



Università degli Studi di Padova

DEPARTMENT OF SOMETHING

Master Thesis in MASTER IN SOMETHING

**Implementazione di modelli di
programmazione matematica per
problemi di bin packing**

Supervisor

BIGNAME SCIENTIST
UNIVERSITÀ DI PADOVA

Co-supervisor

DELIGHTFUL RESEARCHER
UNIVERSITÀ BLABLA

Master Candidate

DANIEL ROSSI

THIS IS THE DEDICATION.

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetur. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec porttitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.

Quisque facilisis erat a dui. Nam malesuada ornare dolor. Cras gravida, diam sit amet rhoncus ornare, erat elit consectetur erat, id egestas pede nibh eget odio. Proin tincidunt, velit vel porta elementum, magna diam molestie sapien, non aliquet massa pede eu diam. Aliquam iaculis. Fusce et ipsum et nulla tristique facilisis. Donec eget sem sit amet ligula viverra gravida. Etiam vehicula urna vel turpis. Suspendisse sagittis ante a urna. Morbi a est quis orci consequat rutrum. Nullam egestas feugiat felis. Integer adipiscing semper ligula. Nunc molestie, nisl sit amet cursus convallis, sapien lectus pretium metus, vitae pretium enim wisi id lectus. Donec vestibulum. Etiam vel nibh. Nulla facilisi. Mauris pharetra. Donec augue. Fusce ultrices, neque id dignissim ultrices, tellus mauris dictum elit, vel lacinia enim metus eu nunc.

Sommario

LOREM IPSUM DOLOR SIT AMET, consectetur adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetur. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec porttitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.

Indice

ABSTRACT	v
LIST OF FIGURES	x
LIST OF TABLES	xiii
1 INTRODUZIONE	1
1.1 L'azienda	1
1.2 L'idea	1
1.3 Organizzazione del testo	3
1.4 Convenzioni tipografiche	3
1.5 Convenzioni modelli	4
2 PROCESSI E METODOLOGIE	5
2.1 Contesto	5
2.2 Introduzione al progetto	6
2.3 Vincoli temporali, tecnologici e metodologici	6
2.4 Requisiti e obiettivi	7
2.5 Pianificazione	8
2.5.1 Formazione	8
2.5.2 Preparazione ambiente	9
2.5.3 Scelta framework	9
2.5.4 Realizzazione modelli	9
2.5.5 Validazione	9
2.5.6 Esecuzione test	9
2.5.7 Documentazione dei risultati	9
2.6 Organizzazione oraria	9
2.7 Ambiente di lavoro	10
2.7.1 Metodi di sviluppo	10
2.7.2 Gestione di progetto	11
2.7.3 Linguaggio di programmazione e ambiente di sviluppo	11
2.8 Analisi dei rischi	12

3	INQUADRAMENTO DELLE ATTIVITÀ DI STAGE	15
3.1	Il progetto aziendale	15
3.2	Il progetto di stage	16
3.3	Astrazione del problema	16
3.4	Variante del Bin Packing	17
4	MODELLI PER IL BIN PACKING	19
4.1	Modelli di programmazione matematica	19
4.2	Modelli di programmazione lineari	20
4.3	Modello 2D	21
4.4	Modello 2D con rotazione	23
4.5	Modello 2D con rotazione e sequenza scarico	25
4.6	Modello 3D con rotazione e sovrapposizione	29
5	TEST E VALIDAZIONE	35
5.1	Istanze	35
5.2	Numero di oggetti	36
5.3	Validazione soluzioni fornite	37
6	TECNOLOGIE E STRUMENTI	39
6.1	Tecnologie	39
6.1.1	Python	39
6.1.2	Google Or-Tools	40
6.1.3	Cbc	41
6.1.4	Boost	41
6.1.5	Pandas	42
6.1.6	Matplotlib	42
6.1.7	Multiprocessing	43
6.2	Strumenti	43
6.2.1	Dropbox	43
6.2.2	GitHub	44
6.2.3	Visual Studio Code	44
6.2.4	Jupyter Notebook	45
6.2.5	PIP	45
7	RISULTATI	47
7.1	Risultati modello 2DR	48
7.2	Risultati modello 2DRS	49
7.3	Risultati modello 3D	50

8	CONCLUSION	51
8.1	Consuntivo finale	51
8.2	Raggiungimento degli obiettivi	53
8.3	Conoscenze acquisite	53
	REFERENCES	54
	ACKNOWLEDGMENTS	55

Listing of figures

3.1	Bin packing figures	17
4.1	Veduta area - piano cartesiano	21
4.2	pacchi ruotati	23
4.3	Vie scaricamento pacco	25
4.4	corretta sequenza di scarico	28
4.5	ciucia	33
5.1	Bin packing figures	37
6.1	Bin packing figures	39
6.2	Bin packing figures	40
6.3	Bin packing figures	41
6.4	Bin packing figures	41
6.5	Bin packing figures	42
6.6	Bin packing figures	42
6.7	Bin packing figures	43
6.8	Bin packing figures	43
6.9	Bin packing figures	44
6.10	Bin packing figures	44
6.11	Bin packing figures	45
6.12	Bin packing figures	45

Listing of tables

2.1	A nice table	10
5.1	Le dimensioni dei gruppi di istanze eseguite	36
7.1	Risultati 2DR ottimi	48
7.2	Risultati 2DR best bound	48
7.3	Risultati 2DRS ottimi	49
7.4	Risultati 2DRS best bound	49
7.5	Risultati 3D ottimi	50
7.6	Risultati 3D best bound	50
8.1	A nice table	52

1

Introduzione

L'AZIENDA

Trans-Cel è un'azienda di trasporti che opera nel settore da oltre trent'anni, ha una numerosa flotta composta da bilici e motrici con cui trasporta merci nel nord e centro Italia, tra le qualità che contraddistinguono questa azienda c'è la tecnica del groupage ed il trasporto di merci pericolose.

L'azienda ha sede ad Albignasego in provincia di Padova, qui nell'ufficio operativo vengono organizzati in tempo reale i viaggi dei mezzi per trasportare le merci dei clienti, fornendo magari anche un servizio di deposito.

Da questa realtà si evince come si debba essere sempre pronti a rispondere tempestivamente ad ogni cliente, fornendo loro una soluzione di trasporto adeguata che soddisfi i clienti ma che permetta di far fatturare l'azienda, è da qui che nasce la necessità di un software decisionale di supporto.

L'IDEA

Ad oggi l'azienda ha sviluppato un'euristica che permette di organizzare al meglio le merci all'interno del container del camion. L'idea è quella di valutare la bontà delle soluzioni fornite dall'euristica, confrontando suddette soluzioni con quelle fornite dal modello e individuare se e con quali tipi di oggetti queste riportino

differenze maggiori.

ORGANIZZAZIONE DEL TESTO

Di seguito viene riportata per ogni capitolo una piccola descrizione delle tematiche trattate:

- **Capitolo 1:** in questo capitolo vengono riportati gli obiettivi generali e la pianificazione concordata con l'azienda, inoltre vengono riportate anche le metodologie e strumenti utilizzati, infine una analisi dei rischi.
- **Capitolo 2:** viene descritto il problema generale e come è stato risolto dall'azienda, viene illustrato lo scopo dello stage, viene data una definizione astratta del problema dell'azienda.
- **Capitolo 3:** vengono riportati i modelli utilizzati e una descrizione dell'idea di fondo, per ciascuno di essi verrà riportato integralmente la lista delle variabili utilizzate, descrivendo inoltre anche cosa modellino, riportando per ciascun modello anche le criticità riscontrate e come sono state risolte anche attraverso l'ausilio di materiale grafico.
- **Capitolo 4:** vengono riportate le modalità con cui si sono eseguiti i test illustrando come siano stati strutturati e come sia stato possibile verificare le soluzioni fornite dai modelli.
- **Capitolo 5:** verrà data una descrizione delle funzionalità sviluppate per la verifica delle soluzioni fornite dal modello e successivamente implementate nell'euristica.
- **Capitolo 6:** vengono riportati gli strumenti adottati per lo svolgimento delle attività, corredati da una breve descrizione che riporti come sono stati utilizzati.
- **Capitolo 7:** verranno riportati i risultati ottenuti descrivendo dettagliatamente le informazioni ricavate e fornendo una serie di osservazioni sulle peculiarità dei gruppi di istanze usate nei test, infine si forniranno le conclusioni dello stage.

CONVENZIONI TIPOGRAFICHE

Il testo adotta le seguenti convenzioni tipografiche:

- ogni acronimo, abbreviazione, parola ambigua o tecnica viene spiegata e chiarificata alla fine del testo presso il glossario.
- ogni parola di glossario alla prima apperizione verrà etichetta come segue:
parola^[g]
- nel riportare i modelli verranno adottate alcune convenzione riportate tra le quali .

CONVENZIONI MODELLI

Ogni qual volta si dovrà fare riferimento ad un modello vi si farà riferimento attraverso le seguenti sigle:

- 2D: modello in 2 dimensioni, considera solo profondità e larghezza;
- 2DR: modello in 2 dimensioni con la rotazione;
- 2DRS: modello in 2 dimensioni con rotazione e sequenza di scarico;
- 3D: modello in 3 dimensioni con rotazione e sovrapposizione, la rotazione è rispetto la base e si considerano larghezza, profondità e altezza.

2

Processi e metodologie

In questo capitolo verranno riportati in modo approfondito lo scopo e gli obiettivi dello stage, riportando lo scadenziario delle attività e contestualizzandole alla realtà dell'azienda.

CONTESTO

Il progetto generale nasce dalla visione di Filippo Sottovia, titolare dell'azienda, di realizzare un software decisionale di supporto che permettesse di sopperire a molteplici obiettivi quali:

- agevolazione degli operatori nello svolgimento delle loro mansioni;
- facilitazione del processo decisionale richiedendo così un lavoratore meno esperto;
- nuove attività frutto del tempo risparmiato grazie all'aumento della produttività;
- condivisione in tempo reale delle informazioni sullo stato dei trasporti;
- stima di costi e profitti disponibile in ogni momento.

