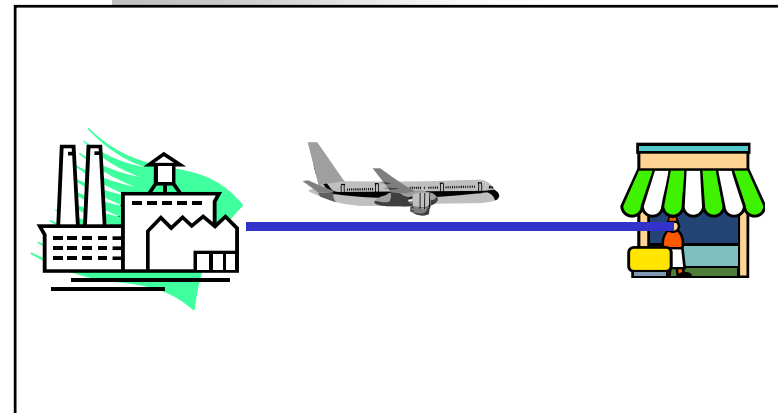




Il problema dello zaino: dalla gita in montagna ai trasporti internazionali

Luca Bertazzi



Il problema dello zaino



Zaino:

- capacità B

Oggetti (items):

- numero n

- indice $i = 1, 2, \dots, n$

- valore p_i

- peso w_i

Quali oggetti inserire nello zaino al fine di massimizzare il valore totale?

Applicazioni classiche

Gita in montagna



Sbarco sulla luna



Applicazioni finanziarie

Un investitore ha un capitale B .

Può sottoscrivere progetti di investimento.
Ogni progetto i :

- richiede un capitale w_i
- ha un rendimento p_i

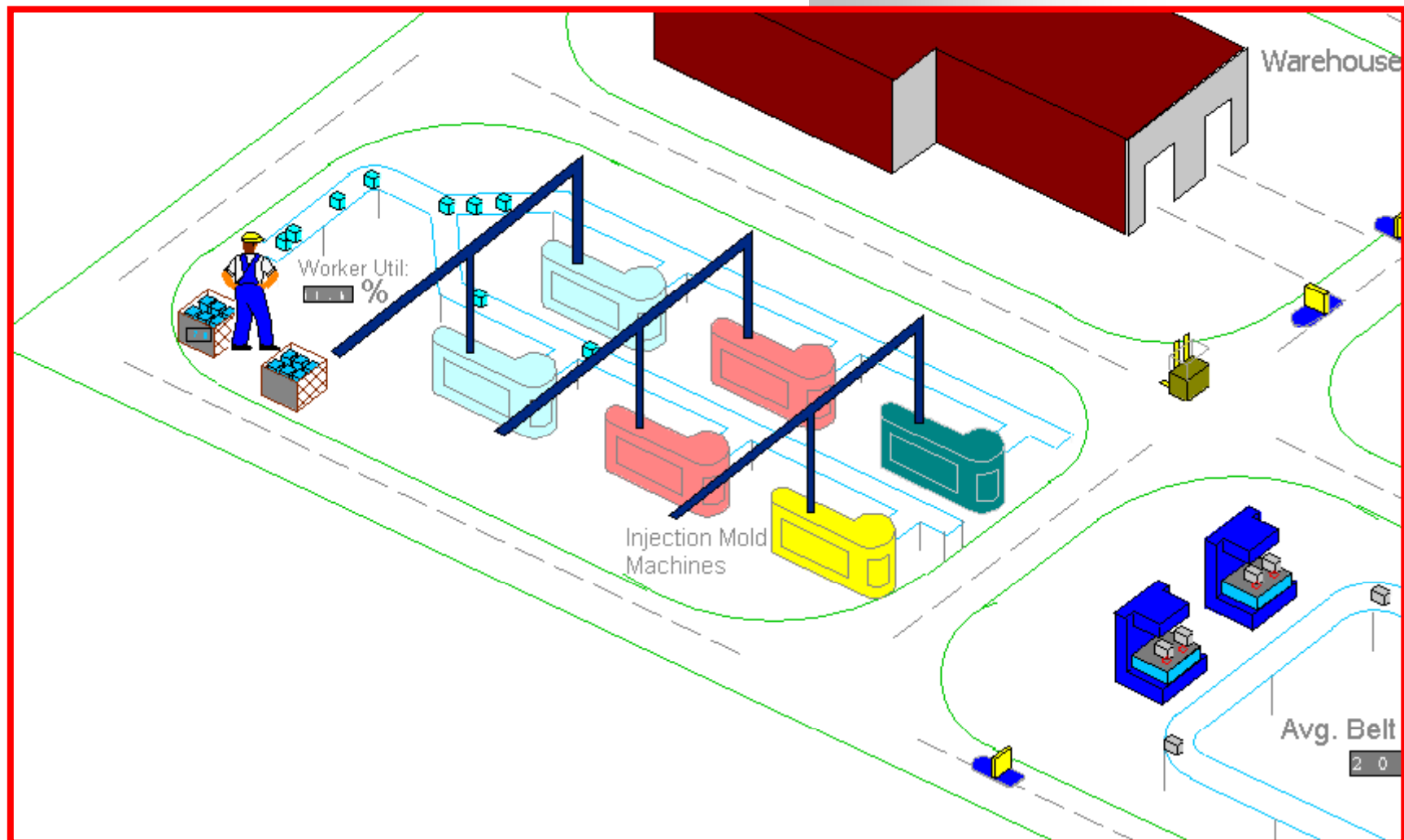
Quali progetti sottoscrivere al fine di massimizzare il rendimento totale?



