

SuperComputação

Aula 5 – projeto + algoritmos aleatorizados

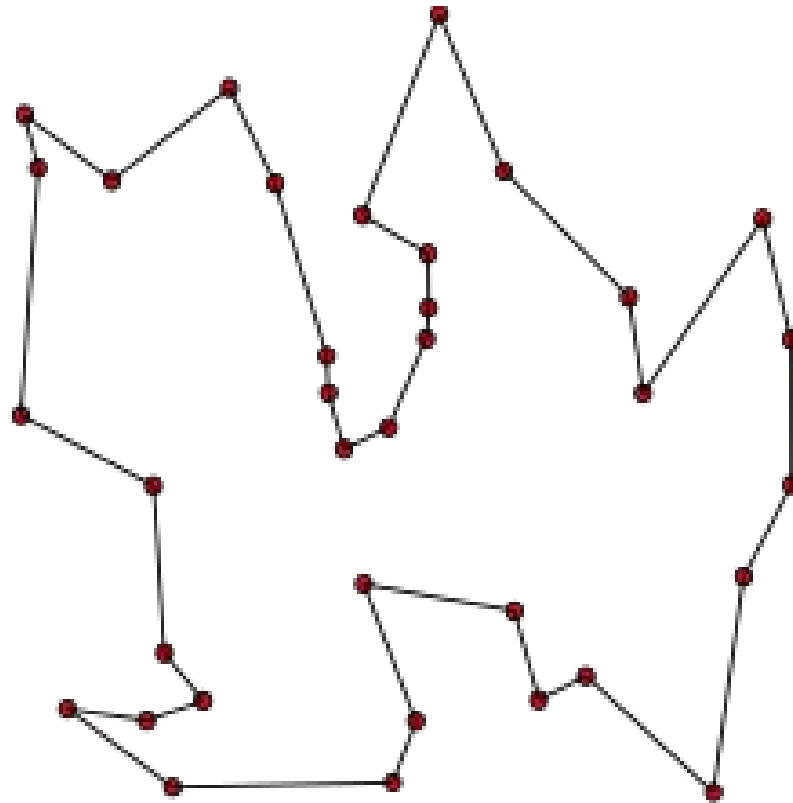
2020 – Engenharia

Luciano Soares <lpsoares@insper.edu.br>

Igor Montagner <igorsm1@insper.edu.br>

Projeto

Projeto - Travelling Salesperson (TSP)



Projeto - Travelling Salesperson (TSP)

Dadas N cidades, escolher um caminho fechado (tour) tal que

- cada cidade é visitada somente uma vez
- o caminho é o mais curto possível

Problema de otimização difícil

Projeto - Travelling Salesperson (TSP)

Teremos três grandes partes no projeto

- algoritmos
- análise de desempenho
- paralelismo

Toda parte será conectada com alguma aula, em que faremos discussões e fixaremos um prazo para os exercícios

Projeto - Travelling Salesperson (TSP)

<https://insper.github.io/supercomp/projetos/>

TSP - heurística

A primeira entrega do projeto será implementar uma heurística para resolver o TSP: **a cidade mais próxima**. A partir de uma cidade inicial iremos

Escolher sempre a cidade mais próxima não visitada

Note que isto depende da cidade inicial. Para esta entrega iremos começar sempre na primeira cidade.

TSP - heurística



1. Corretores automáticos estarão disponíveis no repositório privado
2. Entradas/saídas de exemplo serão fornecidos para ajudar a debugar suas implementações