Il software, fornisce un'interfaccia grafica web intuitiva che fa dell'usabilità la propria punta di diamante, essa infatti persegue l'idea di rendere il più semplice possibile operare sul sistema senza però peccare di professionalità, il sistema

inoltre è equipaggiato con algoritmi che permettono di ottimizzare gli ordini organizzando al meglio le merci da trasportare ripartendole ai camion della flotta, tenendo conto degli orari di carico e scarico delle sedi dei clienti/fornitori, delle condizioni di traffico sulle strade, delle pause dovute per legge agli autisti e della particolarità delle merci.

INTRODUZIONE AL PROGETTO

L'azienda per permettere di stimare lo spazio occupato dalle merci ha sviluppato un'euristica, questa riceve in input le merci da trasportare divise nei diversi ordini, le approssima a parallelepipedi e le dispone al meglio sul pianale del camion, sovrapponendole dove possibile, con l'obiettivo di ridurre al minimo lo spazio lineare occupato dalle stesse, così facendo si risparmia spazio per eventuali altre merci da caricare qualora arrivassero nuovi ordini. A rendere ancora più complicato il lavoro dell'euristica riportiamo tre principali problemi da tenere in considerazione:

- Stabilità degli oggetti: la faccia inferiore di ciascun oggetto deve poggiare per intero sulle facce superiori di altri oggetti sotto di sé o sul pianale del camion;
- Sequenza di scarico: ogni oggetto deve poter essere scaricato lateralmente o frontalmente, questo implica che non può avere in torno a sé merci che ne blocchino lo scarico, in quanto appartenenti ad ordini maggiori;
- Baricentro degli oggetti: peso delle merci deve essere equamente distribuito sul pianale del camion.

VINCOLI TEMPORALI, TECNOLOGICI E METODOLOGICI

Nel periodo di stage svolto presso l'azienda mi è stato chiesto di tenere un *diario* condiviso utilizzando *dropbox*, nel suddetto diario mi si richiedeva di annotare giornalmente l'avanzare del lavoro riportando l'avanzamento dei lavori, idee, osservazioni riguardanti criticità rilevate e positività riscontrate negli strumenti utilizzati, nella cartella condivisa mi è stato chiesto di includere anche gli articoli accademici letti e le presentazioni fatte in azienda man mano che si proseguiva

con lo stage. L'ambiente di lavoro si è dimostrato molto stimolante ed è stato molto utile confrontarsi con i colleghi per ricevere opinioni e consigli su come indirizzare il lavoro. Ogni mattina prima di iniziare a lavorare facevamo un rapido *brainstorming* individuando gli obiettivi della giornata e per fare il punto. Un'altra cosa importante è stata la presentazione di metà stage, alla quale ha presenziato tutto il team di sviluppo, parte del personale d'amministrazione e il Professor De Giovanni che mi ha fornito preziosi consigli su come orientare gli sforzi al meglio per la seconda metà dello stage. Prima di iniziare lo stage è stato concordato con l'azienda un piano di lavoro su un totale di 320 ore, lavorando 5 giorni a settimana, 8 ore per ciascun giorno.

REQUISITI E OBIETTIVI

Nella tabella riportata di seguito vengono riportati gli obiettivi dello stage, a corredo degli stessi vi sarà un codice univoco ed una breve descrizione.

Ogni obiettivo è provvisto di un codice che lo identifica formato da una delle seguenti stringhe **ob,de,op**] rappresentante il livello di importanza e da un numero incrementale positivo, che rispetta la seguente nomenclatura:

[importanza][identificativo].

Il livello di importanza di ciascun obiettivo può essere uno tra i seguenti:

- **Obbligatorio:** individuato dalla stringa *ob*, sono obiettivi fondamentali per la riuscita del progetto, il loro soddisfacimento dovrà verificarsi assolutamente entro la fine dello stage, pena il fallimento dello stesso;
- **Desiderabile:** individuato dalla stringa *de*, sono obiettivi secondari su cui però si nutre dell'interesse, il loro soddisfacimento è auspicabile entro la fine dello stage;
- **Opzionale:** individuato dalla stringa *op*, sono obiettivi di contorno su cui si nutre poco interesse, la loro realizzazione si verificherà nel momento in cui si dovesse soddisfare tutti gli obiettivi obbligatori e desiderabili prima della fine dello stage.

Si prevede lo svolgimento dei seguenti obiettivi:

- Obbligatori

- ob01: individuazione e analisi *constraints* modello 2D;
 - ob02: modello matematico 2D con *framework* di modellazione algebrica;
 - ob03: traduzione modello in Python con l’ausilio di Google OR-Tools;
 - ob04: test sul modello 2D e confronto con l’euristica;
 - ob05: individuazione e analisi *constraints* modello 2DR;
 - ob06: modello matematico 2DR con *framework* di modellazione algebrica;
 - ob07: traduzione modello in Python con l’ausilio di Google OR-Tools;
 - ob08: test sul modello 2DR e confronto con l’euristica;
- Desiderabili
 - de01: individuazione e analisi *constraints* modello 3D;
 - de02: modello matematico 3D con *framework* di modellazione algebrica;
 - de03: traduzione modello in Python con l’ausilio di Google OR-Tools;
 - de04: test sul modello 3D e confronto con l’euristica;
 - Opzionali
 - op01: evoluzione euristica, fornita dall’azienda, con nuove funzionalità;

PIANIFICAZIONE

Con le ore a disposizione per questo stage si è proceduto a organizzare come segue le attività.

FORMAZIONE

Si è visto necessario approfondire la programmazione lineare e la letteratura correlata al problema del bin packing, imparare il linguaggio di programmazione Python e l’ambiente fornito dallo strumento *Jupyter*.

PREPARAZIONE AMBIENTE

La preparazione dell'ambiente di lavoro ha richiesto l'installazione di numerose librerie e framework sia pubblici che aziendali per riuscire a integrare i modelli con le librerie aziendali.

SCELTA FRAMEWORK

La scelta finale del framework di modellazione algebrica è ricaduta sullo stesso Or-Tools in quanto forniva una interfaccia Python ed era quello che forniva una documentazione migliore oltre che una community di sviluppatori molto attiva.

REALIZZAZIONE MODELLI

Periodo di maggior peso rispetto ad altri, esso ha portato allo sviluppo dei modelli e delle loro varianti, includendo in sé varie attività.

VALIDAZIONE

Periodo richiesto per la validazione delle soluzioni fornite dai diversi modelli per verificare la loro correttezza.

ESECUZIONE TEST

Periodo in cui sono stati eseguiti test massivi sia dei modelli che dell'euristica con cui successivamente estrapolare informazioni utili al confronto.

DOCUMENTAZIONE DEI RISULTATI

Analisi dei risultati ottenuti e conseguente documentazione al fine di capire dove e come si potesse migliorare l'euristica.

ORGANIZZAZIONE ORARIA

La pianificazione, in termini di quantità di ore di lavoro, è stata così distribuita:

Durata in ore		Descrizione dell'attività
56		A: Formazione
	8	• Ricerca <i>framework</i> di modellazione algebrica
	24	• Studio di tale <i>framework</i>
	24	• Studio Google OR - Tools
104		B: Versione algoritmo 2D
	8	• Individuazione ed analisi <i>constraints</i>
	48	• Prototipazione modello
	24	• Traduzione in Python - Google OR - Tools
	24	• Test e confronto con euristica
72		C: Versione algoritmo 2D con rotazione
	8	• Individuazione ed analisi <i>constraints</i>
	24	• Prototipazione modello
	16	• Traduzione in Python - Google OR - Tools
	16	• Test e confronto con euristica
72		D: Versione algoritmo 3D con sovrapposizione
	8	• Individuazione ed analisi <i>constraints</i>
	32	• Prototipazione modello
	16	• Traduzione in Python - Google OR - Tools
	24	• Test e confronto con euristica
16		E: Lavoro sviluppo euristica
	16	• Realizzazione nuove funzioni euristica
Totale: 320		

Table 2.1: A nice table

AMBIENTE DI LAVORO

METODI DI SVILUPPO

Il ciclo di vita di un prodotto in Trans-Cel segue il metodo incrementale, in particolare il tutto inizia da una fase di concezione dell'idea, analisi della stessa, progettazione ed infine partendo dagli obiettivi più importanti si realizza il prodotto rilasciando periodicamente una versione dello stesso che mostri il procedere dei lavori e delle nuove funzionalità inserite. In quest'ottica lo sviluppo di un modello

può essere associato ad un incremento, la cui *milestone* è la ricezione dei risultati, a sua volta lo sviluppo di ciascun modello segue il metodo incrementale, lo sviluppo di un modello può essere visto come l'insieme di tre principali attività:

- Analisi letteratura: lettura articoli accademici riportanti modelli simili o idee per lo sviluppo degli stessi.
- Scrittura: scrittura del modello e integrazione con il sistema grazie al framework Or-Tools.
- Verifica: testing massimo e verifica delle soluzioni fornite.

Inoltre è possibile cogliere analogie con il metodo *agile*^[g] se si pensa ai brainstorming, momenti importanti in quanto ci si confronta in modo critico e si fa il punto della situazione oltre che esporre le problematiche che si stanno incontrando.

GESTIONE DI PROGETTO

Per quanto riguarda la gestione di progetto sono stati utilizzati alcuni strumenti descritti con maggiore dettaglio nel capitolo, in generale per la gestione dei task da eseguire si è fatto uso di *Taiga*^[g], uno strumento di gestione delle task molto simile per funzionamento a *Trello*^[g], per la gestione della comunicazione e condivisione informazioni si è fatto uso dell'applicazione *Telegram*^[g], per la condivisione di documentazione e articoli si è fatto uso del servizio Dropbox, per il versionamento si è fatto uso del servizio *GitHub*^[g] per la familiarità dello strumento, per quanto riguarda l'interfaccia con cui versionare ho utilizzato *git*^[g] da terminale.