Applicazioni alla Logistica



1) Gestione della produzione



Un produttore ha a disposizione una barra di ferro di una data lunghezza B .

La barra può essere tagliata in pezzi.
Ogni pezzo i :

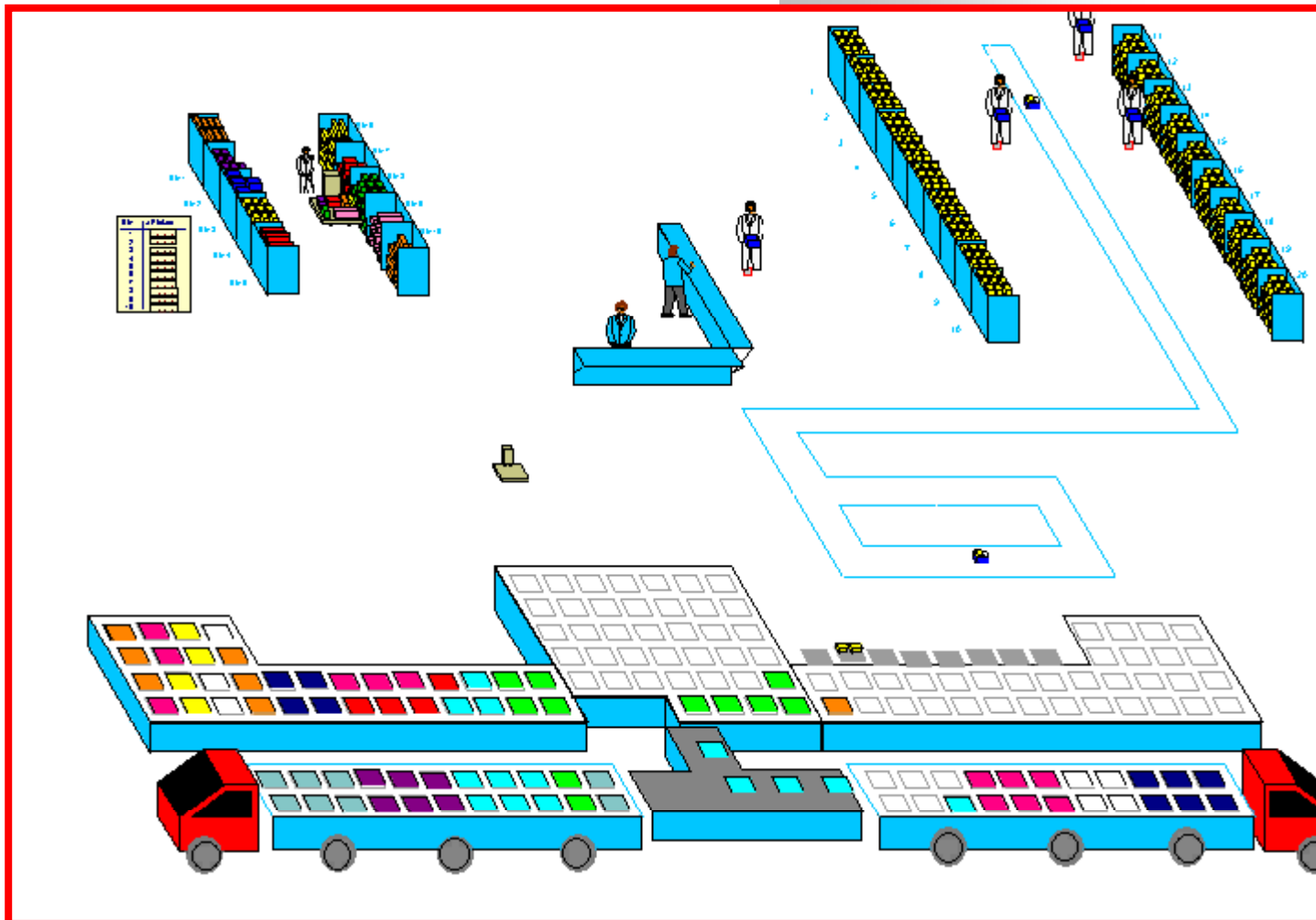
-ha una lunghezza w_i

-ha un prezzo di vendita p_i



Quali pezzi tagliare al fine di massimizzare il ricavo totale?

