

SuperComputação

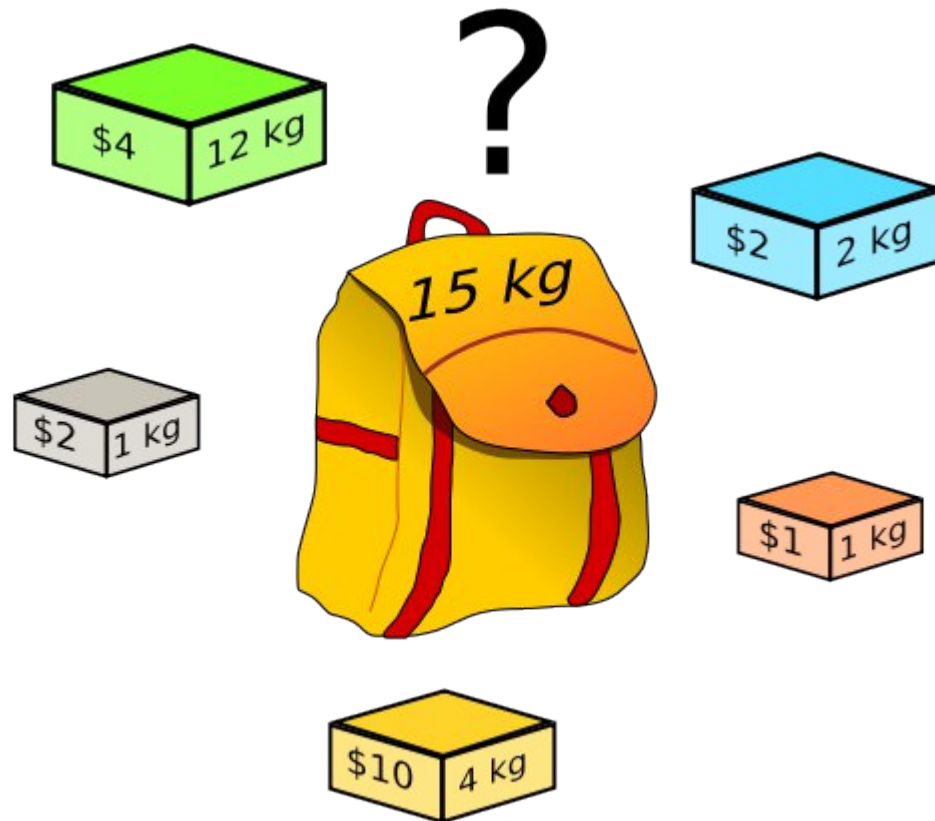
Aula 07 – Busca local

2021 – Engenharia

Igor Montagner <igorsm1@insper.edu.br>
Antônio Selvatici <antoniohps1@insper.edu.br>

Algoritmos aleatorizados

A mochila binária



<https://commons.wikimedia.org/wiki/File:Knapsack.svg>

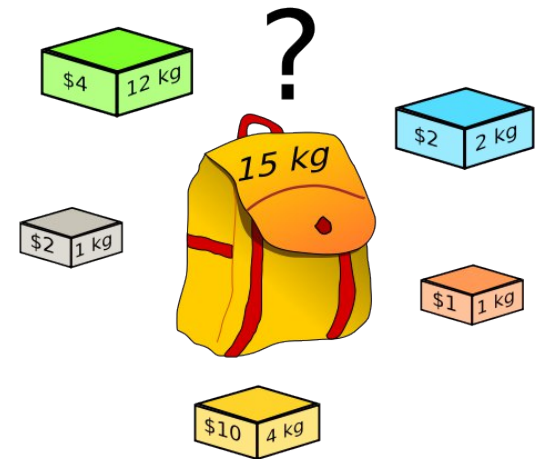
A mochila binária

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?

Algumas opções:

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

Heurística

"truque" usado para resolver um problema rapidamente

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

- Não garante resultados ótimos
- Nem resultados bons em todas situações

Exploration x Exploitation

Nossa heurística é **100% Exploitation**.

Como podemos adicionar **Exploration**?