LINGUAGGIO DI PROGRAMMAZIONE E AMBIENTE DI SVILUPPO

Per la totalità dello stage si è lavorato utilizzando Jupyter Notebook, con questo strumento è stato possibile scrivere programmi in linguaggio Python in modo molto agevole. Questo linguaggio di programmazione è orientato agli oggetti ed interpretato dinamicamente al momento dell'esecuzione da un interprete. Python risulta veramente versatile in quanto fornisce incredibili funzionalità utilizzabili in modo semplice e intuitivo, dispone di moltissimi moduli che permettono di fare le più svariate cose, il linguaggio inoltre permette di esporre costrutti *C++*^[g] in Python permettendo di utilizzarli nei propri programmi, questo permette di

mantenere efficienza e modularità. L'unico punto a suo sfavore è la non possibilità di realizzare programmi *multithreading*^[g] se non utilizzando il modulo *multiprocessing*^[g] che comporta però dei compromessi.

ANALISI DEI RISCHI

In questa sezione vengono riportati i principali rischi che si prefiggevano iniziando lo stage, ciascuno di essi oltre ad avere una breve descrizione riportano un livello di rischio e come si possa fare per evitarli:

- **Difficoltà nelle tecnologie adottate**

Fin da quando si è concordato lo stage si era capito che Python avrebbe avuto un ruolo dominante ma la mole di librerie e estensioni rendeva il linguaggio troppo vasto da poter approfondire nella sua interezza, ed oltre a questo vi era il modulo Or-Tools e *Pandas*^[g] da approfondire e utilizzare.

- **Livello di rischio:** Basso;
- **Contromisure:** Studiare in modo il più possibile approfondito Python e i moduli sopra citati.

- **Difficoltà di integrazione nel team**

Di fondamentale importanza per la riuscita di un progetto è la cooperazione con i colleghi e la creazione di un ambiente di lavoro sano e che stimoli la creatività, essendo un nuovo arrivato inserito in un ambiente a forte stress per le stringenti scadenze vi era la possibilità di entrare in conflitto con taluni colleghi.

- **Livello di rischio:** Basso;
- **Contromisure:** Perseguire un atteggiamento positivo, critico e oggettivo.

- **Contrattempi dovuti a malattie e impegni**

Un rischio da tenere in considerazione è quello dovuto a impegni o malattie che precludano la possibilità di recarsi nel luogo di lavoro, data la durata dello stage è sicuramente possibile possa verificarsi.

- **Livello di rischio:** Basso;

- **Contromisure:** Organizzare precedentemente ogni impegno non lavorativo e tempo di *slack*^[g] per evitare contrattempi.

- **Difficoltà di stima dei tempi previsti**

Con un progetto di così lunga durata e l'inesperienza che ci portiamo appresso è possibile che vengano fatti degli errori di valutazione in termini di tempistiche per lo svolgimento delle diverse attività pianificate.

- **Livello di rischio:** Medio;
- **Contromisure:** Rendere partecipi nella definizione del piano di lavoro persone esperte, come il tutor aziendale.

This is some random quote to start off the chapter.

Firstname lastname

3

Inquadramento delle attività di stage

IL PROGETTO AZIENDALE

Per riuscire ad inquadrare al meglio lo scopo del progetto di stage è importante capire a fondo il progetto generale che l'azienda porta avanti da ormai alcuni anni, poiché essi sono tra loro intrinsecamente legati. Il progetto generale mira a fornire un prodotto che possa far progredire il mercato del trasporto attraverso la digitalizzazione dello stesso, questo attraverso il superamento dei vecchi gestionali, utilizzati concretamente solo per la fatturazione, e l'introduzione di una piattaforma web attraverso cui gestire ogni passaggio nel rapporto cliente-operatore. Il software permette di gestire la flotta di mezzi e organizzare gli ordini di carico e scarico commissionati dai clienti, questa organizzazione può essere fatta manualmente da un operatore esperto oppure affidandosi ad algoritmi che organizzino gli ordini utilizzando i camion disponibili nel miglior modo possibile. Il sistema dispone di un algoritmo per il *vehicle routing*^[g], su cui si può dire faccia capo l'intero progetto, questo si occupa di ottimizzare i viaggi dei camionisti tenendo conto di un numero incredibile di variabili, questo algoritmo ha subito diverse upgrade in un'ottica di miglioramento continuo, per renderlo più efficiente, più versatile e più preciso. In questo contesto l'euristica che fornisce la valutazione approssimativa dello spazio occupato da un certo numero di merci è molto importante, questo perché è necessario sapere quanti mezzi siano necessari per trasportare le merci

richieste e quanto spazio sia ancora disponibile per ordini futuri.

IL PROGETTO DI STAGE

Il progetto di stage dopo l'evento *StageIt*^[g] ed alcune riunioni presso l'azienda è stato studiato dettagliatamente e riportato nel documento piano di lavoro, nello stesso sono riportati gli obiettivi e la pianificazione delle attività. L'importanza di ottenere una valutazione reale dello spazio occupato da un certo numero di oggetti è fondamentale, per fare ciò lo stage prevede la realizzazione di diversi modelli, lo sviluppo di tali modelli sarà incrementale in quanto ciascun modello eredita quanto sviluppato nel precedente. Ogni modello per definizione permetterà di individuare la disposizione ottima dell'*istanza*^[g] corrente, questa potrà poi essere confrontata con quella fornita dall'euristica, da questo confronto potremo individuare informazioni utili che permettano di valutare oggettivamente la bontà delle soluzioni fornite dall'euristica, questo permette di capire se con il passare delle versioni rilasciate l'euristica sta migliorando e di quanto rispetto al passato.

ASTRAZIONE DEL PROBLEMA

Il problema sopra riportato può essere approssimato considerando il container del camion e le merci come se fossero dei parallelepipedi, questa astrazione è necessaria per poter approssimare il tutto come richiesto dal *Bin Packing Problem*, problema su cui si è studiato e discusso molto in ambito accademico portando alla realizzazione di algoritmi esatti ed euristiche.

Si consideri un insieme $I = \{1, \dots, n\}$ di oggetti aventi dimensioni w_i , d_i e h_i con $i \in I$, un insieme di contenitori di uguale dimensione W , D e H , ogni oggetto ha la possibilità di essere ruotato di 90° rispetto la propria base e si dà per ipotesi che $w_i \leq W$, $d_i \leq D$ e $h_i \leq H$, inoltre è possibile che gli oggetti possano essere sovrapposti, l'obiettivo è di utilizzare il minor numero di contenitori che riescano a contenere tutti gli oggetti dell'insieme I .

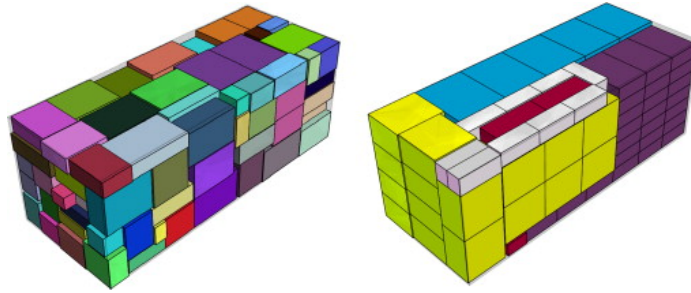


Figure 3.1: Due contenitori completamente pieni

VARIANTE DEL BIN PACKING

Il problema del Bin Packing appartiene ad una famiglia di problemi e ne è il rappresentante più famoso, una variante di questo problema è lo *Strip Packing Problem*, le differenze principali dal precedente possono essere individuate di seguito:

- **Numero di contenitori:** viene utilizzato un singolo contenitore;
- **profondità:** la profondità del contenitore è infinita;
- **Oggetti:** come obiettivo si quello di occupare minor spazio lineare possibile rispetto la profondità del contenitore.

4

Modelli per il Bin Packing

In questo capitolo presenteremo prima una rapida introduzione ai modelli matematici per poi analizzare nel dettaglio ciascun modello utilizzato spiegando vincolo per vincolo il loro funzionamento.

MODELLI DI PROGRAMMAZIONE MATEMATICA

Quando si parla di problemi di ottimizzazione spesso si fa riferimento alla ricerca operativa e ai modelli matematici, questi modelli producono una soluzione che massimizza o minimizza una funzione obiettivo su un certo dominio, questa funzione può rappresentare un aspetto di costo o guadagno, inoltre la soluzione restituita ha la proprietà di restituire una soluzione ottima. La funzione obiettivo solitamente viene rappresentata come segue:

$$\max z = f(x) \quad (\text{oppure } \min z = f(x))$$

s.t.

$$g_i(x) = \begin{cases} \leq b_i \\ = b_i, & i = 1, \dots, m \\ \geq b_i \end{cases}$$

$$x = (x_1, \dots, x_n) \in X \subseteq \mathbb{R}^n$$

In un modello sono presenti:

- **Variabili decisionali:** sono variabili con un dominio prefissato, vengono utilizzate per formulare tutti gli altri elementi del modello, agendo sui valori assunti dalle stesse si troverà la soluzione ottima;
- **Funzione obiettivo:** funzione che deve essere massimizzata o minimizzata in base agli altri elementi su un dominio dato;
- **Vincoli:** serie di vincoli che correlano tra loro le variabili decisionali e permettono di descrivere condizioni fisiche o requisiti particolari richiesti dalla soluzione.

MODELLI DI PROGRAMMAZIONE LINEARI

Particolari tipi di problemi di programmazione matematica sono quelli in cui la funzione obiettivo $f(x)$ e i vincoli $g_i(x)$ sono funzioni lineari, in tal caso si può parlare di modello di *programmazione lineare*, espresso nel seguente modo:

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

s.t.

$$\sum_{j=1}^n a_{ij} x_j = \begin{cases} \leq b_i \\ = b_i, \\ \geq b_i \end{cases} \quad i = 1, \dots, m$$

$$x = (x_1, \dots, x_n) \in X \subseteq \mathbb{R}^n$$

Nonostante i modelli non lineari siano a volte molto più compatti ed intuitivi da capire, i modelli lineari mantengono comunque semplicità e sono facilmente risolvibili, sono molti i solver *commerciali*^[g] e *open source*^[g] disponibili per la loro risoluzione, quello utilizzato durante lo stage è il solver open source di programmazione lineare intera CBC scritto in C++ con interfacciato con Python grazie a Or-Tools. La creazione di un modello lineare parte dall'osservazione di un problema reale, fissando un obiettivo e creando vincoli che descrivano i requisiti

estratti dal problema reale astraendo il tutto ad un insieme di vincoli lineari. Ogni variabile può avere un dominio differente, continua ($x_j \in \mathbb{R}$), intera positiva ($x_j \in \mathbb{Z}^+$) o binaria ($x_j \in \{0, 1\}$) a seconda dell'utilizzo che si deve farne.

