



Insper Supercomputação

Aula - 05

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

Resolução de problemas

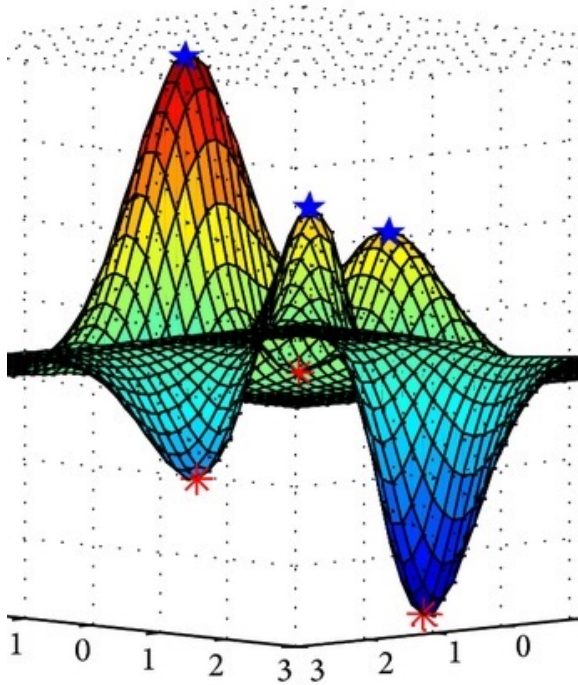
© 2000 Randy Glasbergen.
www.glasbergen.com



"THE COMPUTER SAYS I NEED TO UPGRADE MY BRAIN
TO BE COMPATIBLE WITH ITS NEW SOFTWARE."

- Problemas difíceis aparecem em muitas áreas
 - Pesquisa operacional (logística, produção, etc.)
 - Machine Learning
 - Marketing
 - 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:
 - Programação Linear / Inteira
 - Programação não-linear
 - Otimização combinatória

Otimização combinatória

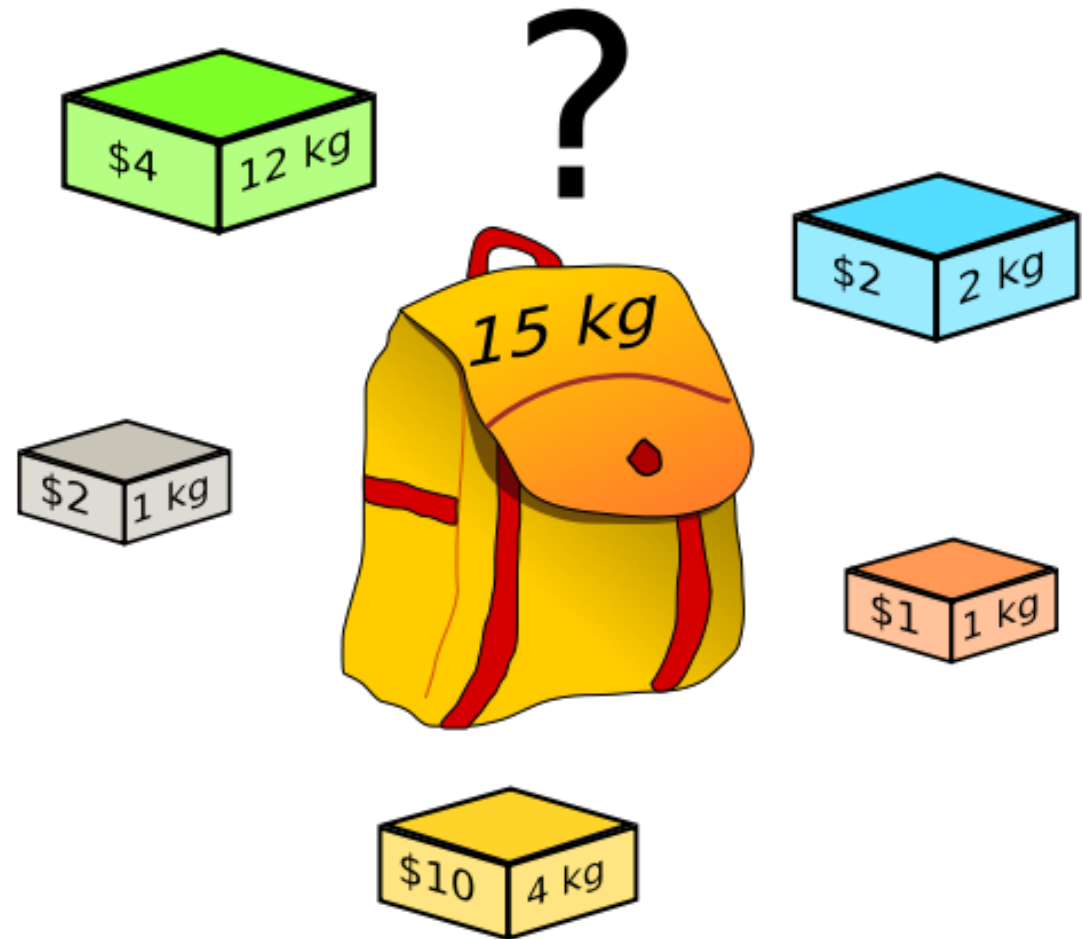
- Tem por objetivo selecionar um objeto com melhor função objetivo dentre uma coleção finita
- Não tem derivada
- Não tem vizinhança
- Coleção não é densa
- Técnicas tradicionais de cálculo e otimização não funcionam, pois nosso problema é **discreto**