2) Gestione magazzini e scorte



Area di ricezione



Area di stoccaggio remoto



Area ad accesso rapido



Area di spedizione



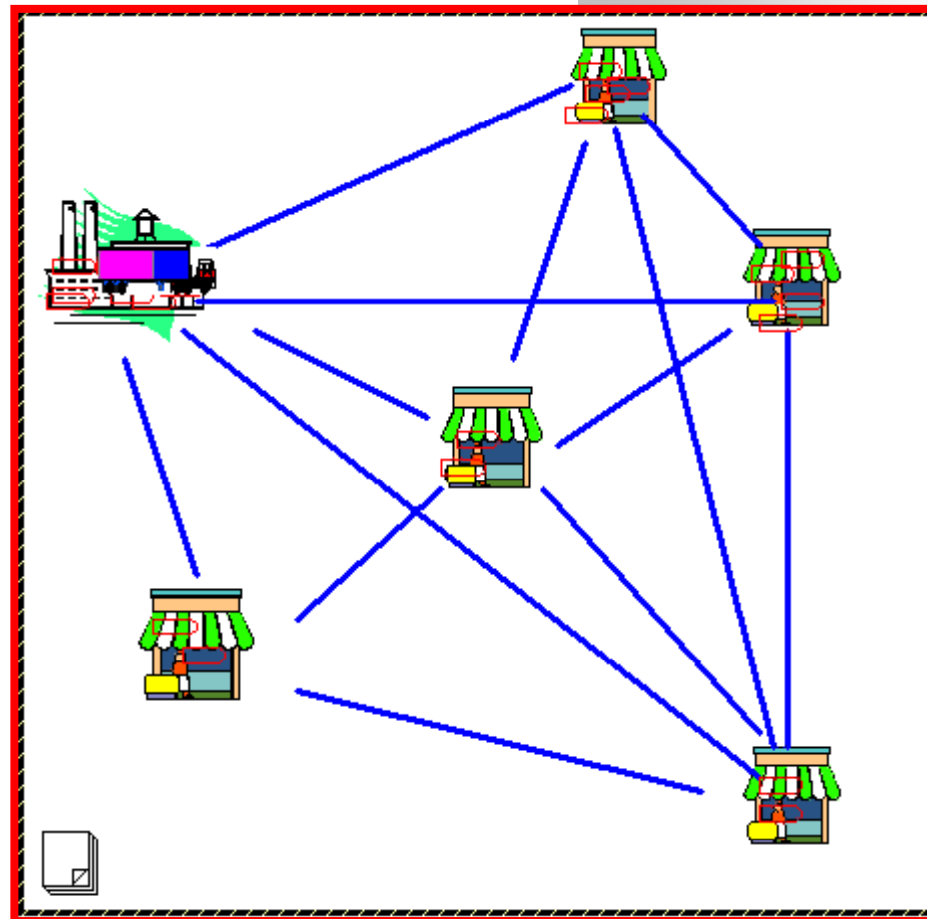
Area ad accesso rapido

- Dimensione dell'area: B
- Ogni prodotto:
 - occupa un volume w_i
 - implica un time saving p_i se collocato nell'area

Quali prodotti collocare nell'area per massimizzare il time saving totale?



3) Gestione dei trasporti



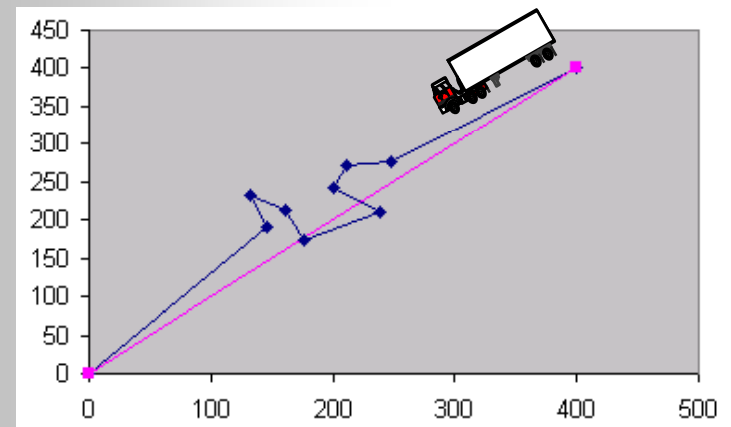
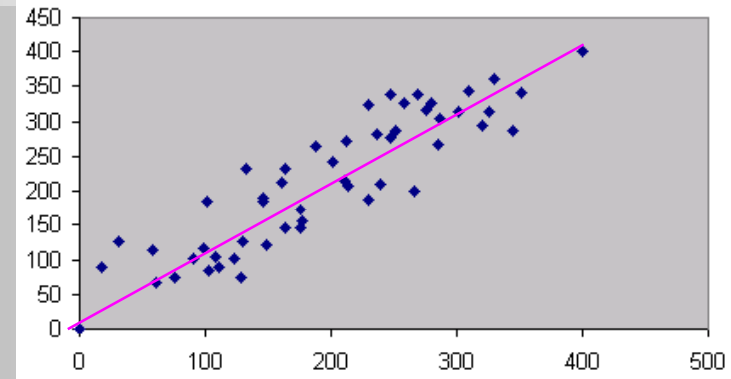
Trasporti nell'e-business

Un veicolo di capacità B , durante il viaggio di ritorno, può scegliere di servire alcuni clienti.

