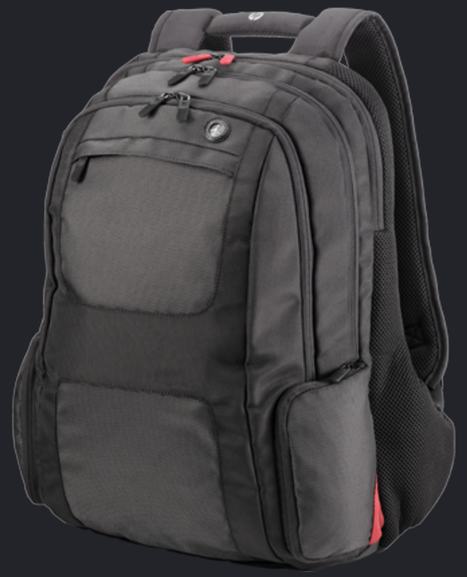
Otimização Combinatória
Profa. Luciana Buriol

Augusto Boranga
Matheus Pereira

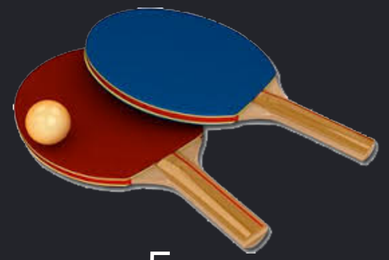
Problema da mochila com grafos de conflito / VNS