MODELLO 2D

Il modello 2D considera solo larghezza e profondità degli oggetti e non ne permette la rotazione, l'idea di base è quella di considerare solo la *visione aerea* del contenitore che viene così approssimata ad un rettangolo, questo rettangolo ha larghezza (x) prefissita e profondità (y) infinita, le due dimensioni posso essere considerate come assi di un piano cartesiano aventi origine nel punto di intersezione tra i due assi, per convenzione diciamo che l'asse x sarà quello della larghezza e l'asse y quello della profondità.

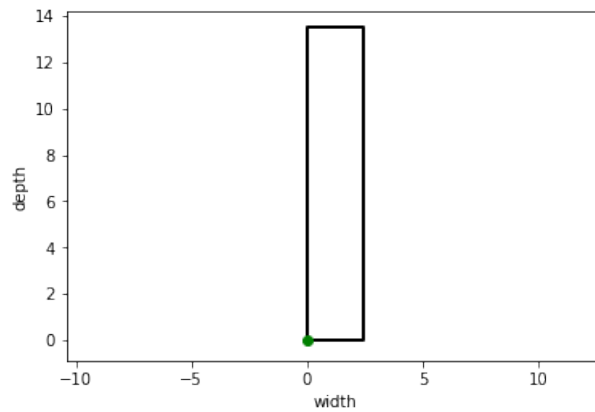


Figure 4.1: Veduta area del container, piano cartesiano con origine nel punto verde.

Il modello consiste delle seguenti variabili continue positive:

- la variabile intera x_i con $i \in I$ individua la coordinata sull'asse x dell'angolo in basso a sinistra dell'oggetto i ;
- la variabile intera y_i con $i \in I$ individua la coordinata sull'asse y dell'angolo in basso a sinistra dell'oggetto i ;

Il modello consiste delle seguenti variabili binarie:

- la variabile binaria l_{ij} con $i, j \in I$ assume il valore 1 se l'oggetto i è situato alla sinistra dell'oggetto j , altrimenti è 0;
- la variabile binaria b_{ij} con $i, j \in I$ assume il valore 1 se l'oggetto i è situato al di sotto dell'oggetto j , altrimenti è 0.

$$\min D \quad (4.1)$$

$$\text{s.t.} \quad l_{ij} + l_{ji} + b_{ij} + b_{ji} \geq 1 \quad i < j \quad i, j \in I \quad (4.2)$$

$$y_i - y_j + M_d b_{ij} \leq M_d - d_i \quad i, j \in I \quad (4.3)$$

$$x_i - x_j + M_w l_{ij} \leq M_w - w_i \quad i, j \in I \quad (4.4)$$

$$x_i + w_i \leq W \quad i \in I \quad (4.5)$$

$$y_i + d_i \leq D \quad i \in I \quad (4.6)$$

$$b_{ij}, l_{ij} \in \{0, 1\} \quad i \neq j \quad i, j \in I \quad (4.7)$$

$$x_i, y_i \in \mathbb{R}^+ \quad i \in I \quad (4.8)$$

Spieghiamo ora il significato di ciascun vincolo:

- il vincolo (4.2) impone che presi due oggetti almeno uno dei due si trovi o a sinistra o a destra dell'altro;
- il vincolo (4.3) dice che se $b_{ij} = 1$ allora impone che l'oggetto i si trovi dietro l'oggetto j , altrimenti il vincolo viene disattivato.
- il vincolo (4.4) dice che se $l_{ij} = 1$ allora impone che l'oggetto i si trovi alla sinistra l'oggetto j , altrimenti il vincolo viene disattivato.
- il vincolo (4.5) impone che dato il valore x_i , corrispondente alla coordinata x dell'angolo sinistro più arretrato dell'oggetto, sommandoci la larghezza w_i dell'oggetto e ottenendo quindi la coordinata x dell'angolo destro più arretrato, questa sia contenuta nell'intervallo $[0, W]$;
- il vincolo (4.6) impone che dato il valore y_i , corrispondente alla coordinata y dell'angolo sinistro più arretrato dell'oggetto, sommandoci la profondità d_i dell'oggetto e ottenendo quindi la coordinata y dell'angolo sinistro più avanzato, questa sia contenuta nell'intervallo $[0, D]$;

MODELLO 2D CON ROTAZIONE

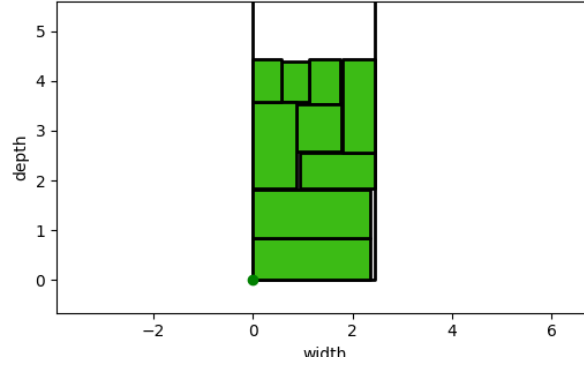


Figure 4.2: Veduta aerea di 9 pacchi, alcuni ruotati.

Il modello consiste delle seguenti variabili continue positive:

- la variabile intera x_i con $i \in I$ individua la coordinata sull'asse x dell'angolo in basso a sinistra dell'oggetto i ;
- la variabile intera y_i con $i \in I$ individua la coordinata sull'asse y dell'angolo in basso a sinistra dell'oggetto i ;
- la variabile intera Ω_i con $i \in I$ assume il valore d_i se l'oggetto è stato ruotato altrimenti w_i ;
- la variabile intera Δ_i con $i \in I$ assume il valore d_i se l'oggetto è stato ruotato altrimenti w_i ;

Il modello consiste delle seguenti variabili binarie:

- la variabile binaria l_{ij} con $i, j \in I$ assume il valore 1 se l'oggetto i è situato alla sinistra dell'oggetto j , altrimenti è 0;
- la variabile binaria b_{ij} con $i, j \in I$ assume il valore 1 se l'oggetto i è situato al di sotto dell'oggetto j , altrimenti è 0.
- la variabile binaria r_i con $i \in I$ assume il valore 1 se l'oggetto i è ruotato su se stesso di 90° , ne risulta che w_i e d_i sono invertiti, altrimenti è 0;

$$\min D \quad (4.9)$$

$$\text{s.t.} \quad l_{ij} + l_{ji} + b_{ij} + b_{ji} \geq 1 \quad i < j \quad i, j \in I \quad (4.10)$$

$$\Delta_i = d_i(1 - r_i) - w_i r_i \quad i, j \in I \quad (4.11)$$

$$\Omega_i = w_i(1 - r_i) - d_i r_i \quad i, j \in I \quad (4.12)$$

$$y_i - y_j + M_d b_{ij} \leq M_d - \Delta_i \quad i, j \in I \quad (4.13)$$

$$x_i - x_j + M_w l_{ij} \leq M_w - \Omega_i \quad i, j \in I \quad (4.14)$$

$$x_i + \Omega_i \leq W \quad i \in I \quad (4.15)$$

$$y_i + \Delta_i \leq D \quad i \in I \quad (4.16)$$

$$b_{ij}, l_{ij} \in \{0, 1\} \quad i \neq j \quad i, j \in I \quad (4.17)$$

$$x_i, y_i \in \mathbb{R}^+ \quad i \in I \quad (4.18)$$

Spieghiamo ora il significato di ciascun vincolo:

- il vincolo (4.10) impone che presi due oggetti almeno uno dei due si trovi o a sinistra o a destra dell'altro;
- il vincolo (4.11) impone che Δ_i corrisponda alla profondità corretta considerando la rotazione;
- il vincolo (4.12) impone che Ω_i corrisponda alla larghezza corretta considerando la rotazione;
- il vincolo (4.13) dice che se $b_{ij} = 1$ allora impone che l'oggetto i si trovi dietro l'oggetto j , altrimenti il vincolo viene disattivato.
- il vincolo (4.14) dice che se $l_{ij} = 1$ allora impone che l'oggetto i si trovi alla sinistra l'oggetto j , altrimenti il vincolo viene disattivato.
- il vincolo (4.15) impone che dato il valore x_i , corrispondente alla coordinata x dell'angolo sinistro più arretrato dell'oggetto, sommandoci la larghezza w_i dell'oggetto e ottenendo quindi la coordinata x dell'angolo destro più arretrato, questa sia contenuta nell'intervallo $[0, W]$;
- il vincolo (4.16) impone che dato il valore y_i , corrispondente alla coordinata y dell'angolo sinistro più arretrato dell'oggetto, sommandoci la profondità d_i dell'oggetto e ottenendo quindi la coordinata y dell'angolo sinistro più avanzato, questa sia contenuta nell'intervallo $[0, D]$;

MODELLO 2D CON ROTAZIONE E SEQUENZA SCARICO

Con questo modello ci proponiamo di disporre nel miglior modo possibile gli oggetti sul pianale del camion tenendo conto anche della sequenza di scarico, un oggetto può essere scaricato lateralmente da destra o da sinistra, oppure dal fondo del camion.