Ogni cliente:

- occupa un volume w_i
- fornisce un profitto p_i

Quali clienti servire per massimizzare il profitto totale?



Backhauling

Trasporti internazionali

Un gestore di cargo ha a disposizione un aereo merci di capacità B .

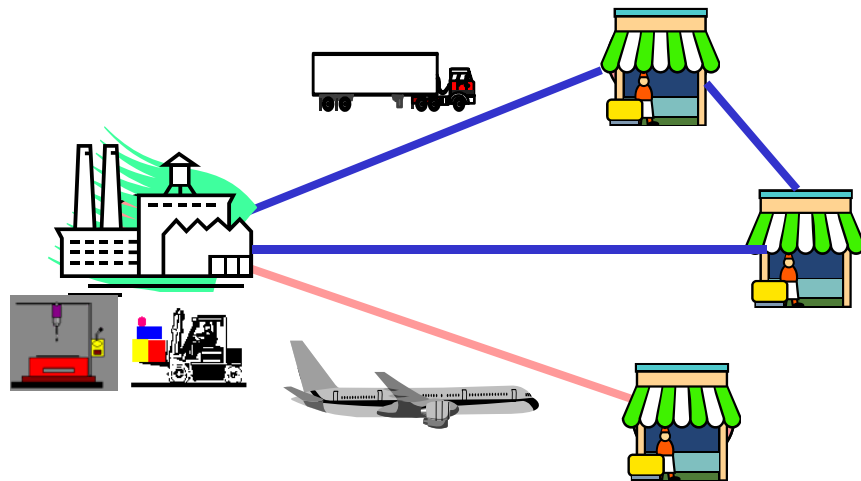
Ha una lista di carichi. Ogni carico:

- occupa un volume w_i
- fornisce un profitto p_i



Quali carichi scegliere al fine di massimizzare il profitto totale?

4) Logistica integrata



Strategie di produzione e spedizione che ottimizzano:

- Produzione
- Trasporto
- Scorte

sistemi integrati di produzione e distribuzione

Problema dello zaino: soluzioni intuitive



$$B = 15$$

$$n = 5$$

i	p_i	w_i
1	4	12
2	2	2
3	1	1
4	10	4
5	2	1

Algoritmo p_i - greedy

- Ordinare gli oggetti sulla base di p_i in modo non crescente

4 \rightarrow 1 \rightarrow 2 \rightarrow 5 \rightarrow 3

- Inserire gli oggetti fino al raggiungimento della capacità

$$4 \quad z^{p_i\text{-greedy}} = 10$$



Esiste una soluzione migliore?

$$B = 15$$

$$n = 5$$

i	p_i	w_i
1	4	12
2	2	2
3	1	1
4	10	4
5	2	1



Se inseriamo nello zaino gli oggetti 2-3-4-5 otteniamo:

Profitto totale: 15
Peso totale: 8

→ Esiste!!

Conclusione:

L'algoritmo p_i -greedy non è esatto (non assicura di ottenere una soluzione ottima)

Qual è il limite di questo algoritmo?

Non tener conto del peso degli oggetti
e quindi del profitto per unità di peso

Un secondo algoritmo

$$B = 15$$

$$n = 5$$

Profitto per
unità di peso



i	p_i	w_i	p_i / w_i
1	4	12	0,33
2	2	2	1
3	1	1	1
4	10	4	2,5
5	2	1	2

Algoritmo Greedy

- Ordinare gli oggetti sulla base di p_i / w_i in modo non crescente

4 → 5 → 2 → 3 → 1

- Inserire gli oggetti fino al raggiungimento della capacità

4 → 5 → 2 → 3

$$z^{Greedy} = 15$$



L'algoritmo Greedy è esatto?

$$B = 200$$

$$n = 3$$

i	p_i	w_i	p_i / w_i
1	2	1	2
2	100	100	1
3	100	100	1

Greedy:

$$1 \rightarrow 2 \quad z^{Greedy} = 102$$

Altra soluzione:

inserire 2 e 3

Profitto totale: 200

- Profitto totale del Greedy: 102
 - Esiste una soluzione con profitto 200
- Il profitto ottimo è almeno 200

Conclusione:

- L'algoritmo Greedy non è esatto
- Esiste almeno un caso in cui il profitto del Greedy è il 51% del profitto ottimo



E' il caso peggiore?

$$B = 200$$

$$n = 2$$

i	p_i	w_i	p_i / w_i
1	2	1	2
2	200	200	1

Greedy:

1

$$z^{Greedy} = 2$$

Altra soluzione:

inserire 2

Profitto totale: 200

L'algoritmo Greedy può generare soluzioni con profitto molto basso rispetto all'ottimo



Esiste un algoritmo con profitto minimo garantito?

