

Esame di Laurea in Informatica
Dipartimento di Matematica "Tullio Levi Civita"

Implementazione di modelli di programmazione matematica per problemi di bin packing

Candidato
Daniel Rossi

Relatore
prof. Luigi De Giovanni

18 Dicembre 2018



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

- 1 Introduzione
- 2 Progetto
- 3 Raggiungimento degli obiettivi
- 4 Consuntivo
- 5 Conclusioni

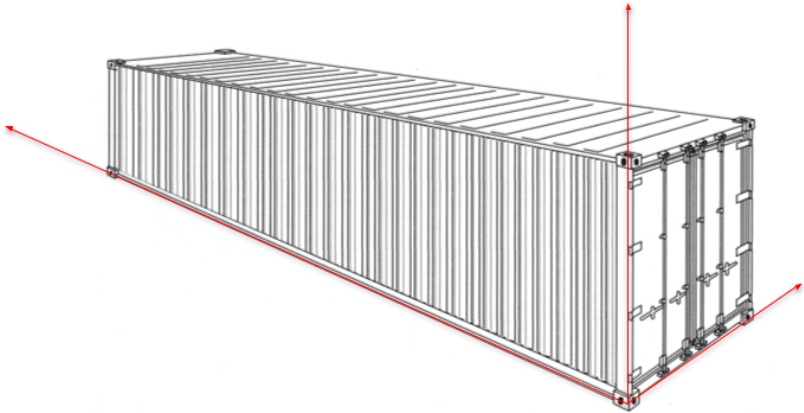


- tecnica groupage
- merci ADR
- servizio deposito

SOFTWARE SUPPORTO DECISIONALE

- operatori meno esperti
- aumento della produttività
- informazioni sullo stato dei trasporti
- stima di costi e profitti

L'azienda ha sviluppato un'euristica per l'ottimizzazione dello spazio occupato dalle merci nel container del camion



Scopo

Lo scopo dello stage è quello di realizzare dei modelli di programmazione lineare per la risoluzione dello **Strip Packing Problem** da usare per valutare l'euristica aziendale

- **2D**: versione 2D
- **2DR**: versione 2D con rotazione
- **3D**: versione 3D con rotazione e sovrapposizione

Insieme $I = \{1, \dots, n\}$ di oggetti aventi dimensioni w_i , d_i e h_i
Insieme $J = \{1, \dots, m\}$ di contenitori di dimensione W , D e H
Per ipotesi $w_i \leq W$, $d_i \leq D$ e $h_i \leq H$

Obiettivo Strip Packing

Minimizzare i metri lineari occupati dagli oggetti dell'insieme I
rispetto la profondità del contenitore

Obiettivo Bin Packing è invece minimizzare il numero di contenitori
 J che riescano a contenere tutti gli oggetti dell'insieme I

min D

s.t.

$$l_{ij} + l_{ji} + b_{ij} + b_{ji} \geq 1 \quad i < j \quad i, j \in I$$

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

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

$$x_i + w_i \leq W \quad i \in I$$

$$y_i + d_i \leq D \quad i \in I$$

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

$$x_i, y_i, w_i, d_i \in \mathbb{R}^+ \quad i \in I$$

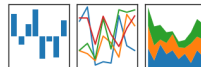
Articolo: Solving the 2D bin packing problem by means of a hybrid evolutionary algorithm, C. Blum and V. Schmid, 2013



matplotlib

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



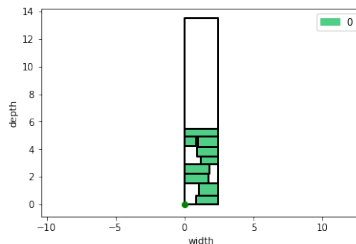
Google

Optimization

Tools

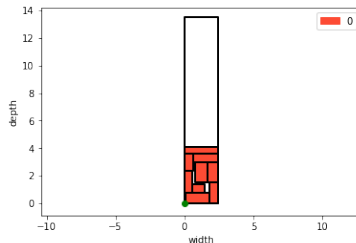
Modello 2D:

Punto di partenza



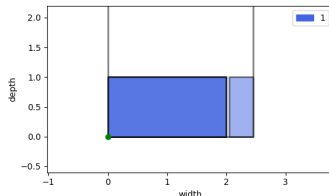
Modello 2DR:

Possibilità di sfruttare meglio
gli spazi



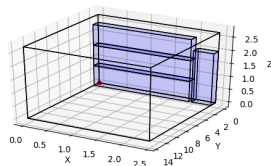
Sovrapposizione stabile:

