



**Algoritmi Euristici**  
**One Dimensional Bin Packing Problem**

*Università degli Studi Di Milano*

**Marco Odore**

10 aprile 2017

## Indice

<b>1</b>	<b>Introduzione</b>	<b>4</b>
1.1	One Dimensional Bin Packing . . . . .	4
<b>2</b>	<b>Euristiche implementate</b>	<b>4</b>
2.1	FirstFit . . . . .	4
2.2	Minimum Bin Slack . . . . .	5
2.3	MBS Sampling . . . . .	5
2.4	Variable Neighbour Search . . . . .	5

# 1 Introduzione

Lo scopo del lavoro è quello di proporre una possibile implementazione in C di diversi metodi euristici applicati al problema del *One Dimensional Bin Packing*, per la ricerca di soluzioni ottime o che comunque vi si avvicinano.

## 1.1 One Dimensional Bin Packing

Dato un multiset di  $n$  oggetti  $O = \{o_1, o_2, o_3 \dots o_n\}$ , ognuno con dimensione  $d_i$ , lo scopo è quello di minimizzare il numero di contenitori  $b_j$  (bin)  $M = \{b_1, b_2, b_3 \dots b_n\}$ , ognuno con dimensione fissata  $B$ , che contengono tali oggetti.

Il problema è soggetto a diversi vincoli:

- Ogni oggetto deve essere inserito in un solo contenitore.
- La somma delle dimensioni  $d_i$  degli oggetti  $o_i$ , nel contenitore  $b_j$ , non deve superare la dimensione del contenitore.

$$\sum_{o_i \in b_j} d_i \leq B$$

- Il numero dei contenitori  $b_j$  deve essere il minimo possibile. Si cercherà quindi di minimizzare tale funzione:

$$\min \sum_{j=1}^n y_j$$

In cui  $y_i$  è una variabile binaria associata agli  $n$  possibili contenitori  $b_j$  (il caso peggiore contempla un contenitore per ogni oggetto presente nel multi insieme).

Secondo la teoria della complessità, tale problema ha complessità *NP-hard*. Per tale motivo sono state studiate diverse tecniche euristiche, con lo scopo di ottenere un trade-off tra velocità di esecuzione e ottimalità delle soluzioni generate.

## 2 Euristiche implementate

Per la risoluzione del problema sono state implementate due principali euristiche costruttive greedy:

- FirstFit
- Minimum Bin Slack (MBS)

Che poi sono servite da base per altre due meta euristiche:

- MBS Sampling
- Variable Neighbour Search (VNS)

### 2.1 FirstFit

Tale algoritmo è molto banale, e si basa sull'idea greedy che, scorrendo iterativamente la lista di oggetti, se nel contenitore  $b_j$  corrente c'è abbastanza spazio, allora vi si inserisce l'oggetto corrente  $o_i$ .

---

**Algorithm 1** FirstFit

---

```
1: for obj in objectList do
2:   for bin in binList do
3:     if obj fit in bin then
4:       Pack object in bin
5:       break
6:     end if
7:   end for
8:   if obj did not fit in any available bin then
9:     Create new bin and pack object in it
10:  end if
11: end for
```

---

**2.2** Minimum Bin Slack**2.3** MBS Sampling**2.4** Variable Neighbour Search