Nosso problema: A mochila binária



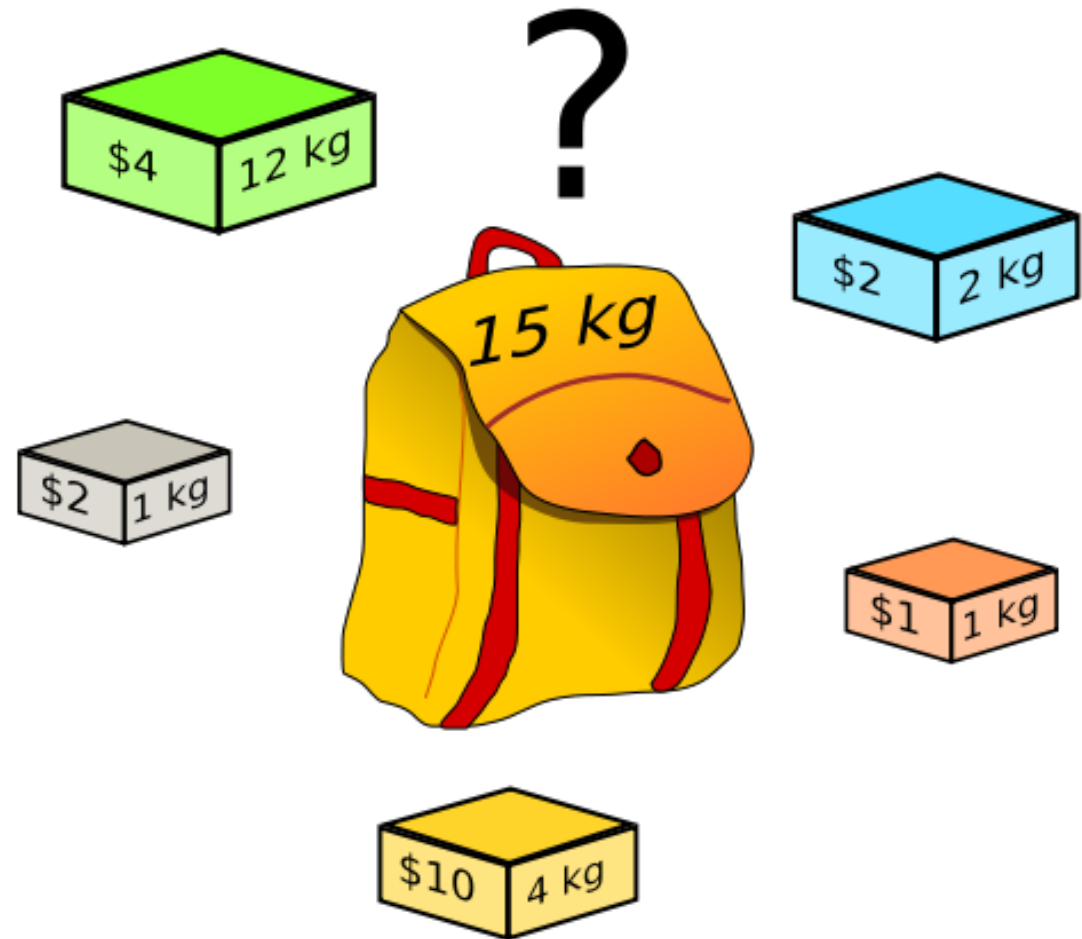
Civil War Knapsack. U.S. government image. Vicksburg National Military Park. Public domain.