1. Alternar heurísticas **de vez em quando**
2. **De vez em quando** faço uma escolha qualquer
3. Inverto a heurística **de vez em quando**

Números (pseudo-)aleatórios

Sorteio de números aleatórios

- 1. Gerador:** produz bits aleatórios a partir de um parâmetro **seed**. Cada **seed** gera uma sequência diferente de bits.
- 2. Distribuição de probabilidade:** gera sequência de números a partir de um conjunto de parâmetros



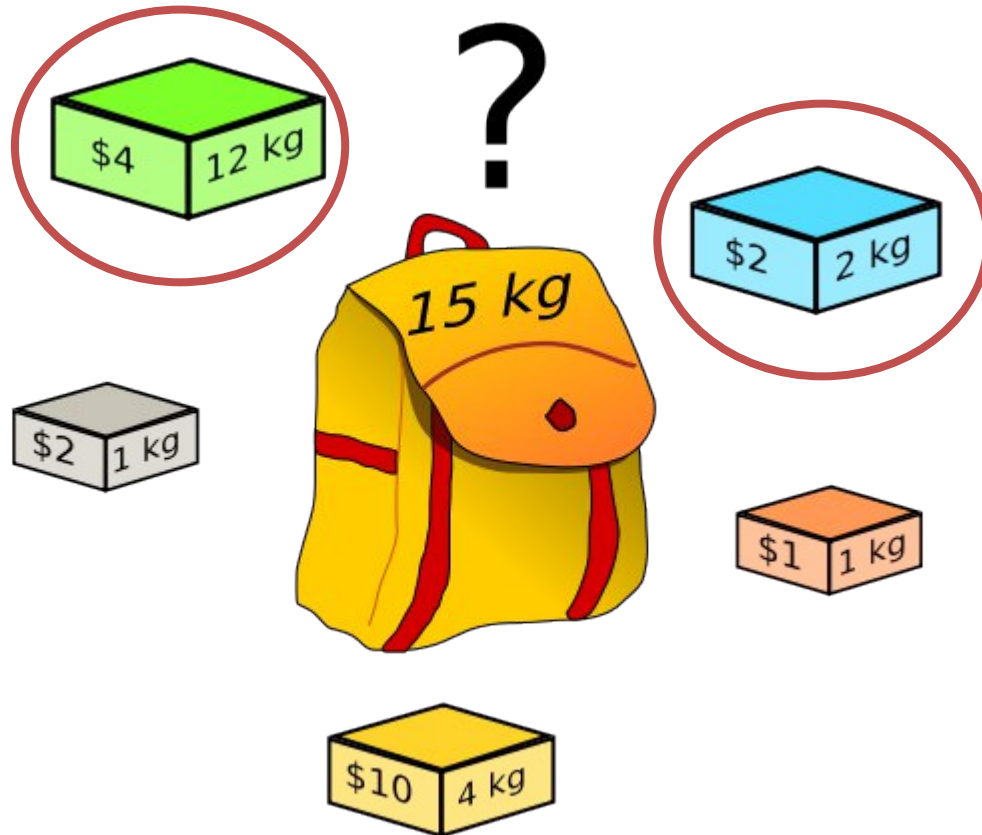
Atividade prática

E se fizermos 100% exploration? (30 minutos)