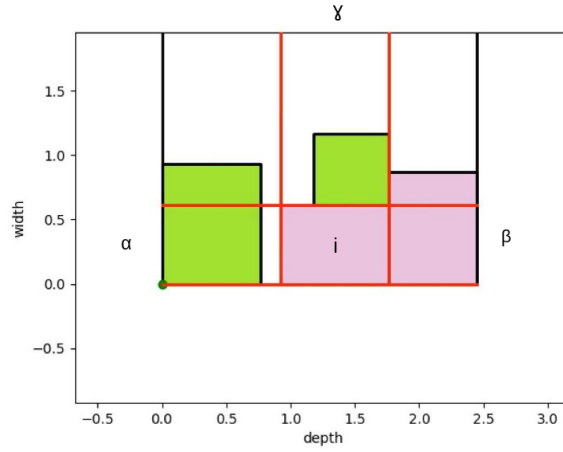


Figure 4.3: Vie con cui scaricare un pacco.

Nel momento in cui l'oggetto verrà scaricato questo dovrà avere almeno una di queste vie libere affinché possa essere scaricato.

Come si può notare nella figura 4.3, le vie attraverso cui può essere effettuato uno scarico vengono deciso attraverso l'indice $v \in \theta = \{\alpha, \beta, \gamma\}$ che individua rispettivamente:

- $v = \alpha$: via alla sinistra del pacco da cui può essere sfilato;
- $v = \beta$: via alla destra del pacco da cui può essere sfilato;
- $v = \gamma$: via di fronte al pacco da cui può essere sfilato;

Il modello introduce le seguenti variabili continue positive:

- la variabile intera x_i con $i \in I$ individua la coordinata sull'asse x dell'angolo in basso a sinistra dell'oggetto i ;

- la variabile intera y_i con $i \in I$ individua la coordinata sull'asse y dell'angolo in basso a sinistra dell'oggetto i;
- la variabile intera Ω_i con $i \in I$ assume il valore d_i se l'oggetto è stato ruotato altrimenti w_i ;
- la variabile intera Δ_i con $i \in I$ assume il valore d_i se l'oggetto è stato ruotato altrimenti w_i ;
- la parametro binario o_{ij} assume il valore 1 se l'oggetto i appartiene ad un ordine che deve essere scaricato prima dell'ordine a cui appartiene l'oggetto j, altrimenti è 0;

Il modello introduce le seguenti variabili binarie:

- la variabile binaria l_{ij} con $i, j \in I$ assume il valore 1 se l'oggetto i è situato alla sinistra dell'oggetto j, altrimenti è 0;
- la variabile binaria b_{ij} con $i, j \in I$ assume il valore 1 se l'oggetto i è situato al di sotto dell'oggetto j, altrimenti è 0.
- la variabile binaria r_i con $i \in I$ assume il valore 1 se l'oggetto i è ruotato su se stesso di 90° , ne risulta che w_i e d_i sono invertiti, altrimenti è 0;
- la variabile binaria α_{ij} con $i, j \in I$ assume il valore 1 se l'oggetto i ha alla sua sinistra l'oggetto j e questo gli impedisca di essere sfilato da sinistra, altrimenti è 0;
- la variabile binaria β_{ij} con $i, j \in I$ assume il valore 1 se l'oggetto i ha alla sua destra l'oggetto j e questo gli impedisca di essere sfilato da destra, altrimenti è 0;
- la variabile binaria γ_{ij} con $i, j \in I$ assume il valore 1 se l'oggetto i ha davanti a se l'oggetto j e questo gli impedisca di essere sfilato centralmente, altrimenti è 0;
- la variabile binaria s_{vi} con $i \in I$ e $v \in \theta$ assume il valore 1 se l'oggetto i ha nella direzione v almeno un oggetto che gli impedisca di essere sfilato nella rispettiva direzione, altrimenti è 0;

$$\min D \quad (4.19)$$

$$\text{s.t.} \quad l_{ij} + l_{ji} + b_{ij} + b_{ji} \geq 1 \quad i < j \quad i, j \in I \quad (4.20)$$

$$y_i - y_j + M_d b_{ij} \leq M_d - \Delta_i \quad i, j \in I \quad (4.21)$$

$$x_i - x_j + M_w l_{ij} \leq M_w - \Omega_i \quad i, j \in I \quad (4.22)$$

$$x_i + \Omega_i \leq W \quad i, j \in I \quad (4.23)$$

$$y_i + \Delta_i \leq D \quad i, j \in I \quad (4.24)$$

$$\Delta_i = d_i(1 - r_i) - w_i r_i \quad i, j \in I \quad (4.25)$$

$$\Omega_i = w_i(1 - r_i) - d_i r_i \quad i, j \in I \quad (4.26)$$

$$\alpha_{ij} \leq l_{ij} \quad i, j \in I \quad (4.27)$$

$$\alpha_{ij} \leq 1 - (b_{ij} + b_{ji}) \quad i, j \in I \quad (4.28)$$

$$\alpha_{ij} \geq l_{ij} - M_\alpha(b_{ij} + b_{ji}) \quad i, j \in I \quad (4.29)$$

$$\beta_{ij} \leq l_{ji} \quad i, j \in I \quad (4.30)$$

$$\beta_{ij} \leq 1 - (b_{ij} + b_{ji}) \quad i, j \in I \quad (4.31)$$

$$\beta_{ij} \geq l_{ji} - M_\beta(b_{ij} + b_{ji}) \quad i, j \in I \quad (4.32)$$

$$\gamma_{ij} \leq b_{ij} \quad i, j \in I \quad (4.33)$$

$$\gamma_{ij} \leq 1 - (l_{ij} + l_{ji}) \quad i, j \in I \quad (4.34)$$

$$\gamma_{ij} \geq b_{ij} - M_\gamma(l_{ij} + l_{ji}) \quad i, j \in I \quad (4.35)$$

$$\alpha_{ij} o_{ij} \leq s_{ai} \quad i, j \in I \quad (4.36)$$

$$\beta_{ij} o_{ij} \leq s_{\beta i} \quad i, j \in I \quad (4.37)$$

$$\gamma_{ij} o_{ij} \leq s_{\gamma i} \quad i, j \in I \quad (4.38)$$

$$\sum_{v \in \theta} s_{vi} \leq 2 \quad i \in I \quad (4.39)$$

$$b_{ij}, l_{ij}, r_i \in \{0, 1\} \quad i \neq j \quad i, j \in I \quad (4.40)$$

$$x_i, y_i \in \mathbb{Z}^+ \quad i \in I \quad (4.41)$$

$$s_{vi} \in \{0, 1\} \quad i \in I \quad v \in \theta \quad (4.42)$$

$$(4.43)$$

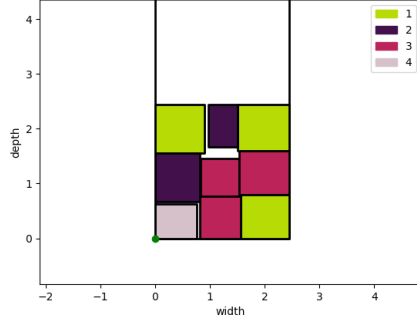


Figure 4.4: 9 pacchi con corretta sequenza di scarico.

Spieghiamo ora il significato di ciascun vincolo:

- il vincolo (4.20) impone che presi due oggetti almeno uno dei due si trovi o a sinistra o a destra dell'altro;
- il vincolo (4.21) dice che se $b_{ij} = 1$ allora impone che l'oggetto i si trovi dietro l'oggetto j , altrimenti il vincolo viene disattivato;
- il vincolo (4.22) dice che se $l_{ij} = 1$ allora impone che l'oggetto i si trovi alla sinistra l'oggetto j , altrimenti il vincolo viene disattivato;
- il vincolo (4.23) impone che dato il valore x_i , corrispondente alla coordinata x dell'angolo sinistro più arretrato dell'oggetto, sommandoci la larghezza w_i dell'oggetto e ottenendo quindi la coordinata x dell'angolo destro più arretrato, questa sia contenuta nell'intervallo $[0, W]$;
- il vincolo (4.24) impone che dato il valore y_i , corrispondente alla coordinata y dell'angolo sinistro più arretrato dell'oggetto, sommandoci la profondità d_i dell'oggetto e ottenendo quindi la coordinata y dell'angolo sinistro più avanzato, questa sia contenuta nell'intervallo $[0, D]$;
- il vincolo (4.25) impone che Δ_i corrisponda alla profondità corretta considerando la rotazione;
- il vincolo (4.26) impone che Ω_i corrisponda alla larghezza corretta considerando la rotazione;
- i vincoli (4.27), (4.28) e (4.29) impongono che se $l_{ij} = 1 \wedge t_{ij} + t_{ji} = 0 \Rightarrow \alpha_{ij} = 1$, che significa che se un'oggetto i si trovi a destra di un oggetto j ma nessuno

dei due è dietro all'altro, allora l'oggetto i non può essere sfilato dalla sponda sinistra del camion;

- i vincoli (4.30), (4.31) e (4.32) impongono che se $l_{ji} = 1 \wedge t_{ij} + t_{ji} = 0 \Rightarrow \beta_{ij} = 1$, che significa che se un'oggetto i si trovi a sinistra di un oggetto j ma nessuno dei due è dietro all'altro, allora l'oggetto i non può essere sfilato dalla sponda destra del camion;
- i vincoli (4.33), (4.34) e (4.35) impongono che se $b_{ij} = 1 \wedge l_{ij} + l_{ji} = 0 \Rightarrow \gamma_{ij} = 1$, che significa che se un'oggetto i si trovi dietro un oggetto j ma nessuno dei due è alla sinistra dell'altro, allora l'oggetto i non può essere sfilato centralmente del camion;
- il vincolo (4.36) impone che se $s_{1i} = 1$ allora potranno esservi dei pacchi posizionati alla sua sinistra che ne blocchino lo scarico, se $s_{1i} = 0$ allora la via dovrà essere libera permettendo così lo scarico;
- il vincolo (4.37) impone che se $s_{2i} = 1$ allora potranno esservi dei pacchi posizionati alla sua destra che ne blocchino lo scarico, se $s_{2i} = 0$ allora la via dovrà essere libera permettendo così lo scarico;
- il vincolo (4.38) impone che se $s_{3i} = 1$ allora potranno esservi dei pacchi posizionati davanti a sè che ne blocchino lo scarico, se $s_{3i} = 0$ allora la via dovrà essere libera permettendo così lo scarico;
- il vincolo (4.39) impone che $\sum_{v \in \theta} s_{vi} \leq 2$ ossia che possano essere occupate al massimo 2 vie di scarico lasciandone libera almeno 1.