A mochila binária



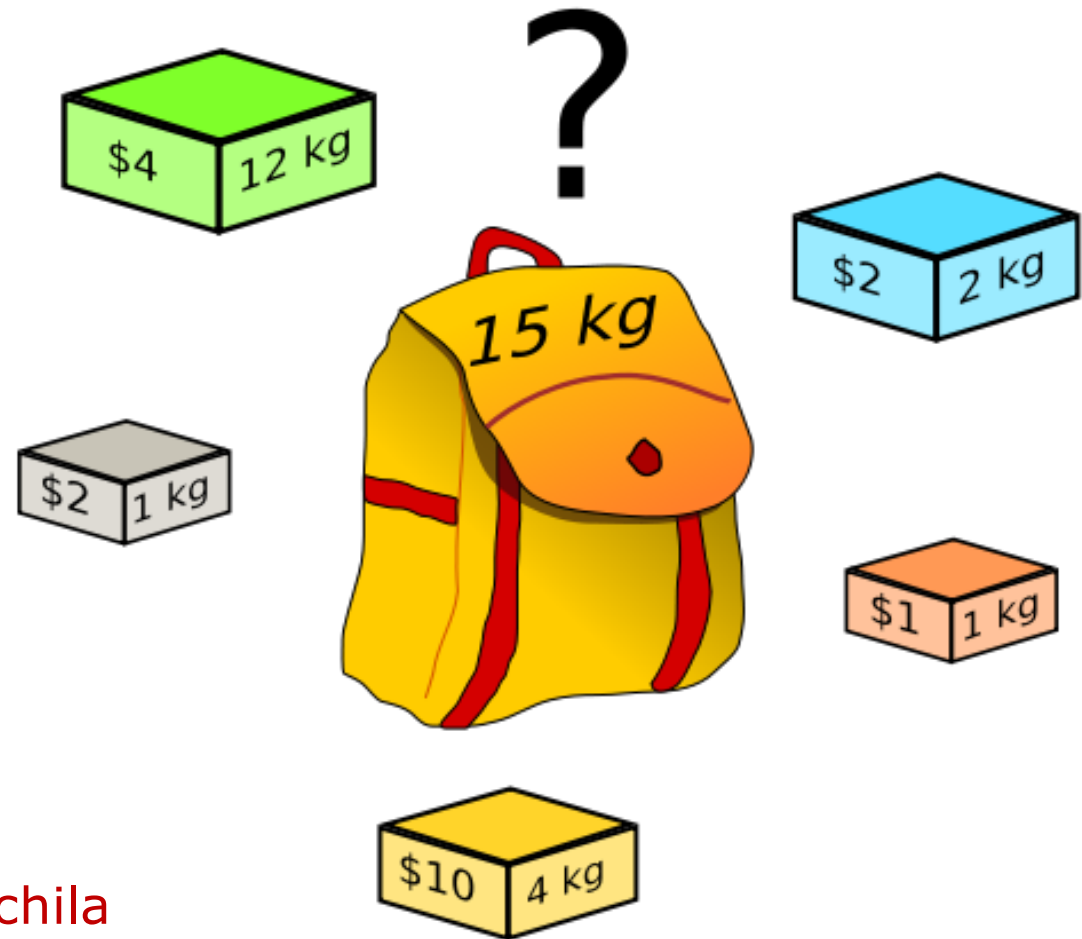
A mochila binária

- Quais escolhas podem ser feitas?
- Qual é a função objetivo?
- Quais são as restrições?



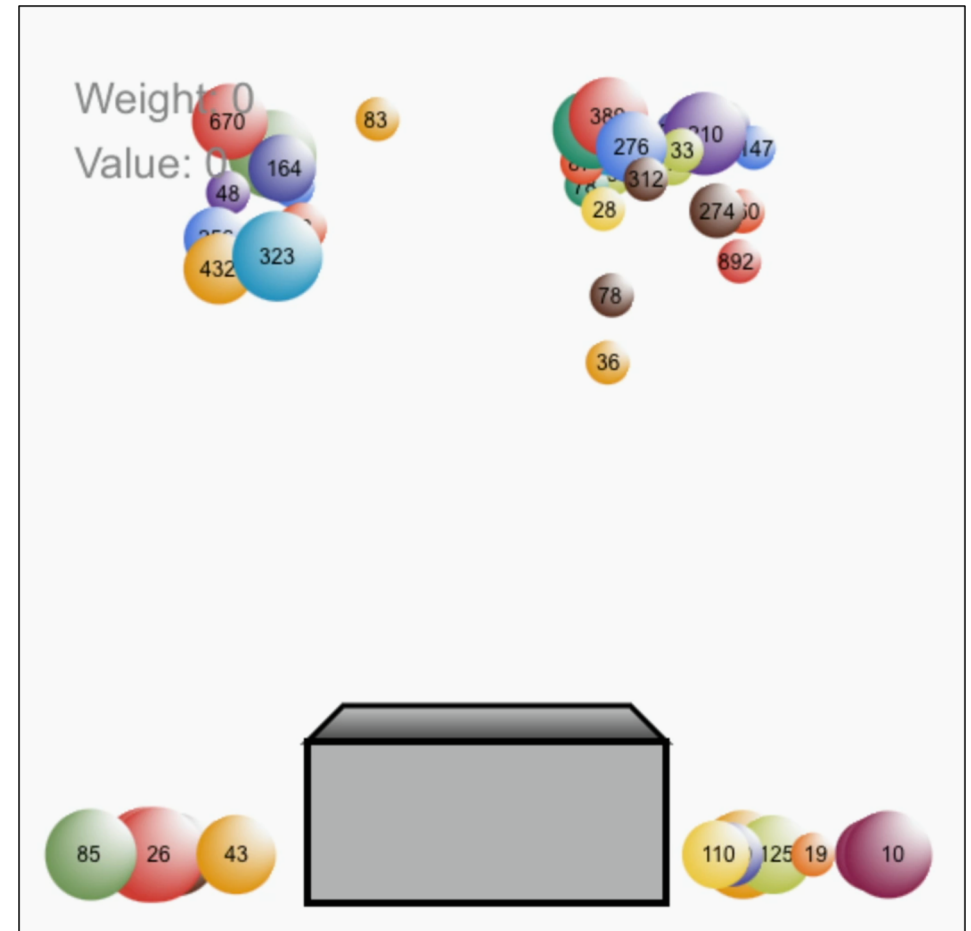
A mochila binária

- Quais escolhas podem ser feitas?
 - Quais produtos pegar?
- Qual é a função objetivo?
 - Maximizar valor os objetos capturados
- Quais são as restrições?
 - Peso dos objetos não pode exceder a capacidade da mochila



