

SuperComputação

Aula 05 – Heurísticas

2021 – Engenharia

André Filipe M. Batista <andrefmb@insper.edu.br>

Hoje

- Início do projeto
- Heurísticas e problemas difíceis
- C ++ / Struct



Resolução de problemas

Resolução de problemas

Problemas difíceis aparecem em muitas áreas

1. Pesquisa Operacional (logística, produção, etc)
2. Machine Learning
3. Marketing
4. Planejamento urbano (mobilidade)

Resolução de problemas - Otimização

Função objetivo:

- algo que queremos maximizar ou minimizar

Restrições:

- definem quais possíveis soluções são válidas

Muitas classes de problemas:

1. Programação Linear / Inteira
2. Programação convexa
3. Programação não linear
4. Otimização combinatória

Resolução de problemas - Otimização

Função objetivo:

- algo que queremos maximizar ou minimizar

Restrições:

- definem quais possíveis soluções são válidas

Muitas classes de problemas:

1. Programação Linear / Inteira
2. Programação convexa
3. Programação não linear
- 4. Otimização combinatória**

Otimização combinatória

Selecionar um objeto com melhor função objetivo dentre uma coleção finita.

1. Não tem derivada
2. Não tem vizinhança
3. Coleção não é densa

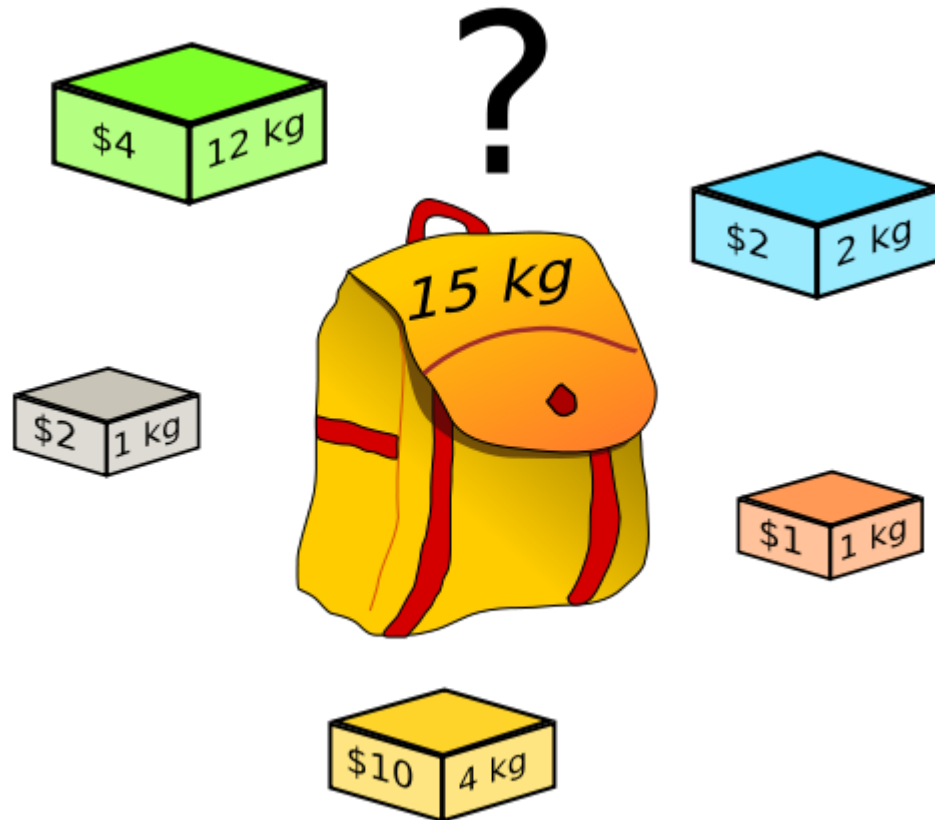
Técnicas tradicionais de cálculo e otimização não funcionam, pois nosso problema é discreto

A mochila binária



Civil War Knapsack. U.S. government image. Vicksburg National Military Park. Public domain.

A mochila binária

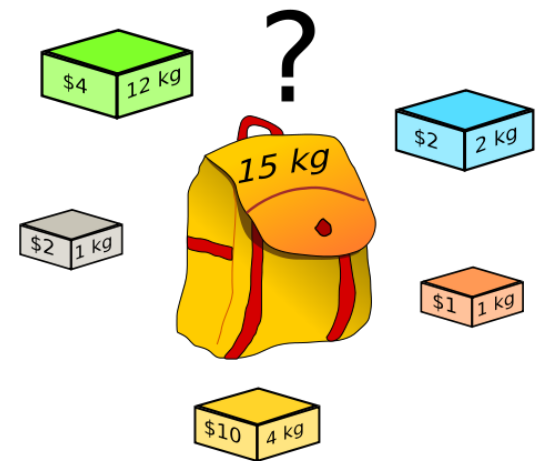


A mochila binária

Quais escolhas podem ser feitas?

Qual é a função objetivo?

Quais são as restrições?