MODELLO 3D CON ROTAZIONE E SOVRAPPOSIZIONE

Con questo modello, evoluzione del modello 2D con rotazione, ci si propone di introdurre la sovrapposizione degli oggetti, considerando però che alcuni oggetti potranno avere come condizione quella di non poter avere altri oggetti sopra di sè. Un problema non indifferente inerente questo modello viene individuato nella stabilità degli oggetti, soluzioni che riportino oggetti sopraelevati, le cui aree di base non poggino completamente sugli oggetti sottostanti sono da ritenersi non valide. Per risolvere questo problema si è deciso di optare per una semplificazione, se un oggetto i si trova sopra ad un oggetto j, allora l'area di base dell'oggetto i dovrà essere interamente appoggiata alla faccia superiore dell'oggetto j.

Il modello 3D con rotazione e sovrapposizione introduce in più rispetto al precedente le seguenti variabili continue positive:

- la variabile intera x_i con $i \in I$ individua la coordinata sull'asse x dell'angolo in basso a sinistra dell'oggetto i;
- la variabile intera y_i con $i \in I$ individua la coordinata sull'asse y dell'angolo in basso a sinistra dell'oggetto i;
- la variabile intera z_i con $i \in I$ individua la coordinata sull'asse z dell'angolo in basso a sinistra dell'oggetto i;
- la variabile intera Ω_i con $i \in I$ assume il valore d_i se l'oggetto è stato ruotato altrimenti w_i ;
- la variabile intera Δ_i con $i \in I$ assume il valore d_i se l'oggetto è stato ruotato altrimenti w_i ;

Introduce in più rispetto al precedente anche le seguenti variabili binarie:

- la variabile binaria l_{ij} con $i, j \in I$ assume il valore 1 se l'oggetto i è situato alla sinistra dell'oggetto j, altrimenti è 0;
- la variabile binaria b_{ij} con $i, j \in I$ assume il valore 1 se l'oggetto i è situato al di sotto dell'oggetto j, altrimenti è 0.
- la variabile binaria r_i con $i \in I$ assume il valore 1 se l'oggetto i è ruotato su se stesso di 90° , ne risulta che w_i e d_i sono invertiti, altrimenti è 0;
- la variabile binaria t_{ij} assume il valore 1 se l'oggetto i ha sopra di sé l'oggetto j, altrimenti è 0;
- la variabile binaria f_{ij} assume il valore 1 se l'oggetto i ha sopra di sé l'oggetto j e la faccia superiore dell'oggetto i è a contatto con la faccia inferiore dell'oggetto j, altrimenti è 0;
- la variabile binaria k_{ij} con $i, j \in I$ assume il valore 1 se l'oggetto i si trova alla sinistra o/e dietro un oggetto j oppure al di sotto di esso.

Inoltre la variabile binaria k_{ij} viene utilizzata per imporre che un oggetto possa trovarsi alla sinistra o dietro ad un altro oggetto ma non al di sotto di un altro e viceversa, imponendo ciò che segue:

$$\begin{aligned}
\min \quad & D & (4.44) \\
\text{s.t.} \quad & k_{ij} \leq l_{ij} + l_{ji} + b_{ij} + b_{ji} & i < j, \quad i, j \in I & (4.45) \\
& 2k_{ij} \geq l_{ij} + l_{ji} + b_{ij} + b_{ji} & i < j, \quad i, j \in I & (4.46) \\
& 1 - k_{ij} = t_{ij} + t_{ji} & i < j, \quad i, j \in I & (4.47) \\
& x_i + \Omega_i \leq W & i, j \in I & (4.48) \\
& y_i + \Delta_i \leq D & i, j \in I & (4.49) \\
& z_i + h_i \leq H & i \in I & (4.50) \\
& y_i - y_j + M_d b_{ij} \leq M_d - \Delta_i & i, j \in I & (4.51) \\
& x_i - x_j + M_w l_{ij} \leq M_w - \Omega_i & i, j \in I & (4.52) \\
& z_i - z_j + M_b t_{ij} \leq M_b - h_i & i, j \in I & (4.53) \\
& z_i - z_j + M_b f_{ij} \geq -M_b - h_i & i, j \in I & (4.54) \\
& x_i - x_j \leq M_w(1 - f_{ij}) & i, j \in I & (4.55) \\
& y_i - y_j \leq M_d(1 - f_{ij}) & i, j \in I & (4.56) \\
& x_i - x_j + \Omega_i - \Omega_j \geq -M_w(1 - f_{ij}) & i, j \in I & (4.57) \\
& y_i - y_j + \Delta_i - \Delta_j \geq -M_d(1 - f_{ij}) & i, j \in I & (4.58) \\
& f_{ij} \leq t_{ij} & i \in I & (4.59) \\
& M_b \sum_{i \in I} f_{ij} \geq z_j & i \in I & (4.60) \\
& \Delta_i = d_i(1 - r_i) - w_i r_i & i, j \in I & (4.61) \\
& \Omega_i = w_i(1 - r_i) - d_i r_i & i, j \in I & (4.62) \\
& b_{ij}, l_{ij}, t_{ij}, f_{ij}, k_{ij} \in \{0, 1\} & i \neq j, \quad i, j \in I & (4.63) \\
& x_i, y_i, z_i, r_i \in \mathbb{Z}^+ & i \in I & (4.64) \\
& & & (4.65)
\end{aligned}$$

Spieghiamo ora il significato di ciascun vincolo:

- i vincoli (4.45), (4.46) e (4.47) impongono che presi due oggetti uno sia dietro e/o alla sinistra dell'altro oppure sia sotto di esso;
- il vincolo (4.48) impone che dato il valore x_i , corrispondente alla coordinata x dell'angolo sinistro più arretrato dell'oggetto, sommandoci la larghezza w_i dell'oggetto e ottenendo quindi la coordinata x dell'angolo destro più arretrato, questa sia contenuta nell'intervallo $[0, W]$;
- il vincolo (4.49) impone che dato il valore y_i , corrispondente alla coordinata y dell'angolo sinistro più arretrato dell'oggetto, sommandoci la profondità d_i dell'oggetto e ottenendo quindi la coordinata y dell'angolo sinistro più avanzato, questa sia contenuta nell'intervallo $[0, D]$;
- il vincolo (4.50) impone che dato il valore z_i , corrispondente alla coordinata z dell'angolo sinistro più arretrato dell'oggetto, sommandoci l'altezza h_i dell'oggetto e ottenendo quindi la coordinata z dell'angolo sinistro più in alto, questa sia contenuta nell'intervallo $[0, H]$;
- il vincolo (4.51) dice che se $b_{ij} = 1$ allora impone che l'oggetto i si trovi dietro l'oggetto j , altrimenti il vincolo viene disattivato;
- il vincolo (4.52) dice che se $l_{ij} = 1$ allora impone che l'oggetto i si trovi alla sinistra l'oggetto j , altrimenti il vincolo viene disattivato;
- il vincolo (4.53) dice che se $t_{ij} = 1$ allora impone che l'oggetto i si trovi al di sotto dell'oggetto j , altrimenti il vincolo viene disattivato;
- il vincolo (4.54) dice che se $f_{ij} = 1$ allora impone che l'oggetto i si trovi al di sotto dell'oggetto j e a contatto con lo stesso, altrimenti il vincolo viene disattivato;
- i vincoli (4.55) e (4.56) impongono che se un oggetto i è sotto e a contatto con un oggetto j , allora la coordinata dell'angolo più vicino all'origine dovrà trovarsi all'interno del piano individuato dalla base superiore dell'oggetto i ;
- i vincoli (4.57) e (4.58) impongono che se un oggetto i è sotto e a contatto con un oggetto j , allora la coordinata dell'angolo più lontano all'origine dovrà trovarsi all'interno del piano individuato dalla base superiore dell'oggetto i ;
- il vincolo (4.59) impone che un oggetto i possa essere al di sotto e a contatto con un oggetto j solo se l'oggetto i è al di sotto dell'oggetto j ;

-
- A 3D plot showing a rectangular box with internal shelves. The box is defined by the axes X, Y, and Z. The X-axis ranges from 0.0 to 2.5, the Y-axis from 0 to 14, and the Z-axis from 0.0 to 2.5. The box is filled with a light blue color, and the internal shelves are represented by horizontal lines within the box.

33

5

Test e validazione

Una attività di fondamentale importanza è stata quella di testare i modelli e archiviare i dati.

ISTANZE

Una *istanza*^[k] è un insieme formato dai pacchi da disporre nel contenitore e il contenitore stesso.

Un gruppo di istanze è un insieme di istanze diverse tra loro accomunate tra loro dalle dimensioni simili o dal numero di oggetti.

Inizialmente ogni qual volta testavamo un modello dopo aver apportato una modifica, eseguivamo un centinaio di test con un gruppo di istanze completamente randomico, l'obiettivo in questo caso era quello di individuare imperfezioni o testare la tenuta dei vincoli. Successivamente si passava all'esecuzione di 100 test per ciascuna delle istanze riportate di seguito.

	width_a	width_b	depth_a	depth_b
0	0.5	2.45	0.5	2.45
1	0.5	1.50	0.5	4.00
2	1.5	2.45	0.5	4.00
3	0.5	1.50	3.0	4.00
4	1.5	2.45	3.0	4.00
5	0.1	1.00	0.1	1.00
6	0.1	1.00	3.0	4.00
7	2.0	2.45	3.0	4.00
8	2.0	2.45	2.0	2.45
9	0.1	1.00	0.1	4.00

Table 5.1: Le dimensioni dei gruppi di istanze eseguite

La tabella riporta gli intervalli entro cui si è deciso randomicamente le dimensioni dei pacchi, ogni gruppo di istanze aveva dimensioni particolari entro cui dovevano trovarsi i pacchi:

- **Larghezza:** [width_a, width_b]
- **Profondità:** [depth_a, depth_b]

NUMERO DI OGGETTI

Fin dal modello 2D abbiamo capito che il numero di pacchi per ogni istanza sarebbe stato in numero limitato in quanto dopo 7 pacchi i tempi di esecuzione esplodevano, abbiamo quindi deciso di scegliere in modo randomico in un intervallo [3,10] il numero di pacchi da assegnare a ciascuna istanza, imponendo un *time limit*^[g] di 300 secondi, questo portava ad ottenere soluzioni ottime e soluzioni *best bound*^[g], l'analisi dei risultati sarà divisa in queste due categorie infatti.