L'algoritmo Ext-Greedy

Scegliere la migliore fra:

- la soluzione del Greedy
- la soluzione che contiene solo l'oggetto con profitto massimo

L'algoritmo Ext-Greedy genera un profitto pari ad almeno il 50% del profitto ottimo



Alla ricerca della soluzione ottima



Come determinare una soluzione ottima?

Algoritmo di completa enumerazione



- Generare tutte le soluzioni
- Scegliere la soluzione ammissibile con il profitto maggiore

$B = 200$

$n = 3$

i	p_i	w_i
1	2	1
2	100	100
3	100	100

1	2	3	profitto	peso
SI	NO	NO	2	1
SI	NO	SI	102	101
SI	SI	NO	102	101
SI	SI	SI	202	201
NO	NO	NO	0	0
NO	NO	SI	100	100
NO	SI	NO	100	100
NO	SI	SI	200	200

3 oggetti → 8 soluzioni

L'algoritmo di completa enumerazione genera sempre una soluzione ottima, ma...

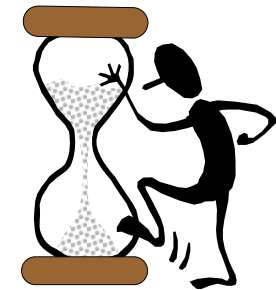
Numero di soluzioni:

n oggetti
↓
 2^n soluzioni

oggetti	numero di soluzioni
3	8
60	1.152.921.500.000.000.000
90	1.23794 E 27

Come varia il tempo di calcolo al variare di n ?

n	Tempo di calcolo
3	< 1 secondo
60	19 ore
90	392 secoli



un milione di miliardi di operazioni al secondo

→ L'algoritmo di completa enumerazione può essere impraticabile

Non solo brutte notizie...



È possibile ottenere la soluzione ottima di problemi con migliaia di oggetti in pochi secondi su un PC

n	<i>tempo</i>
5000	15 sec
15000	1 min
25000	1 min e 25 sec

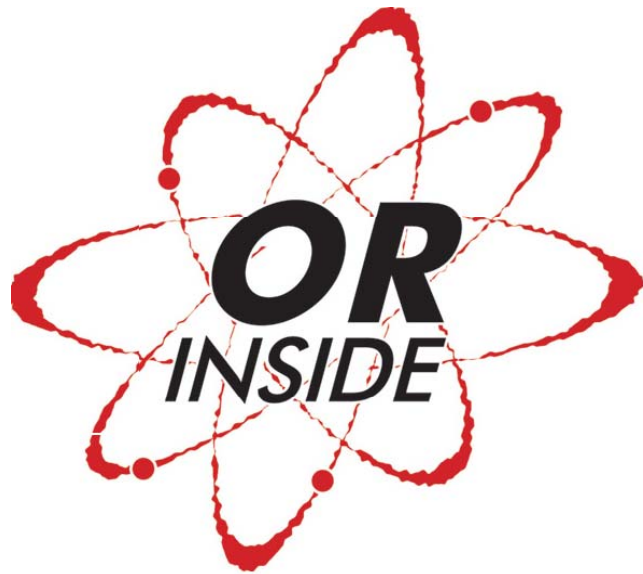
Come è possibile ottenere questo risultato?

Solver: CoinMP 1.3

PC: AMD Athlon 64 X2 Dual Core Processor 5600+ 3.4 GB RAM

Ricerca operativa

metodi quantitativi e scientifici nei
processi decisionali



Operations Research:
The Science of Better



Applicazioni:

Logistica
Finanza
Telecomunicazioni
Data Mining
Bio-informatica
Gestione delle Risorse Umane
Gestione dei Servizi Sanitari
...



What-if? What-is-best?

Metodo:

PROBLEMA



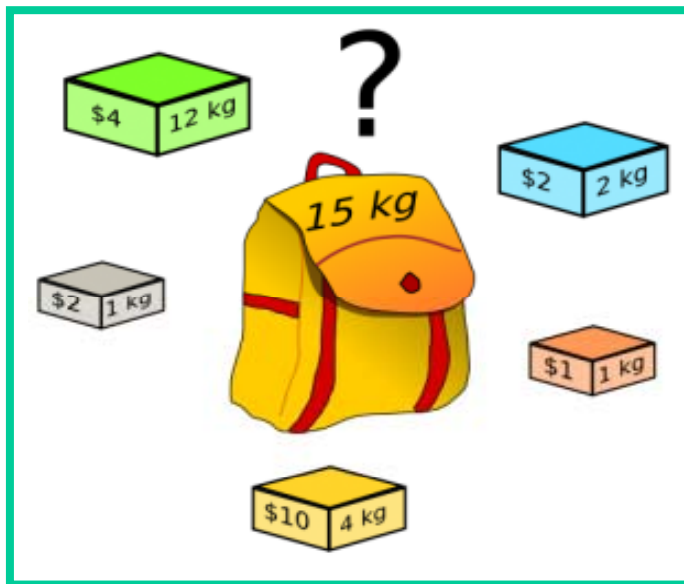
MODELLO



ALGORITMI

software

1) Definizione del problema



Quali oggetti inserire nello zaino al fine di massimizzare il profitto totale?