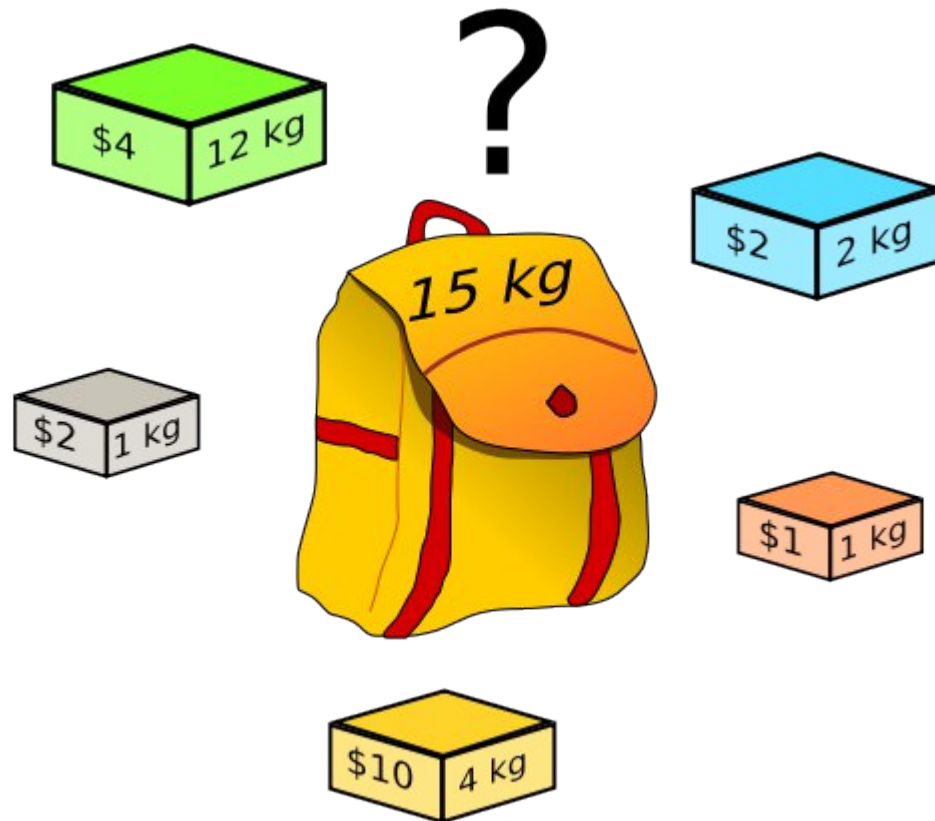


Hoje

- Algoritmos aleatorizados

Algoritmos aleatorizados

A mochila binária



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

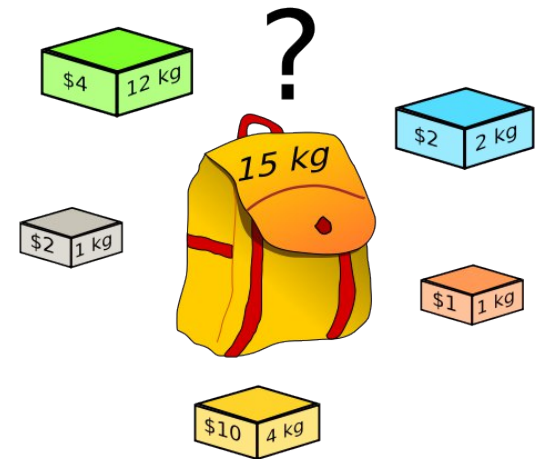
A mochila binária

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

Heurísticas - limitações

1. E se a solução gerada não for boa? Consigo "tentar" de novo e gerar outras parecidas?
2. Será que é possível melhorar a solução gerada? Como?

Exploration x Exploitation

Exploitation:

- explorar alguma propriedade do problema
- pode ser uma intuição que leve a bons resultados em curto prazo

Exploration:

- decisão não localmente ótima feita "de propósito"
- visa adicionar variabilidade nas soluções geradas

Exploration x Exploitation

Nossa heurística é **100% Exploitation**.

Como podemos adicionar **Exploration**?

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

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**

Exploration

Exploration requer a capacidade de criar um programa que execute de maneira diferente a cada execução.

Precisamos

1. de uma fonte de aleatoriedade;
2. uma maneira de gerar sequências de números aleatórios

Números aleatórios

Um gerador de números aleatórios é impossível de ser criado usando um computador:

- 1.É impossível predizer qual será o próximo número aleatório "de verdade"
- 2.Um computador executa uma sequência de comandos conhecida baseada em dados guardados na memória.
Execução é **Determinística**.

Números (pseudo-)aleatórios

Gerador de números pseudo-aleatórios (**pRNG**): algoritmo determinístico que gera sequências de números que parecem aleatórias

- 1. Determinístico:** produz sempre a mesma sequência.
- 2. Sequências que parecem aleatórias:** não conseguiríamos distinguir uma sequência gerada por um pRNG e uma sequência aleatória de verdade.

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

Aleatorizando a mochila binária (45 minutos)

1. Adicionar aleatoriedade em nossas heurísticas



Comentários sobre RNGs



Atividade prática

E se fosse tudo aleatório?

1. Criando uma solução completamente aleatória

Fechamento

Adicionar aleatoriedade melhorou os resultados?

Qual a qualidade das soluções aleatórias?

Insper

www.insper.edu.br