1. Revisar geração de números aleatórios
2. Comparar soluções de algoritmos aleatorizados

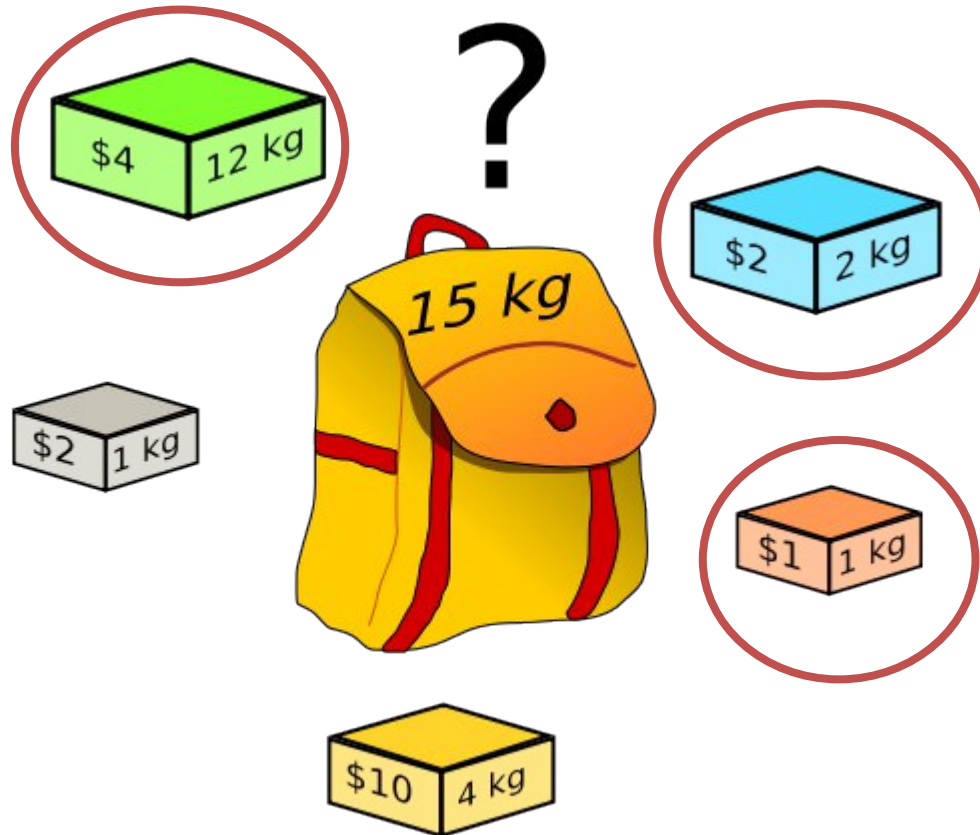
Solução aleatorizada

Uma solução para a mochila



Peso: 14kg
Valor: \$6

Uma solução para a mochila (II)



Peso: 15kg
Valor: \$7

Melhorando uma solução

- 1. Encher a mochila:** verificar se algum objeto não selecionado cabe na mochila
- 2. Trocar dois objetos:** verificar se é possível substituir um objeto selecionado por outro de melhor valor que foi deixado de fora

Solução ótima global

- 1. Encher a mochila:** não é possível
- 2. Trocar dois objetos:** não é possível

Solução ótima global

- 1. Encher a mochila:** não é possível
- 2. Trocar dois objetos:** não é possível

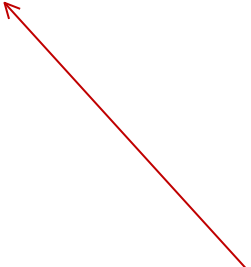
**Ambas são condições
necessárias, mas não
suficientes, para
otimalidade**

Busca local

1. Cria uma solução
2. Aplicar, sucessivamente, uma operação que melhora esta solução.
3. Parar quando não for mais possível

Busca local

1. Cria uma solução
2. Aplicar, sucessivamente, uma operação que melhora esta solução.
3. Parar quando não for mais possível



Aplicar alguma condição necessária mas não suficiente

Busca local

1. Cria uma solução
2. Aplicar, sucessivamente, uma operação que melhora esta solução.
3. Parar quando não for mais possível

**Não tem nada
melhor por perto!**

**Aplicar alguma condição
necessária mas não
suficiente**

Busca local

1. Cria uma solução
2. Aplicar, sucessivamente, uma operação que melhora esta solução.
3. Parar quando não for mais possível

Depende de onde começou!

Não tem nada melhor por perto!

Aplicar alguma condição necessária mas não suficiente

Busca local

1. Repetir N vezes:

1. Cria uma solução

2. Aplicar, sucessivamente, uma operação que melhora esta solução.

3. Parar quando não for mais possível

2. Retorne a melhor solução

Busca local (vantagens)

1. Rápida
2. Resultados bons para N grande
3. Oferece "garantia" (fraca) de qualidade
4. Não ficou bom? Roda mais vezes!

Busca local (desvantagens)

1. Depende de gerar soluções iniciais
2. Aleatorizado (pode não ser problema)
3. Oferece "garantia" (**fraca**) de qualidade

Atividade prática

Busca local e aleatoriedade

1. Criar algoritmo que seleciona a melhor de N soluções
2. Implementar "Mochila cheia"
3. Implementar "Substitui objeto"

Fechamento

As soluções ficaram melhores que as heurísticas?

E o tempo de execução?

Insper

www.insper.edu.br