A mochila binária

Na animação ao lado, 50 itens são colocados em uma mochila. Cada item tem um valor (o número no item) e um peso (aproximadamente proporcional à área do item). A mochila é tem capacidade de 850, e nosso objetivo é encontrar o conjunto de itens que irão maximizar o valor total sem exceder a capacidade.





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



- Uma boa heurística é suficiente para obter resultados aproximados ou ganhos de curto prazo
- Processo:
 - Explorar alguma propriedade do problema
 - Dividir em partes menores, que podem ser resolvidas rapidamente e combinar os resultados

Heurística para a mochila

- Algumas opções:
- Pegar o mais caro primeiro
- Pegar o mais leve primeiro



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



Atividade prática

Resolvendo a mochila binária

Sua missão: implementar as duas heurísticas e comparar seus resultados



Dica: C++ Struct

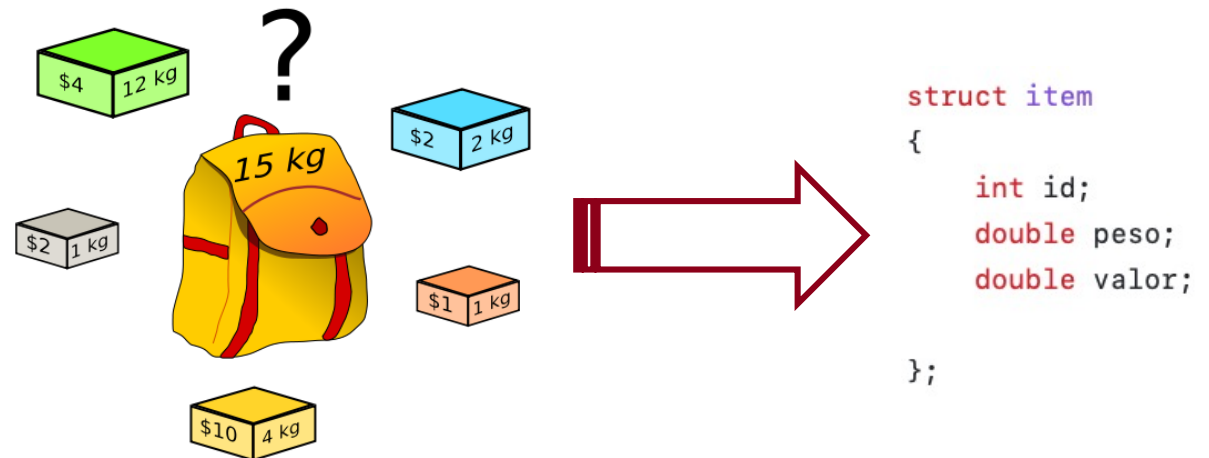
Uma boa abordagem para modelar uma mochila que é fazer uso de vector. Além disso, os itens (que possuem peso e valor) podem ser criados como structs em c++

```
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
15 int main(void) { 8
16
17     struct Person p; 9
18
19     p.citizenship = 1; 10
20
21     p.age = 27; 11
22
23     cout << "Person citizenship: " << p.citizenship << endl; 12
24
25     cout << "Person age: " << p.age << endl; 13
26
27     return 0; 14
28 } 15
29
```



Dica: C++ Struct

Uma boa abordagem para modelar uma mochila que é fazer uso de vector. Além disso, os itens (que possuem peso e valor) podem ser criados como structs em c++





Discussão

- Qual a complexidade computacional das abordagens?
- Quando uma é melhor que a outra?
- Alguma consegue obter o melhor valor possível?



Obrigado

Insper

www.insper.edu.br