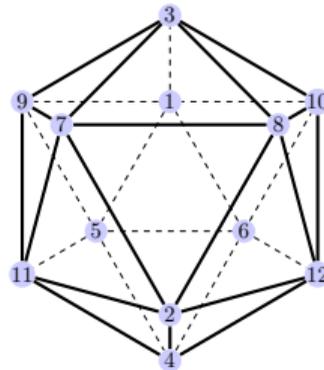


# Metaheurísticas - Lab2 Modelagem computacional

Alexandre Checoli Choueiri

11/03/2023



- ① Problemática
- ② O que deve ser definido na modelagem
- ③ Exemplo - o problema da mochila
- ④ Conclusão
- ⑤ Atividades

# Problemática

## Representação do problema

Um aspecto crucial no *design* de qualquer algoritmo é a modelagem computacional do problema. Afinal de contas teremos que representar um problema do mundo real computacionalmente, e já vimos que alguns desses problemas podem ser bem complexos. Como desdobramos o problema em estruturas de dados computacionais? Esse processo é chamado de **modelagem computacional**.

# Problemática

## Representação do problema

### Problema

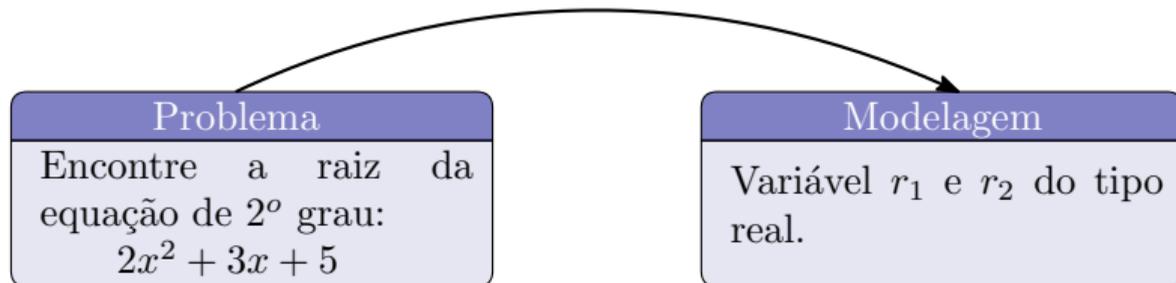
Encontre a raiz da  
equação de 2º grau:

$$2x^2 + 3x + 5$$

Para alguns problemas, a modelagem computacional é trivial.

# Problemática

## Representação do problema



Para alguns problemas, a modelagem computacional é trivial.

# Problemática

## Representação do problema

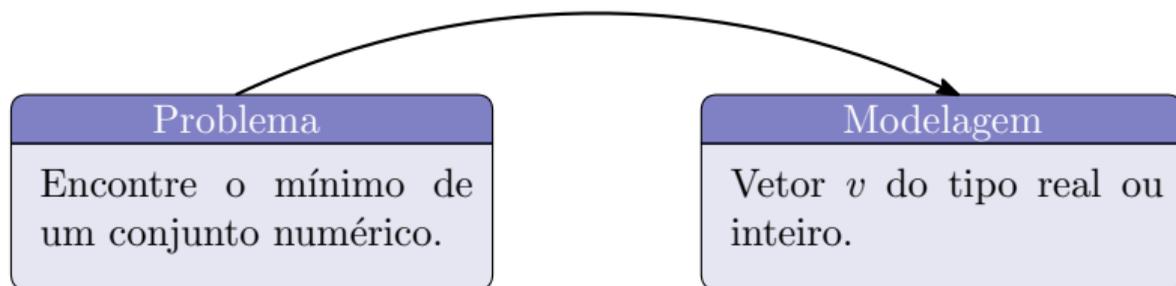
### Problema

Encontre o mínimo de um conjunto numérico.

Para alguns problemas, a modelagem computacional é trivial.

