

Block Notes Matematico

Block Notes Matematico è un'area "Math building block" per matematici e appassionati.

Iscriviti a Block Notes

 Post 

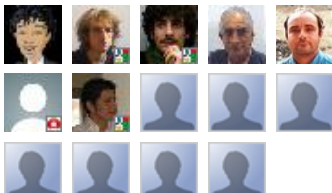
 Commenti 

Lettori fissi

Segui

con Google Friend Connect

Follower (14)



Archivio blog

► 2011 (1)

▼ 2010 (38)

- dicembre (4)
- novembre (4)
- ottobre (3)
- ▼ settembre (5)

Prolog, Web Semantico e Social Network

Schedulazioni vincenti e labirinti
tenebrosi, con ...

Un sacco di mal di testa: Il problema dello
zaino ...

Problemi matematici minori?

Programmazione lineare

- agosto (2)
- luglio (1)
- giugno (1)
- maggio (3)
- aprile (7)
- marzo (1)
- febbraio (1)
- gennaio (6)

► 2009 (27)

► 2008 (9)

Informazioni personali

Rosario Turco

[Visualizza il mio profilo completo](#)

martedì 21 settembre 2010

Un sacco di mal di testa: Il problema dello zaino (Knapsack)

Il problema dello zaino è uno dei "21 problemi NP-completi della lista di Karp". Esso si può formulare nel seguente modo: sia N , il numero totale di oggetti x_i , sia w_i il peso del singolo oggetto x_i , c_i il valore del singolo oggetto x_i e W il peso totale sopportabile dallo zaino.

La funzione obiettivo da massimizzare è in tal caso, considerando $x_i=1$ se si inserisce nello zaino o 0 in caso

contrario: $\max Z = \sum_{i=1..N} c_i x_i$ mentre il vincolo impone che: $\sum_{i=1..N} x_i w_i \leq W$

In sostanza $\max Z$ è il massimo valore ottenibile come oggetti che posso mettere nello zaino, non necessariamente N , ma nello stesso tempo il massimo numero di oggetti è limitato dal massimo peso trasportabile.

Ovviamente il problema dello zaino è isomorfo ad altri problemi simili, anziché dello zaino potrebbe essere la stiva di un aereo, o di una barca etc. e il valore ed il peso possono avere un differente significato, ma trattabile allo stesso modo matematicamente.

A seconda dei valori assunti da x_i il problema è classificato in modalità diversa. Se $x_i \in \{0,1\}$ allora si definisce

"problema dello zaino binario o 0-1"; se $x_i \leq b_i$ si definisce "problema dello zaino con limiti" ed infine se

x_i appartiene a N e per ogni i appartenente a N con $i > 0$ allora si parla di "problema dello zaino senza limiti".

Esso è risolvibile attraverso una "programmazione dinamica", termine non finalizzato ad un fine informatico, ma entrato in uso negli anni Quaranta ad indicare la suddivisione in sottoproblemi e strutture ottimali. Sulla strada del "Subset-sum" furono ideate anche tecniche crittografiche a chiave pubblica, come quella Markle-Hellman successivamente abbandonata poiché facilmente risolvibile con algoritmi lineari.

Facciamo un esempio, Nella tabella successiva abbiamo 9 possibili oggetti, con peso e valore, che potremmo mettere nello zaino tenendo presente che il massimo peso trasportabile è $W=27\text{Kg}$:

Oggetto	Peso (kg)	Valore (euro)	Valore/Peso
x_1	10	200	20
x_2	12	225	18,75
x_3	15	250	16,67
x_4	9	150	16,67
x_5	7	125	17,86
x_6	6	125	20,83
x_7	6	120	20
x_8	6	120	20
x_9	4	75	18,75

Esistono varie strategie algoritmiche euristiche e di rilassamento che per la sua soluzione non richiede molto tempo, ad esempio con tecnica "Branch e bound" in ambito della Linear Programming e dovuto a A.H. Land e A.G. Doig nel 1960. Tale

tecnica è una enumerazione implicita cioè prova a enumerare tutte le possibilità, individuando quella ottima o quella possibile, scartando le altre.

L'algoritmo Greedy di tipo euristico è dovuto, invece, a Martello e Toth (1990). Esso ordina gli oggetti in ordine decrescente di peso, per inserirli nello zaino fino all'esaurimento dello spazio disponibile. Se k è il massimo numero di oggetti che possono essere inseriti nello zaino, l'algoritmo Greedy garantisce che ne siano inseriti nello zaino almeno $k/2$.