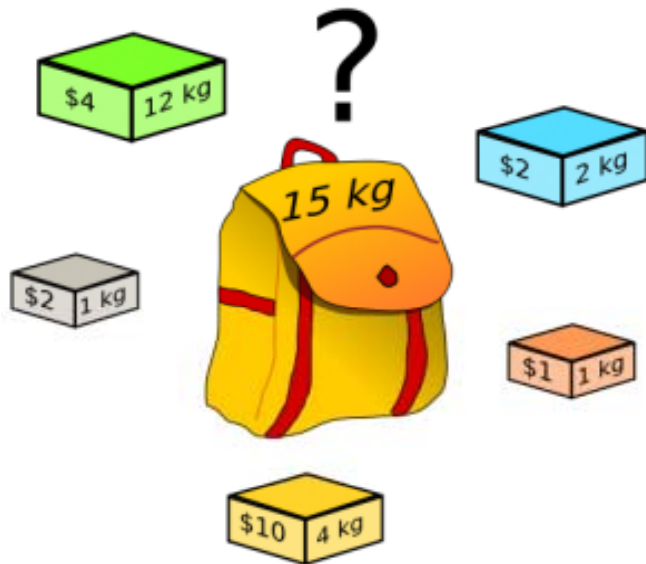
PROBLEMA

MODELLO

ALGORITMI

software

Dati:



B : capacità dello zaino

n : numero di oggetti

p_i : profitto dell'oggetto i

w_i : peso dell'oggetto i

2) Formulazione di un modello

1) Variabili decisionali:

L'oggetto i viene inserito nello zaino?

Variabile binaria $x_i \in \{0,1\}$

x_i → 0 → i non viene inserito
 x_i → 1 → i viene inserito

PROBLEMA

MODELLO

ALGORITMI

software

2) Funzione obiettivo:

massimizzare il profitto totale

Profitto oggetto i $\begin{cases} 0 \text{ se } x_i = 0 \\ p_i \text{ se } x_i = 1 \end{cases}$

Profitto totale: $\sum_{i=1}^n p_i x_i$

PROBLEMA

MODELLO

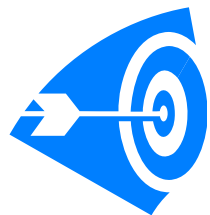
ALGORITMI

software

Funzione obiettivo:

massimizzare il profitto totale

$$\max \sum_{i=1}^n p_i x_i$$



PROBLEMA

MODELLO

ALGORITMI

software

3) Vincoli:

Vincolo di capacità

Peso oggetti inseriti \leq capacità

$$\sum_{i=1}^n w_i \cdot x_i \leq B$$

PROBLEMA

MODELLO

ALGORITMI

software

Modello:

$$\begin{aligned} \max \quad & \sum_{i=1}^n p_i x_i \\ \sum_{i=1}^n w_i x_i & \leq B \\ x_i & \in \{0,1\} \quad i = 1,2,\dots,n \end{aligned}$$

modello di programmazione
lineare intera (binaria)

PROBLEMA

MODELLO

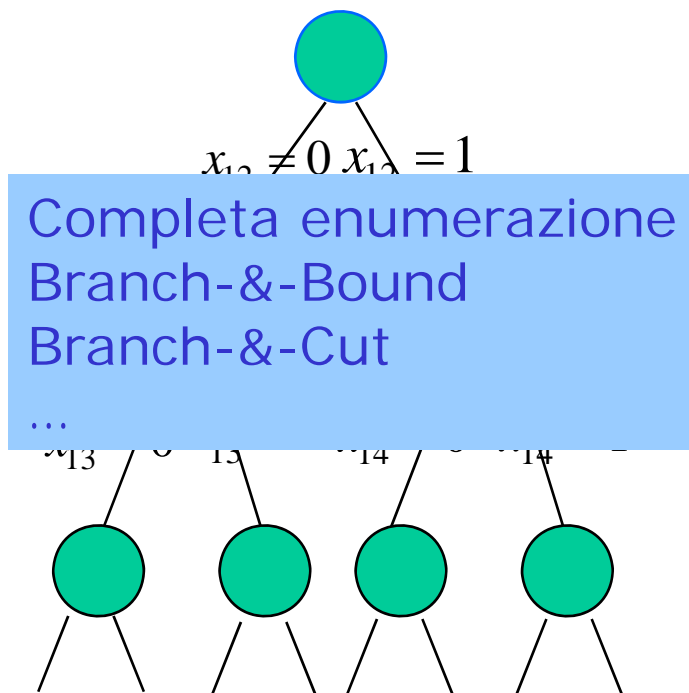
ALGORITMI

software

3) Applicazione di algoritmi

a) Esatti

soluzione ottima



PROBLEMA

MODELLO

ALGORITMI

software

b) Euristici

soluzione buona

p_i -greedy
Greedy
Ext-Greedy
PTAS
FPTAS
...

