanto

possibile i costi di trasporto da un paese all'altro, soprattutto a fronte di determinate azioni politiche-commerciali non prevedibili; il tasso di cambio della moneta e il costo dei materiali anch'essi fortemente influenzati dalle dinamiche del mercato e della politica.

Bibliografia

- [1] Operational Research Society. *OPERATIONAL RESEARCH QUARTERLY*, volume 13(3), 1962. p. 282.
- [2] *Relazione e Bilancio al 31 Dicembre 2016*.
https://www.crf.it/IT/Sfogliabile/Bilancio_CRF_2016/index.html.
ultimo accesso 24 Febbraio 2018.
- [3] Jeroslow R. Balas E. Canonical cuts on the unit hypercube. *SIAM Journal of Applied Mathematics*, 23(1):61–69, 1972.