# Problemática

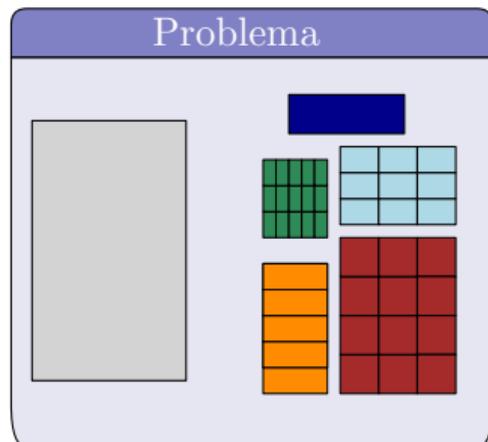
## Representação do problema



Para alguns problemas, a modelagem computacional é trivial.

# Problemática

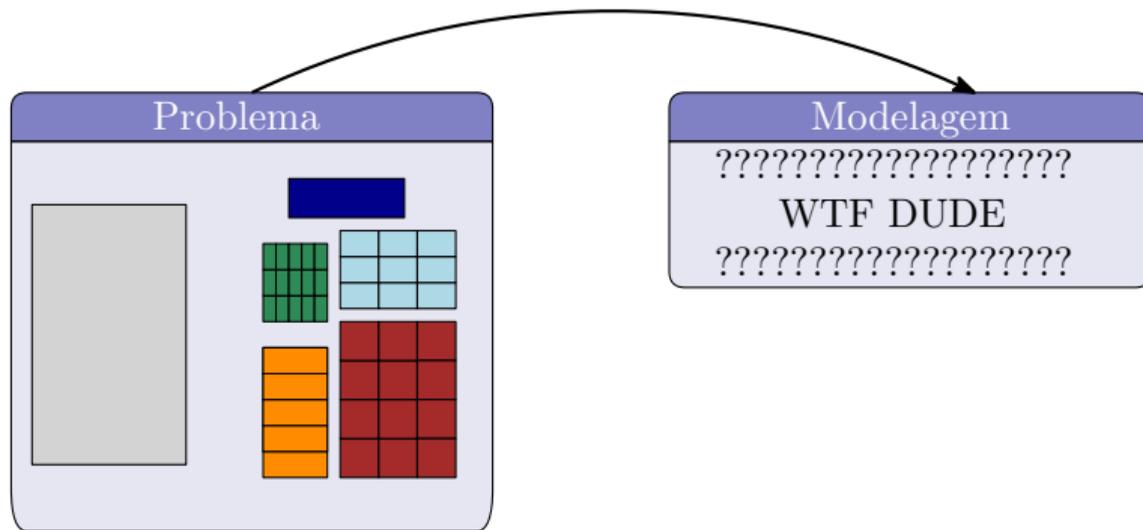
## Representação do problema



Já para outros...

# Problemática

Representação do problema



Já para outros...**nem tanto!**

# Questões da modelagem

O que deve ser definido?

Em se tratando de problemas de otimização, 4 questões de modelagem computacional devem estar totalmente resolvidas antes da implementação de qualquer algoritmo:

# Questões da modelagem

O que deve ser definido?

Em se tratando de problemas de otimização, 4 questões de modelagem computacional devem estar totalmente resolvidas antes da implementação de qualquer algoritmo:

1. Como a "instancia" do problema é definida? Ou seja, quais são os dados de entrada (parâmetros) e como eles serão imputados no programa? (de forma genérica, para qualquer tamanho de instancia).

# Questões da modelagem

O que deve ser definido?

Em se tratando de problemas de otimização, 4 questões de modelagem computacional devem estar totalmente resolvidas antes da implementação de qualquer algoritmo:

1. Como a "instancia" do problema é definida? Ou seja, quais são os dados de entrada (parâmetros) e como eles serão imputados no programa? (de forma genérica, para qualquer tamanho de instancia).
2. Quais estruturas de dados serão usadas para armazenar os parâmetros e a solução do problema?

# Questões da modelagem

O que deve ser definido?

Em se tratando de problemas de otimização, 4 questões de modelagem computacional devem estar totalmente resolvidas antes da implementação de qualquer algoritmo:

1. Como a "instancia" do problema é definida? Ou seja, quais são os dados de entrada (parâmetros) e como eles serão imputados no programa? (de forma genérica, para qualquer tamanho de instancia).
2. Quais estruturas de dados serão usadas para armazenar os parâmetros e a solução do problema?
3. Qual o procedimento para se verificar se a solução é factível ou não?