Sono possibili molte variazioni di questo algoritmo, la più efficiente prevede di ordinare gli elementi in base al loro costo unitario, vale a dire c_i/w_i ed inserirli in ordine decrescente (euristica CUD). Questi algoritmi essendo euristici non garantiscono l'ottimalità della soluzione ma sono in grado di fornire una "buona" soluzione in tempo ragionevole. Possiamo usare GnuWin32 (gplsol), scrivendo un file in formato GMPL oppure LP; ma se volessimo adottare strategie senza "aiuti software" potremmo avere diverse soluzioni (esaminiamo il problema dello zaino binario):

- Soluzione 1: prendo gli oggetti più pesanti che non superano i 27 kg: ad esempio $x_2 + x_3$ che mi danno 27 kg ed un valore di 475 euro
- Soluzione 2: prendo gli oggetti più leggeri per aumentarne il numero ad esempio $x_6 + x_7 + x_8 + x_9$ in questo caso si ottiene 22Kg ed un valore di 530 euro (meno peso e maggior valore).
- Soluzione 3: prendo gli oggetti di maggiore valore in base al rapporto valore/peso $x_1 + x_6 + x_7 + x_9$ ovvero 26kg e 520 euro
- Soluzione 4: raggiungo i 27 kg con gli oggetti di maggiore valore $x_1 + x_5 + x_6 + x_9$ ovvero 27Kg e 525 euro

La soluzione 4, come si vede, massimizza la funzione del valore. In questo caso erano poche le variabili in gioco ma se fossero state molte sarebbe stato necessario l'"aiuto software".

OK, vi do la formulazione del problema di sopra in formato GMPL da risolvere con GLPSOL.

Eccola:

```
# This file shows how to model a knapsack problem in GMPL.
```

```
# Size of knapsack
param c;
```

```
# Items: index, size, profit
set I, dimen 3;
```

```
# Indices
set J := setof{(i,s,p) in I} i;
```

```
# Assignment
var a{J}, binary;
```

```
maximize obj :
    sum{(i,s,p) in I} p*a[i];
```

```
s.t. size :
    sum{(i,s,p) in I} s*a[i] <= c;
```

```
solve;
```

```
printf "The knapsack contains:\n";
printf {(i,s,p) in I: a[i] == 1} " %i", i;
```

```
printf "\n";

data;

# Size of the knapsack
param c := 27;

# Items: index, size, profit
set I :=
  1 10 200
  2 12 225
  3 15 250
  4 9 150
  5 7 125
  6 6 125
  7 6 120
  8 6 120
  9 4 75;

end;
```

Ora che avete il file knapsack.txt lanciate il comando:

```
glpsol --model knapsack.txt
```

e otterrete l'output:

```
K:\Work\LP>glpsol --model knapsack.txt
Reading model section from knapsack.txt...
Reading data section from knapsack.txt...
44 lines were read
Generating obj...
Generating size...
Model has been successfully generated
ipp_basic_tech: 1 row(s) and 0 column(s) removed
ipp_reduce_bnds: 1 pass(es) made, 0 bound(s) reduced
ipp_basic_tech: 0 row(s) and 0 column(s) removed
ipp_reduce_coef: 1 pass(es) made, 0 coefficient(s) reduced
glp_intopt: presolved MIP has 1 row, 9 columns, 9 non-zeros
glp_intopt: 9 integer columns, all of which are binary
Scaling...
  A: min|aij| = 4.000e+000 max|aij| = 1.500e+001 ratio = 3.750e+000
  GM: min|aij| = 1.000e+000 max|aij| = 1.000e+000 ratio = 1.000e+000
  EQ: min|aij| = 1.000e+000 max|aij| = 1.000e+000 ratio = 1.000e+000
  2N: min|aij| = 7.500e-001 max|aij| = 1.250e+000 ratio = 1.667e+000
Crashing...
Size of triangular part = 1
Solving LP relaxation...
  0: obj = 5.400000000e+002 infeas = 1.360e+001 (0)
*   2: obj = 3.500000000e+002 infeas = 0.000e+000 (0)
*   8: obj = 5.450000000e+002 infeas = 0.000e+000 (0)
OPTIMAL SOLUTION FOUND
Integer optimization begins...
+   8: mip = not found yet <= +inf (1; 0)
+  15: >>>> 5.150000000e+002 <= 5.350000000e+002 3.9% (4; 0)
+  25: >>>> 5.200000000e+002 <= 5.350000000e+002 2.9% (6; 4)
+  29: >>>> 5.250000000e+002 <= 5.250000000e+002 0.0% (1; 11)
+  29: mip = 5.250000000e+002 <= tree is empty 0.0% (0; 17)
INTEGER OPTIMAL SOLUTION FOUND
Time used: 0.1 secs
Memory used: 0.1 Mb (155039 bytes)
The knapsack contains:
1 5 6 9
Model has been successfully processed
```

Alla prox

Pubblicato da Block Notes Matematico a 14:07

Reazioni: divertente {1} interessante {1} eccezionale {1}

1 commenti:**michele** ha detto...

Caio Rosario,

Come sempre interessante anche questo argomento. Ti volevo dire che ti avevo mandato due messaggi, ma non ho avuto risposta. Ci sono stati problemi di posta? Comunque, sto leggendo un lavoro di Connes molto attuale, che ha pubblicato questo mese, ci sono molte connessioni possibili con altre aree della fisica e della teoria dei numeri. Saluti ed augurissimi da Michele

21 settembre 2010 22:02**Posta un commento**Commenta come: [Post più recente](#)[Home page](#)[Post più vecchio](#)Iscriviti a: [Commenti sul post \(Atom\)](#)