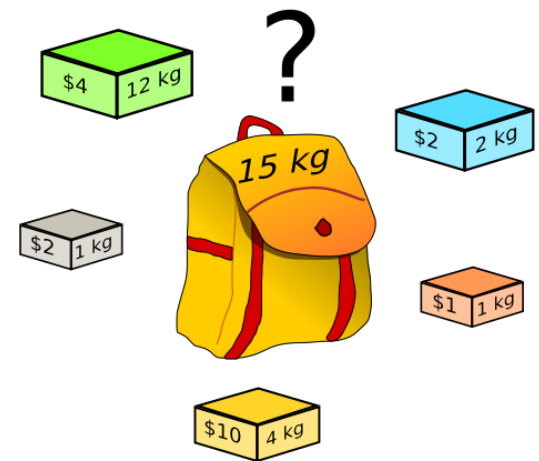
A mochila binária

Quais escolhas podem ser feitas?

- Quais produtos pegar?

Qual é a função objetivo?

Quais são as restrições?



A mochila binária

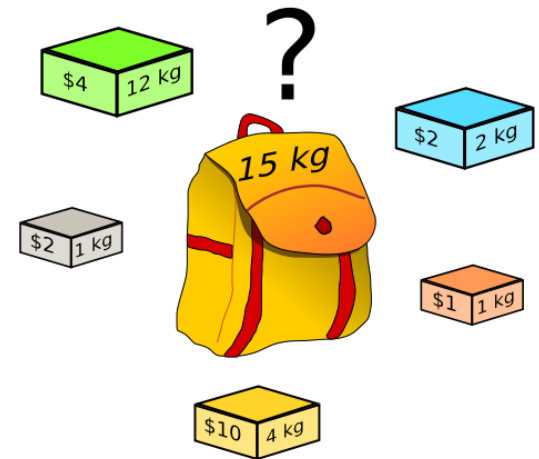
Quais escolhas podem ser feitas?

- Quais produtos pegar?

Qual é a função objetivo?

- Maximizar valor dos objetos guardados

Quais são as restrições?



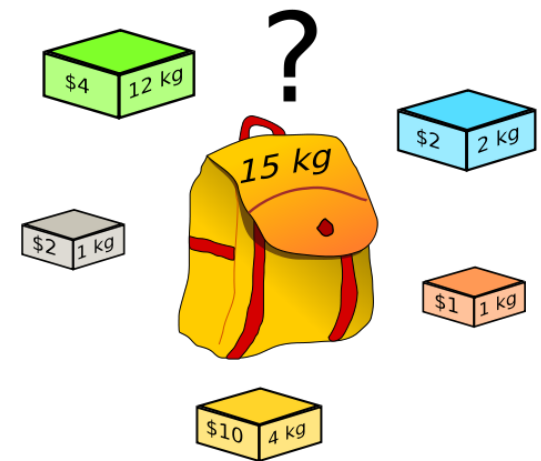
A mochila binária

Quais escolhas podem ser feitas?

- Quais produtos pegar?

Qual é a função objetivo?

- Maximizar valor dos objetos guardados



Quais são as restrições?

- Peso dos objetos não pode exceder capacidade da mochila



Como resolver esse problema?

Como resolver esse problema?

Algumas opções:

- tentar tudo e ver qual é melhor
- pegar o mais caro primeiro
- pegar o mais leve primeiro

É possível resolver de maneira eficiente?

Como resolver esse problema?

Algumas opções:

- tentar tudo e ver qual é melhor
- pegar o mais caro primeiro
- pegar o mais leve primeiro

É possível resolver de maneira eficiente?

NÃO

Heurística

"truque" usado para resolver um problema rapidamente

Por velocidade, sacrificamos ao menos um entre

- otimalidade
- corretude
- precisão
- exatidão

Heurística

"truque" usado para resolver um problema rapidamente

Ainda assim, uma boa heurística é suficiente para obter resultados aproximados ou ganhos de curto prazo.

- explorar alguma propriedade do problema
- dividir em partes menores que podem ser resolvidas rapidamente e combinar os resultados

Como resolver esse problema?

Algumas opções:

- tentar tudo e ver qual é melhor
- pegar o mais caro primeiro
- pegar o mais leve primeiro

Heurísticas para a Mochila binária



Atividade prática

Resolvendo a mochila binária (45 minutos)

1. Implementar duas heurísticas e comparar seus resultados



Comentários sobre "mais caro"



Atividade prática

Resolvendo a mochila binária (15 minutos)

1. Análise de entradas e saídas

Análise das heurísticas

Qual sua complexidade computacional?

Quando uma é melhor que a outra?

Alguma consegue o melhor valor possível?

C++ Struct

```
ConsoleApplication2 (Global Scope)
1  #include <iostream> 1
2
3  using namespace std; 2
4
5  struct Person 3
6  { 4
7
8      int citizenship; 5
9
10     int age; 6
11
12 }; 7
13
14 int main(void) { 8
15     struct Person p; 9
16
17     p.citizenship = 1; 10
18
19     p.age = 27; 11
20
21     cout << "Person citizenship: " << p.citizenship << endl; 12
22
23     cout << "Person age: " << p.age << endl; 13
24
25     return 0; 14
26 } 15
27
28
29
```


Insper

www.insper.edu.br