# Questões da modelagem

O que deve ser definido?

Em se tratando de problemas de otimização, 4 questões de modelagem computacional devem estar totalmente resolvidas antes da implementação de qualquer algoritmo:

1. Como a "instancia" do problema é definida? Ou seja, quais são os dados de entrada (parâmetros) e como eles serão imputados no programa? (de forma genérica, para qualquer tamanho de instancia).
2. Quais estruturas de dados serão usadas para armazenar os parâmetros e a solução do problema?
3. Qual o procedimento para se verificar se a solução é factível ou não?
4. Qual o procedimento para se verificar o custo da solução?

# Questões da modelagem

O que deve ser definido?

Um breve comentário sobre instâncias antes de continuarmos:

As chamadas *instancias* são conjuntos de dados para os diversos problemas de otimização. Por exemplo, para o TSP temos o [TSPLib](#), um repositório com muitos problemas do TSP e suas variações. **Mas por que as instancias são importantes?**

# Questões da modelagem

O que deve ser definido?

As instâncias são importantes pois toda a comunidade científica testa os seus algoritmos nelas. Dessa forma podemos comparar as nossas soluções com a de outros algoritmos. Por experiência própria com o VRP:

			BKS											
Instancia fechada			C1		C2		R1		R2		RC1		RC2	
			828,38	10	589,86	3	1210,34	11,92	951,03	2,73	1384,17	11,5	1119,24	3,25
			C1		C2		R1		R2		RC1		RC2	
Data	Tempo	OBS	distance	n_vehicles										
11/10/2021	180		957,44	11,11	639,27	3,75	1258,48	14,58	1039,58	4,00	1468,98	14,38	1239,05	4,50
12/10/2021	180		889,81	10,33	592,02	3,00	1216,16	13,75	985,68	3,73	1419,90	13,88	1260,66	4,25
22/10/2021	180	Novo 2 OPT STAR	828,94	10,00	590,69	3,00	1206,90	13,58	947,50	3,82	1391,99	13,75	1158,92	4,13
24/10/2021	180	Novo OR OPT	828,70	10,00	593,96	3,00	1216,41	13,33	962,99	3,55	1391,38	13,50	1116,48	4,25
24/10/2021	300		828,70	10,00	592,07	3,00	1218,23	13,17	958,99	3,36	1391,85	13,38	1132,569	4,13
19/11/2021	300	500 rotas no sp	828,38	10,00	589,8581	3,00	1203,78	12,67	970,73	3,27	1394,571	12,75	1140,471	3,75
20/11/2021	300	1000 rotas no sp	828,38	10,00	589,8581	3	1189,249	12,5	960,216	3,36	1381,915	12,75	1136,733	3,88
20/11/2021	300	500 rotas, com filtro	828,38	10,00	589,8581	3	1196,33	12,83	-	-	1389,194	12,88	1170,684	3,75
21/11/2021	600	2000 sem filtro	828,38	10,00	589,8581	3	1192,96	12,42	968,673	3,18	1369,104	12,5	1131,291	3,75

# Questões da modelagem

## O problema da mochila

Como essas questões seriam respondidas considerando o **problema da mochila**?

1. Como a "instancia" do problema é definida? Ou seja, quais são os dados de entrada (parâmetros) e como eles serão imputados no programa? (de forma genérica, para qualquer tamanho de instancia).
2. Quais estruturas de dados serão usadas para armazenar os parâmetros e a solução do problema?
3. Qual o procedimento para se verificar se a solução é factível ou não?
4. Qual o procedimento para se verificar o custo da solução?

# Questões da modelagem

## O problema da mochila

Como essas questões seriam respondidas considerando o **problema da mochila**?

1. Como a "instancia" do problema é definida? Ou seja, quais são os dados de entrada (parâmetros) e como eles serão imputados no programa? (de forma genérica, para qualquer tamanho de instancia).
2. Quais estruturas de dados serão usadas para armazenar os parâmetros e a solução do problema?
3. Qual o procedimento para se verificar se a solução é factível ou não?
4. Qual o procedimento para se verificar o custo da solução?