Nel grafico sopra riportato (5.1) viene riportato come ordinata il numero di oggetti e sull'ascissa il tempo espresso in secondi, come si può vedere man mano che gli oggetti aumentano il tempo aumenta notevolmente passando da pochi secondi a minuti, risultano molto interessanti quelle istanze che nonostante abbiano

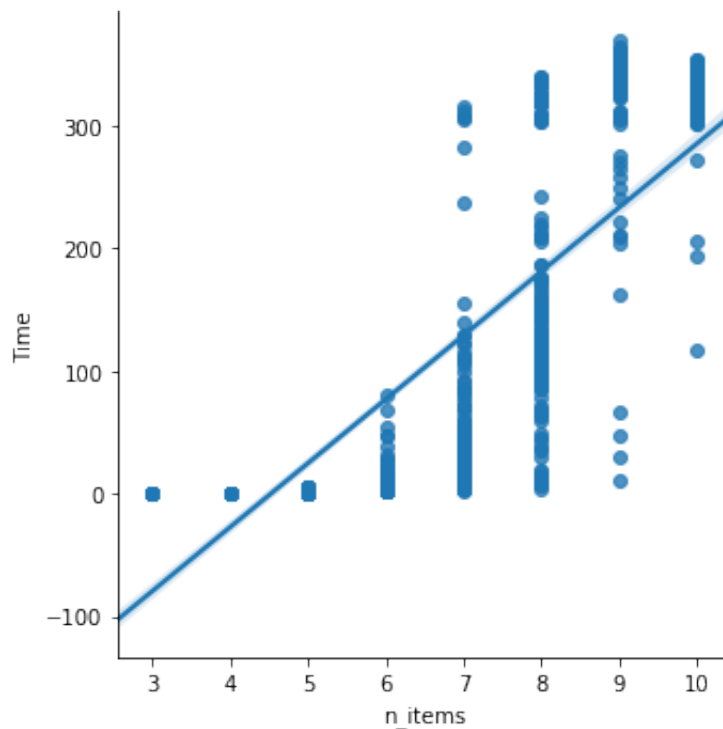


Figure 5.1: Due contenitori completamente pieni

molti oggetti richiedano meno tempo per essere eseguiti, forse questo dovuto alle particolari dimensioni degli oggetti.

VALIDAZIONE SOLUZIONI FORNITE

Di fondamentale importanza per il confronto è stato che le soluzioni fornite dal modello fossero corrette, questo ha richiesto lo sviluppo di alcune funzione che controllassero la validità delle stesse. Per fare questo sono state realizzate le seguenti funzioni:

- **CheckFeasible**: utilizzata per controllare che non vi fossero pacchi che si intersecassero con altri;
- **ChackSequence**: realizzata in due versioni:
 - **2D**: considerando che tutti gli oggetti fossero alla stessa altezza controllava che ciascun pacco avesse almeno una via attraverso cui essere

scaricato.

- **3D**: considerava anche la sovrapposizione tra pacchi e oltre a controllare di avere almeno una delle tre direzioni libere rispetto l'altezza a cui si trovava controllava anche se sopra ciascun pacco al momento dello scarico non vi fossero altri pacchi.
- **CheckStable**: utilizzata per controllare la stabilità degli oggetti, questo verificava che l'area di base inferiore poggiasse completamente sulle basi superiori degli oggetti sottostanti;
- **ProjectionCommonArea**: utilizzata per verificare se due parallelepidi avessero intersezioni tra le loro proiezioni rispetto a 2 assi.

Inoltre per natura stessa dei modelli essi restituiscono la soluzione ottima, quindi possiamo dire che se le soluzioni riportate vengono individuate entro il tempo limite e vengono verificate dalle funzioni precedentemente riportate allora queste possono dirsi effettivamente ottime e reali, perfette candidate per un confronto con la soluzione corrispondente fornita dal modello.

6

Tecnologie e strumenti

Questo capitolo riporta le tecnologie e gli strumenti utilizzati durante il corso dello stage.

TECNOLOGIE

PYTHON



Figure 6.1: Logo Python

Python è un linguaggio di programmazione tra i più famosi attualmente disponibili, è un linguaggio ad oggetto, sintatticamente semplice e intuitivo che permette di avere un approccio ai problemi molto più operativo. Python è stata la base

tecnologica su cui si è basato lo stage e gli strumenti utilizzati. Oltre ad aver usato il linguaggio Python abbiamo utilizzato alcune librerie fornite dalla libreria standard di Python.

GOOGLE OR-TOOLS

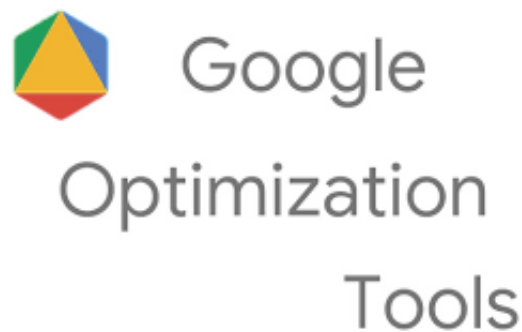


Figure 6.2: Logo Python

Google Or-Tools è una libreria open source sviluppata da Google che permette di risolvere problemi di ottimizzazione potendo scegliere se utilizzare solver commerciali come *CPLEX*^[g] o *Gurobi*^[g] oppure open source come *SCIP*^[g] o *CBC*^[g]. Or-Tools fornisce la libreria per diversi linguaggi come:

- C++
- Java
- Python
- C#

Nel nostro caso dato si è deciso di utilizzare la libreria per il linguaggio Python, utilizzandola per realizzare dei modelli di programmazione lineare intera. La libreria è stata imposta dall'azienda dopo il primo periodo di studio degli strumenti e delle tecnologie.

CBC



Figure 6.3: Logo Python

Cbc (Coin-or branch and cut) è un solver di programmazione lineare intera scritto in linguaggio C++, il motivo per cui si è scelto questo è la compatibilità con la libreria Google Or-Tools e la sua natura open source.

BOOST



Figure 6.4: Logo Python

Boost è un insieme di librerie C++ open source che permettono di aumentare le capacità applicative del linguaggio. Viene utilizzato per progetti open source e commerciali. Il suo utilizzo si è reso necessario in quanto le librerie fornite dall'azienda dipendevano da esso, in particolare grazie alla capacità di esporre classi C++ al linguaggio Python.

PANDAS



Figure 6.5: Logo Python

Pandas è una libreria open source utilizzata con il linguaggio Python, essa è fondamentale per eseguire analisi dei dati, inoltre offre delle funzionalità per la lettura dei dati da file `xlsx`, `csv`, `txt` e la manipolazione degli stessi. Inoltre fornisce dei comodi metodi di scrittura su file anche verso \LaTeX , libreria consigliata dall'azienda.

MATPLOTLIB



Figure 6.6: Logo Matplotlib

La libreria Python Matplotlib è stata utilizzata durante lo stage per la realizzazione dei grafici e su di essa si basano le immagini 2D e 3D dei pacchi precedentemente mostrate, è stato fondamentale per poter eseguire una verifica qualitativa delle soluzioni. Inoltre è stata utilizzata per realizzare grafici importanti grazie ai quali spiegare in modo efficace alcuni concetti.

MULTIPROCESSING



Figure 6.7: Logo Python

La libreria Multiprocessing è stata utilizzata per seguire un numero maggiore di test in modo concorrente superando gli impedimenti del *Global Interpreter Lock*^[g]. Consigliata dall'azienda in quanto i tempi di esecuzione dei test non utilizzando la concorrenza si attestavano alle 10 ore circa, con la suddetta libreria ci attestiamo circa sulle 4 ore.

STRUMENTI

DROPBOX



Figure 6.8: Logo Python

Dropbox è un servizio di archiviazione condivisa in cloud molto famoso e utilizzato nel mondo, permette di condividere file e documenti di vario genere con altri collaboratori, utile per la documentazione. Durante lo stage è stato utilizzato per condividere il diario e le presentazioni con i colleghi.

GITHUB



Figure 6.9: Logo Python

GitHub è un servizio di versionamento basato su repository *git*^[g]. Questo servizio permette di creare repository pubbliche e private con cui versionare il proprio progetto, aspetto importante nello sviluppo software.

VISUAL STUDIO CODE



Figure 6.10: Logo Python

Visual Studio Code è un editor open source che grazie alle estensioni installabili permette di utilizzare lo stesso editor per moltissimi linguaggi di programmazione. Inoltre grazie all'estensione per git è possibile versionare direttamente dall'editor, le estensioni permettono anche di eseguire una verifica della sintassi del codice su cui si lavora a livello statico in modo da individuare eventuali errori ed è integrato in esso il terminale con cui lanciare gli script Python.

JUPYTER NOTEBOOK



Figure 6.11: Logo Python

Jupyter Notebook è applicazione web open source che permette di creare e condividere documenti che condividono del codice. Supporta molti linguaggi di programmazione tra i quali Python e permette una programmazione interattiva generalizzata dal tool IPython. Fondamentale per semplicità di utilizzo e per la possibilità di esportare gli script in molti formati.

PIP



Figure 6.12: Logo Python

PIP è un gestore di pacchetti Python fondamentale per l'installazione di moduli non presenti nella libreria standard, di facile utilizzo ha permesso di installare velocemente moduli indispensabili per il proseguimento dello stage.

7

Risultati

Nel seguente capitolo riporteremo i risultati ottenuti dai confronti tra le soluzioni fornite dai modelli e le corrispondenti fornite dall'euristica.

Per ciascun modello verranno riportate due tabelle con i risultati ottenuti, queste verranno divise nelle due categorie di soluzioni ottenute:

- ottime;
- best bound.