PROBLEMA

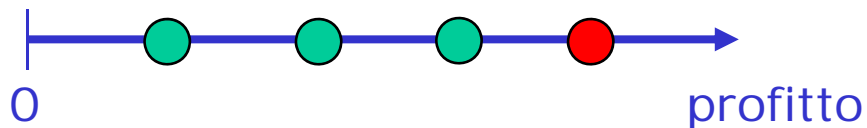
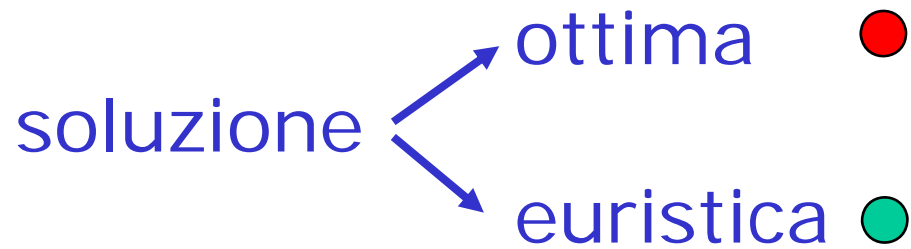
MODELLO

ALGORITMI

software

Obiettivo ideale:

- Soluzione ottima
- Tempo polinomiale



PROBLEMA

MODELLO

ALGORITMI

software




Es: p_i - greedy e Greedy

Es: Completa
enumerazione

n	n^2	2^n
10	100	1.024
20	400	1.048.576
40	1.600	1.099.511.627.776
80	6.400	1.208.925.819.614.630.000.000.000

Per il problema dello zaino:

Tempo/Sol.	Ottima	Euristica
Polinomiale		p_i -greedy Greedy ...
Esponenziale	Completa enumerazione Branch-&-Bound Branch-&-Cut ...	



Algoritmi esatti polinomiali

Non è mai stato ottenuto un algoritmo in grado di fornire la soluzione ottima del problema dello zaino in tempi polinomiali (algoritmo esatto polinomiale)

Inoltre:

Il problema dello zaino è NP-hard

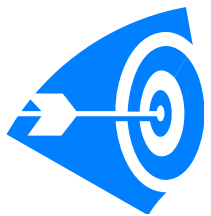
Il problema dello zaino è NP-hard

Appartiene ad una classe di problemi per i quali:

- non è mai stato trovato un algoritmo esatto polinomiale
- se si trovasse un algoritmo esatto polinomiale per uno di questi problemi, ogni problema della classe avrebbe un algoritmo esatto polinomiale

Teoria della complessità computazionale

È altamente probabile
che un algoritmo
esatto polinomiale
per il problema dello zaino
non esista



Potenziare gli

algoritmi esatti

algoritmi euristici

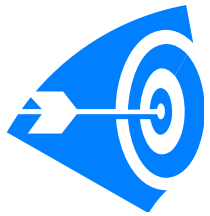
PROBLEMA

MODELLO

ALGORITMI

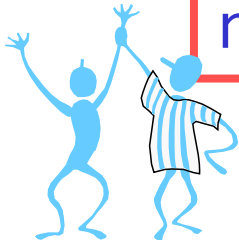
software

Algoritmi esatti:



Risolvere all'ottimo
istanze sempre più
grandi

Soluzione ottima in
meno di un secondo!



PROBLEMA



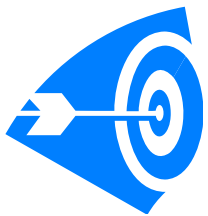
MODELLO



ALGORITMI

software

Algoritmi euristici:



Trovare soluzioni
sempre più vicine
all'ottimo

Garanzie sulla bontà
della soluzione!



PROBLEMA

MODELLO

ALGORITMI

software

4) Utilizzo di software

Gli algoritmi esatti ed euristici sono implementati in software

Esatti general purpose:

- Risolutore di Excel
- MPL
- LINGO



PROBLEMA



MODELLO



ALGORITMI

software

MPL for Windows 4.2 - [D:\luca\lavoro\di

File Edit Search Project Run View Graph

```

TITLE
    "problema dello zaino";
INDEX
    i:=1..5;
DATA
    B:=15;
    p[i]:=(4,2,1,10,2);
    w[i]:=(12,2,1,4,1);
DECISION VARIABLES
    x[i];
MODEL
    MAX SUM(i:p[i]*x[i]);
SUBJECT TO
    capacita: SUM(i:w[i]*x[i])<=B;
BINARY
    x[i];
  
```

MPL for Windows 4.2 - [View File: Copia di zaino.