# Questões da modelagem

## O problema da mochila

$N$	$C$
$c_1$	$p_1$
$c_2$	$p_2$
$\vdots$	$\vdots$
$\vdots$	$\vdots$
$c_n$	$p_n$

Podemos usar um arquivo de texto, como mostrado na Figura, em que:

$\left\{ \begin{array}{l} N : \text{Numero de itens} \\ C : \text{Capacidade da mochila} \\ c_i : \text{Custo do item } i \\ p_i : \text{Peso do item } i \end{array} \right.$

# Questões da modelagem

## O problema da mochila

Como essas questões seriam respondidas considerando o **problema da mochila**?

1. Como a "instancia" do problema é definida? Ou seja, quais são os dados de entrada (parâmetros) e como eles serão imputados no programa? (de forma genérica, para qualquer tamanho de instancia).
2. Quais estruturas de dados serão usadas para armazenar os parâmetros e a solução do problema?
3. Qual o procedimento para se verificar se a solução é factível ou não?
4. Qual o procedimento para se verificar o custo da solução?

# Questões da modelagem

## O problema da mochila

Podemos usar as seguintes estruturas para armazenar os dados de entrada:

$$\left\{ \begin{array}{l} N : \text{Número inteiro com a quantidade de itens.} \\ C : \text{Número inteiro com a capacidade da mochila.} \\ v_c : \text{Vetor de inteiros com os custos.} \\ v_p : \text{Vetor de inteiros com os pesos.} \end{array} \right.$$

# Questões da modelagem

## O problema da mochila

E a solução pode ser representada por um vetor  $v\_sol$  de booleanos (ou inteiros), em que:

$$v\_sol[i] = \begin{cases} 0 \text{ (False)} : \text{O item } i \text{ não é levado na mochila.} \\ 1 \text{ (True)} : \text{O item } i \text{ é levado na mochila.} \end{cases}$$

# Questões da modelagem

## O problema da mochila

**PERGUNTA:** Poderíamos representar a solução como uma vetor  $v_{sol}$  de strings, como mostrado abaixo?

$$v_{sol}[i] = \begin{cases} (\text{NÃO}) : \text{O item } i \text{ não é levado na mochila.} \\ (\text{SIM}) : \text{O item } i \text{ é levado na mochila.} \end{cases}$$

Qual das duas opções você acha que seria melhor?

# Questões da modelagem

## O problema da mochila

Como essas questões seriam respondidas considerando o **problema da mochila**?

1. Como a "instancia" do problema é definida? Ou seja, quais são os dados de entrada (parâmetros) e como eles serão imputados no programa? (de forma genérica, para qualquer tamanho de instancia).
2. Quais estruturas de dados serão usadas para armazenar os parâmetros e a solução do problema?
3. Qual o procedimento para se verificar se a solução é factível ou não?
4. Qual o procedimento para se verificar o custo da solução?

# Questões da modelagem

## O problema da mochila

```
Function verifica_fact_mochila(v_sol() As Integer, v_p() As Integer, C As Integer) As Boolean
    Dim i As Integer
    Dim peso As Integer

    i = 0
    peso = 0
    ' Vai somando os pesos dos itens enquanto duas condicoes sejam satisfeita:
    ' o peso nao excede a capacidade E todos os itens ão aj foram somados
    Do While peso <= C And i <= UBound(v_sol)
        peso = peso + v_sol(i) * v_p(i)
        i = i + 1
    Loop

    If peso > C Then
        verifica_fact_mochila = False
    Else
        verifica_fact_mochila = True
    End If
End Function
```

# Questões da modelagem

O problema da mochila

O código anterior pode ser testado com a subrotina:

```
Sub teste_fact()  
  Dim v_p(2) As Integer  
  Dim v_sol(2) As Integer  
  Dim C As Integer  
  
  C = 5  
  v_p(0) = 5  
  v_p(1) = 10  
  v_p(2) = 5  
  
  v_sol(0) = 1  
  v_sol(1) = 0  
  v_sol(2) = 1  
  
  Dim fact As Boolean  
  fact = verifica_fact_mochila(v_sol, v_p, C)  
  Debug.Print "Solucao factivel? " & fact  
End Sub
```

# Questões da modelagem

## O problema da mochila

Como essas questões seriam respondidas considerando o **problema da mochila**?