Semplificata, un oggetto può poggiare al massimo su un altro oggetto



Oggetti stackable:

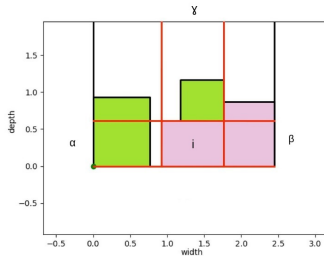
Alcuni oggetti potrebbero non essere sovrapponibili



Vie di scarico:

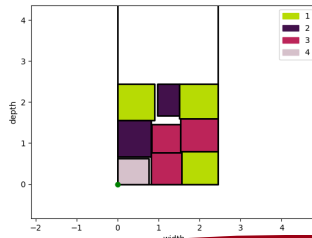
Deve essere presente almeno una via di scarico per ciascun pacco tra:

- α : scarico da sinistra
- β : scarico da destra
- γ : scarico frontale



Sequenza stabile:

Ogni oggetto non può trovarsi isolato dal resto delle merci durante il viaggio



Istanza: insieme formato dai pacchi da disporre nel contenitore

Gruppo di istanze: insieme di 100 istanze accomunate tra loro dalle loro dimensioni, identificati da un ld

Time Limit: 300 secondi

Soluzioni:

- ottime
- best bound

ld	w_a	w_b	d_a	d_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

width: $[w_a, w_b]$, depth: $[d_a, d_b]$

- **CheckFeasible**: utilizzata per controllare che non vi fossero pacchi intersecati tra loro
- **CheckSequence**: realizzata in due versioni:
 - **2D**: considerando oggetti alla stessa altezza, controlla ci sia una via di scarico
 - **3D**: considerando oggetti su più livelli, controlla ci sia una via di scarico e che non vi siano altri pacchi sopra di esso
- **CheckStable**: utilizzata per controllare sovrapposizioni stabili
- **ProjectionCommonArea**: utilizzata per verificare se due parallelepipedi avessero intersezioni tra le loro proiezioni rispetto ad una coppia di assi come altezza e profondità.

Ottime					Best bound			
<i>Id</i>	#ist	ϵ_r	ϵ_a	Time	<i>Id</i>	#ist	ϵ_r	ϵ_a
0	64	3.89	0.23	40.95	0	36	7.35	0.59
1	73	11.90	0.81	31.51	1	27	15.98	1.48
2	76	0.94	0.10	19.76	2	24	0.91	0.17
3	84	12.29	1.26	19.79	3	16	17.26	2.62
4	75	0.00	0.00	27.69	4	25	0.00	0.00
5	73	14.17	0.11	12.58	5	27	16.16	0.21
6	78	6.60	0.47	20.95	6	22	19.40	1.70
7	76	0.00	0.00	36.62	7	24	0.00	0.00
8	81	0.00	0.00	23.70	8	19	0.00	0.00
9	81	10.34	0.45	10.60	9	19	20.58	1.10

■ $\epsilon_a = Obj_h - Obj_m$

■ $\epsilon_r = \frac{\epsilon_a}{Obj_m} \cdot 100$

I 3 periodi di sviluppo prevedevano i seguenti obiettivi:

- analisi constraints
- prototipazione
- integrazione nel sistema
- testing e confronto

Due variazioni nel corso dello stage:

- Confronto con l'euristica eseguito dopo la realizzazione dei modelli
- Nuove funzioni di verifica dell'euristica sviluppate durante il testing dei modelli
- Realizzazione nuovo modello non preventivato

In generale i periodi di sviluppo hanno richiesto:

#	Obiettivi	Ore previste	Ore effettive
1	Modello 2D	96	64
2	Modello 2DR	64	64
3	Modello 2DRS	0	72
4	Modello 3D	64	72
5	Confronto con euristica	24	16

Cause cambiamenti tempistiche:

- Formazione più breve
- Prototipazione modelli 2D e 2DR più rapida
- Testing e sviluppo funzionalità euristica in parallelo

- 1 Tutti gli obiettivi sono stati completati
- 2 Realizzazione modello aggiuntivo
- 3 Valutazione positiva dei risultati forniti
- 4 Importanti conoscenze apprese

Sviluppi futuri:

- Archiviazione soluzioni modelli
- Stabilità sequenza
- Stabilità degli oggetti
- Sequenza di scarico multilivello
- Unione modelli