File Edit Search Project Run View Graph Options

SOLUTION RESULT

Optimal solution found

MAX Z = 15.0000

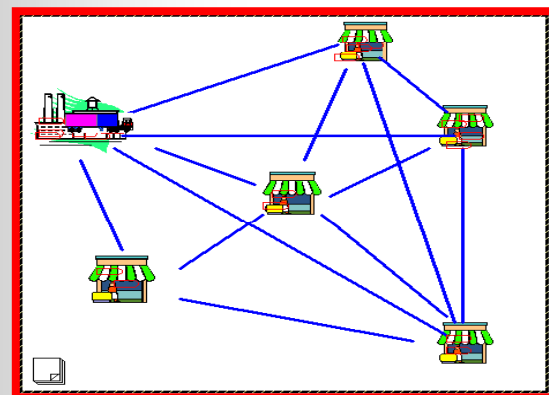
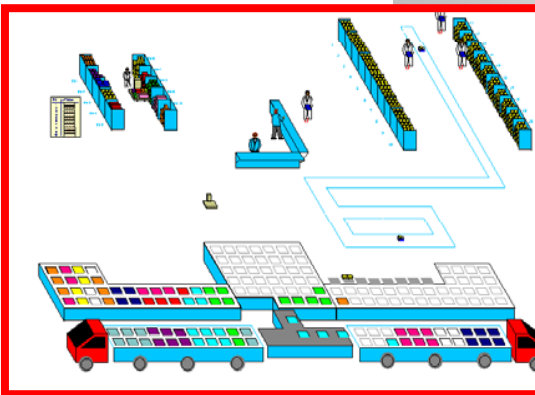
DECISION VARIABLES

VARIABLE x[i] :

i	Activity	Reduced Cost
1	0.0000	0.0000
2	1.0000	0.0000
3	1.0000	0.0000
4	1.0000	0.0000
5	1.0000	0.0000

Il problema dello zaino e la Logistica

Il problema dello zaino è uno dei sottoproblemi della logistica

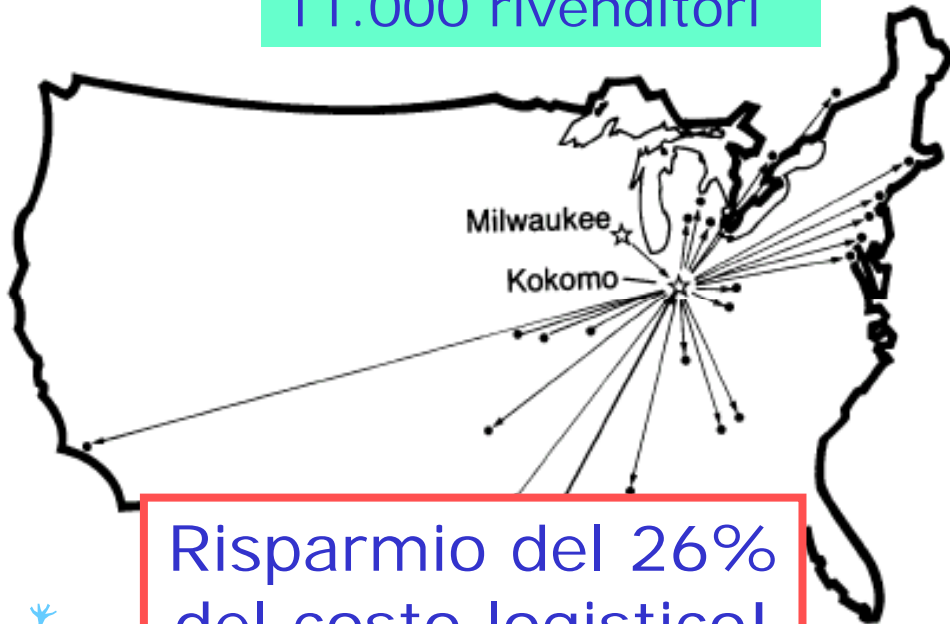


La vera sfida consiste nel trovare la soluzione migliore per l'intero sistema



GM

13.000 componenti
20.000 fornitori
164 impianti
11.000 rivenditori



Risparmio del 26%
del costo logistico!

Division Plants
• GM Assembly Plants

PROBLEMA

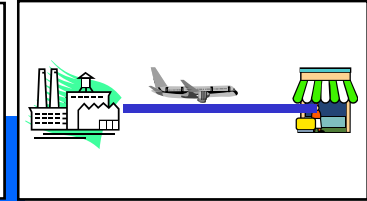
MODELLO

ALGORITMI

software



Conclusione



Un antico proverbio recita:

Se un problema non ha soluzioni, perchè preoccuparsi?
Se un problema ha soluzioni, perchè preoccuparsi?

Un problema dello zaino con
60 oggetti ha
1.152.921.500.000.000.000
soluzioni...