Le tabelle riporteranno il numero di istanze per gruppo, errore relativo ε_r nel rapporto tra i metri lineari ottenute, errore assoluto ε_a espresso in metri e il tempo medio di esecuzione di quel gruppo di istanze. Successivamente verranno riportate alcune osservazioni sui risultati di alcuni gruppi di istanze e un rapporto generale sulla bontà delle soluzioni dei modelli aldilà dei gruppi di istanze, ricordiamo che sia il modello che l'euristica venivano eseguite sulla stessa istanze e con le stesse caratteristiche, questo permette un confronto oggettivo.

RISULTATI MODELLO 2DR

Di seguito verranno riportati i risultati ottenuti dal confronto tra soluzioni ottime fornite da euristica e modello 2DR:

	#ist	ε_r	ε_a	Time		#ist	ε_r	ε_a	Time
0	64.0	3.89	0.23	40.95	0	36.0	7.35	0.59	318.86
1	73.0	11.90	0.81	31.51	1	27.0	15.98	1.48	314.78
2	76.0	0.94	0.10	19.76	2	24.0	0.91	0.17	316.74
3	84.0	12.29	1.26	19.79	3	16.0	17.26	2.62	308.85
4	75.0	0.00	0.00	27.69	4	25.0	0.00	0.00	316.80
5	73.0	14.17	0.11	12.58	5	27.0	16.16	0.21	311.99
6	78.0	6.60	0.47	20.95	6	22.0	19.40	1.70	326.03
7	76.0	0.00	0.00	36.62	7	24.0	0.00	0.00	326.78
8	81.0	0.00	0.00	23.70	8	19.0	0.00	0.00	319.78
9	81.0	10.34	0.45	10.60	9	19.0	20.58	1.10	307.55

Table 7.1: Risultati 2DR ottimi Table 7.2: Risultati 2DR best bound

Le prime osservazioni che mi sento di riferire è il perché gruppi di istanze come il 4,7,8 riportino ε_r e ε_a entrambe uguali a 0, questo perché mediamente i pacchi di tali gruppi di istanze avevano dimensioni che non permettevano di affiancare tra loro i pacchi, per questo ogni caso si è ridotto ad allinearli uno dietro l'altro portando ad ottenere sempre lo stesso risultato. Sembra essere interessante anche il gruppo di istanze 9 che riporta mediamente il tempo di esecuzione minore, questo gruppo di istanze è un formata da mix di pacchi stretti e corti e stretti e lunghi, questo porta gli errori ε_r per quanto riguarda le soluzioni ottime ad un valore del 10% di errore, mentre invece con le soluzioni best bound questo aumenta fino al 20% aumentando così considerevolmente, altra considerazione interessante è come ci si aspetterebbe di veder diminuire gli errori con le soluzioni best bound, questo non accade ma anzi aumentano il distacco dalle soluzioni fornite dall'euristica, segno che le soluzione best bound fornite dal modello sono comunque soluzione valide per il confronto.

RISULTATI MODELLO 2DRS

Di seguito verranno riportati i risultati ottenuti dal confronto tra soluzioni ottime fornite da euristica e modello 2DRS:

	#ist	ε_r	ε_a	Time
0	76.0	4.74	0.26	41.72
1	75.0	12.44	0.83	27.96
2	82.0	0.56	0.07	20.97
3	72.0	14.59	1.62	29.67
4	81.0	0.00	0.00	28.00
5	75.0	14.87	0.12	17.89
6	78.0	9.18	0.68	17.22
7	85.0	0.00	0.00	13.99
8	79.0	0.00	0.00	25.25
9	89.0	7.69	0.32	12.87

Table 7.3: Risultati 2DRS ottimi

	#ist	ε_r	ε_a
0	24.0	5.84	0.52
1	25.0	14.26	1.29
2	18.0	0.99	0.18
3	28.0	15.82	2.40
4	19.0	0.00	0.00
5	25.0	16.20	0.20
6	22.0	20.83	1.97
7	15.0	0.00	0.00
8	21.0	0.00	0.00
9	11.0	23.35	1.06

Table 7.4: Risultati 2DRS best bound

Le prime osservazioni da riportare è che l'euristica nel disporre i pacchi rispettando la corretta sequenza di ordine, li inserisce ordinando i pacchi in base all'ordine di scarico così da essere sicuro che ciascun pacco non abbia davanti a se pacchi di ordini successivi, questo però preclude la possibilità di avere quelle soluzioni, questo però porta ad avere anche soluzioni in cui però non sia garantita la stabilità generale degli oggetti. Possiamo osservare come un aumento della complessità del problema non abbia portato però ad un aumento degli errori generalizzato ma come in alcuni gruppi di istanze sia diminuito mentre i tempi d'esecuzione siano aumentati generalmente. Le istanze con gli errori maggiori restano sempre le stesse del precedente modello, in particolare salta all'occhio come l'errore relativo del gruppo di istanze 3 si attesti al 20%, questo gruppo è formato da pacchi stretti e lunghi, valore importante che porta ad uno scarto tra modello di quasi 3 metri.

RISULTATI MODELLO 3D

Di seguito verranno riportati i risultati ottenuti dal confronto tra soluzioni ottime fornite da euristica e modello 3D:

	#ist	ε_r	ε_a	Time
0	76.0	5.05	0.25	39.15
1	77.0	11.11	0.67	31.87
2	84.0	0.84	0.10	33.15
3	72.0	9.83	0.96	34.07
4	79.0	0.00	0.00	36.59
5	65.0	9.90	0.08	13.53
6	76.0	6.33	0.45	26.50
7	76.0	0.00	0.00	39.09
8	78.0	0.00	0.00	25.45
9	76.0	10.73	0.37	20.24

Table 7.5: Risultati 3D ottimi

	#ist	ε_r	ε_a
0	24.0	5.24	0.49
1	23.0	12.87	1.22
2	16.0	0.80	0.11
3	28.0	21.01	2.95
4	21.0	0.00	0.00
5	35.0	14.47	0.17
6	24.0	16.85	1.55
7	24.0	0.00	0.00
8	22.0	0.00	0.00
9	24.0	13.96	0.59

Table 7.6: Risultati 3D best bound

Le istanze che sono state generate non contenevo solo oggetti *stackable*^[g], ma anche oggetti *fragili* che sopra di sè non potevano avere nessun'altro, quindi nell'ordinare gli oggetti l'euristica inseriva nel container prima gli oggetti *fragili* comportando così che sopra di essi non potesse esservi altri oggetti. Dobbiamo Gli errori in generale comunque si sono mantenuti simili a quelli precedentemente riportati, nonostante con il modello in questione fosse possibile avere sotto di sè solo un oggetto mentre per l'euristica fosse possibile avere più di un oggetto.

8

Conclusion

Nel seguente capitolo verranno riportate le conclusioni che si è potuto trarre alla fine di questo stage.

CONSUNTIVO FINALE

Prima di iniziare lo stage era stato concordato insieme con l'azienda la pianificazione dello stage stesso, definendo obiettivi da raggiungere e calendarizzandoli sulle 320 ore disponibili al fine di poterli soddisfare appieno. La pianificazione iniziale è stata rispettata nella sua visione generale andando a modificarne invece le scadenze orarie, la formazione è stata più breve del previsto visto che la scelta del software di modellazione algebrica è ricaduto sullo stesso Or-Tools, per la realizzazione dei modelli in generale la prototipazione e la scrittura del modello in Or-Tools è risultata meno dispendiosa del previsto, questo ha permesso di realizzare un modello non preventivato su consiglio dell'azienda e del tutor esterno. Si è deciso di eseguire i test di confronto con l'euristica nella parte finale dello stage invece che durante la loro realizzazione, questo perché l'intero sistema usato per testare i modelli fosse unificato, concentrandosi di più nell'esecuzione di test di validazione che permettessero di verificare le soluzioni fornite dai modelli.

Durata in ore		Descrizione dell'attività
32		A: Formazione
	8	• Ricerca <i>framework</i> di modellazione algebrica
	24	• Studio Google OR - Tools
64		B: Versione algoritmo 2D
	8	• Individuazione ed analisi <i>constraints</i>
	8	• Prototipazione modello
	24	• Traduzione in Python - Google OR - Tools
	24	• Test validazione
64		C: Versione algoritmo 2DR
	8	• Individuazione ed analisi <i>constraints</i>
	8	• Prototipazione modello
	24	• Traduzione in Python - Google OR - Tools
	24	• Test validazione
72		C: Versione algoritmo 2DRS
	8	• Individuazione ed analisi <i>constraints</i>
	16	• Prototipazione modello
	16	• Traduzione in Python - Google OR - Tools
	32	• Test validazione
72		D: Versione algoritmo 3D
	8	• Individuazione ed analisi <i>constraints</i>
	16	• Prototipazione modello
	16	• Traduzione in Python - Google OR - Tools
	32	• Test validazione
16		E: Test confronto modello ed euristica
	4	• Confronto modello 2D-euristica
	4	• Confronto modello 2DR-euristica
	4	• Confronto modello 2DRS-euristica
	4	• Confronto modello 3D-euristica
Totale: 320		

Table 8.1: A nice table

RAGGIUNGIMENTO DEGLI OBIETTIVI

Gli obiettivi decisi prima dell'inizio dello stage e riportati nel capitolo 2, questi prevedevano il soddisfacimento di 8 obiettivi obbligatori, 4 obiettivi desiderabili e 1 opzionale, sono stati tutti soddisfatti appieno e anzi è stato realizzato un modello non preventivato.

L'obiettivo opzionale qui di seguito:

op01: evoluzione euristica, fornita dall'azienda, con nuove funzionalità

Non è stato inserito nella pianificazione in quanto si è svolto nel mentre che si realizzavano i modelli, le funzionalità sono le funzioni di verifica con cui validare le soluzioni riportate nel capitolo 5.

CONOSCENZE ACQUISITE

Le conoscenze acquisite spon

Acknowledgments

LOREM IPSUM DOLOR SIT AMET, consectetur adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetur. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec porttitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.