PROBLEMA



s = 0

p = 3
v = 5



p = 5
v = 5

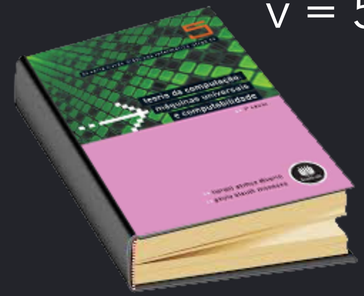
s = 0

s = 1

p = 8
v = 4

s = 1

p = 12
v = 50



p = 1
v = 80

s = 1

Problema da mochila

PROBLEMA

s = 0

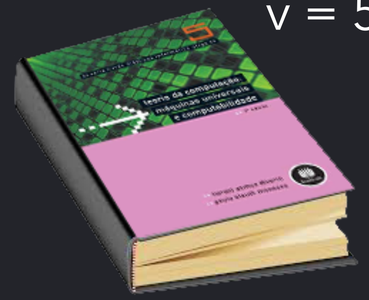
p = 3
v = 5

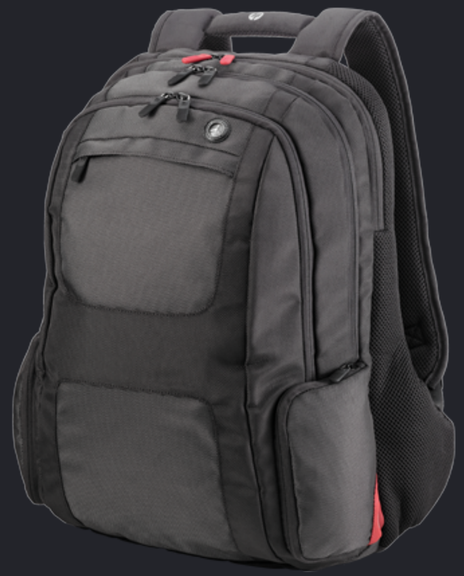
s = 1

p = 8
v = 4

s = 1

p = 12
v = 50



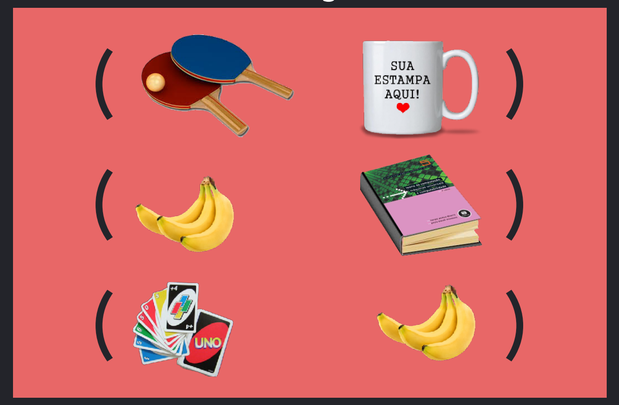


p = 5
v = 5

s = 0

p = 1
v = 80
s = 1

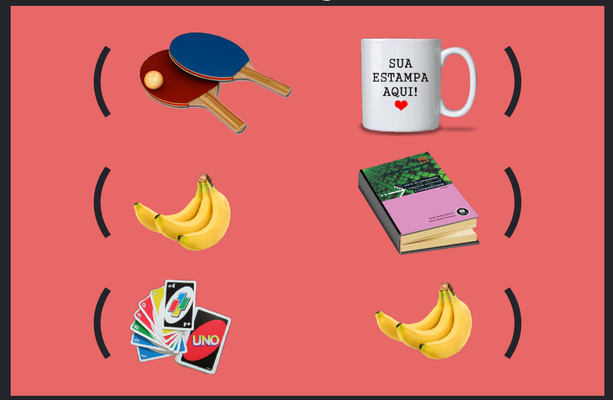
Restrições:



Problema da mochila com grafos de conflito

PROBLEMA

Restrições:



Pares de itens não podem estar simultaneamente na mochila

FORMULAÇÃO

max valor total dos itens na mochila

s.a. peso total mochila < peso máximo
+ restrições de conflito

FORMULAÇÃO

i = 1,2, …, n

j = 1,2, …, n

c = peso máximo da mochila

Pi = valor do item i

Wi = peso do item i

Xi = 1 se item i está na solução,
0 caso contrário

E = conjunto de restrições:
[(i_1, j_1), (i_2, j_2), …]

$$\textbf{max} \quad \sum_{i=1}^{n} Pi * Xi$$

$$\textbf{s.a.} \quad \sum_{i=1}^{n} Wi * Xi \leq c$$

$$Xi + Xj \leq 1 \quad \forall \ (i,j) \in E$$

$$Xi \in \{1,0\}$$

restrições de conflito

VARIABLE NEIGHBORHOOD SEARCH

Realizar a busca por máximos locais alternando sistematicamente as diferentes vizinhanças geradas

IMPLEMENTAÇÃO

python

IMPLEMENTAÇÃO

```
x = gera_solucao_inicial();

Enquanto iteração < NUM_MAX_ITERACAO
  k := 1;

  Enquanto (k <= k_max):
     N_k = gera_vizinhança_k();

     x' = escolhe_vizinho_aleatorio(N_k(x));
     x'' = encontra_maximo_local(x’);

     Se (avalia(x’') > avalia(x)):
        x := x'';
        k = 1;
     Senão:
        k += 1;
```

RESULTADOS

	Solução ótima	glpk Resultado	glpk Tempo (s)	glpk Acerto (%)	VNS Resultado	VNS Tempo (s)	VNS Acerto (%)
C10	7776	7776	0,1	100	7776	9,7	100
C1	1040	1040	0,0	100	1040	13,4	100
C3	600	600	0,0	100	590	8,8	98
R10	2007	2007	2,8	100	1996	18,6	99
R1	559	559	0,0	100	559	19,4	100
R3	1763	1763	8,0	100	1733	289,4	98
test	7310	5445	3600 (limite)	74	5066	1092,3	69
HB10	49836	17376	3600 (limite)	34	22533	88,2	45
HB11	99702	32407	3600 (limite)	32	38373	501,7	38
HB12	199413	0 (não	18000 (limite)	0	71519	5064,3	35

MÉDIAS

glpk Acerto (%)
100
100
100
100
100
100
74
34
32
0

VNS Acerto (%)
100
100
98
99
100
98
69
45
38
35

glpk: **74%**

meta-heurística: **78,2%**

CONCLUSÃO

Um algoritmo melhor de
geração de vizinhos traria melhores resultados

Tempo de execução alto

Ainda assim, resultados em média foram
melhores que o glpk