1. Como a "instancia" do problema é definida? Ou seja, quais são os dados de entrada (parâmetros) e como eles serão imputados no programa? (de forma genérica, para qualquer tamanho de instancia).
2. Quais estruturas de dados serão usadas para armazenar os parâmetros e a solução do problema?
3. Qual o procedimento para se verificar se a solução é factível ou não?
4. Qual o procedimento para se verificar o custo da solução?

# Questões da modelagem

## O problema da mochila

Basta somarmos as utilidades (ou custos) dos itens que estão atualmente na mochila.

```
Function calcula_custo(v_sol() As Integer, v_c()) As Integer

    Dim i As Integer
    Dim custo As Integer

    i = 0
    custo = 0

    For i = 0 To UBound(v_sol)
        custo = custo + v_sol(i) * v_c(i)
    Next i
    calcula_custo = custo

End Function
```

# Questões da modelagem

## O problema da mochila

Agora, sabendo de todas as funções usadas para se verificar a factibilidade e o custo com o vetor de inteiros, o que você tem a dizer sobre a representação da solução como um vetor de strings?

$$v_{sol}[i] = \begin{cases} \text{NÃO} : \text{O item } i \text{ não é levado na mochila.} \\ \text{SIM} : \text{O item } i \text{ é levado na mochila.} \end{cases}$$

# Questões da modelagem

## O problema da mochila

Note que nas duas funções (verificar factibilidade e calcular custo) **aproveitamos a estrutura de dados da solução** para realizar uma multiplicação dos valores pelos custos e pelos pesos.

```
For i = 0 To UBound(v_sol)
    custo = custo + v_sol(i) * v_c(i)
Next i
```

```
Do While peso <= C And i <= UBound(v_sol)
    peso = peso + v_sol(i) * v_p(i)
    i = i + 1
Loop
```

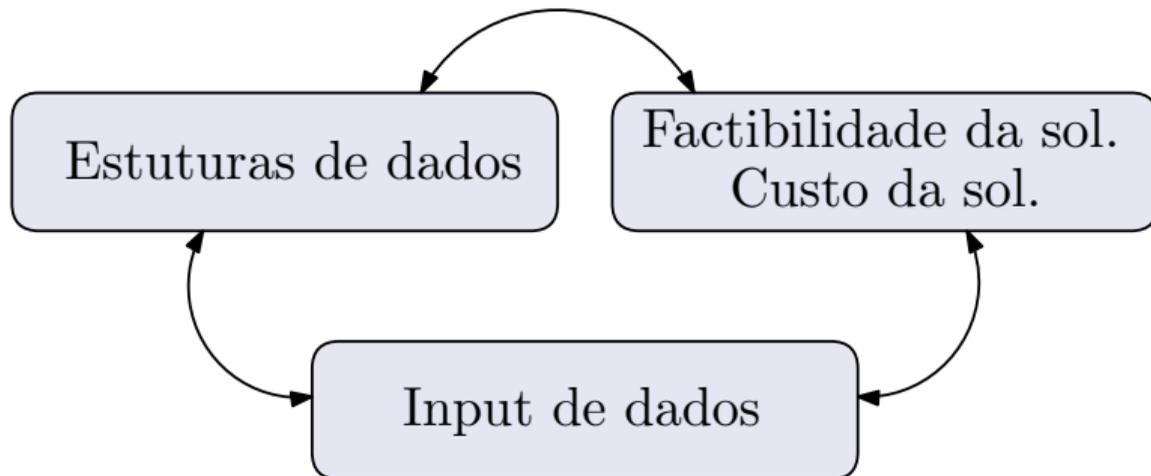
Não poderíamos fazer o calculo dessa forma se o vetor `v_sol` fosse de string. Seria necessário colocar pelo menos uma condicional em cada loop.

# Questões da modelagem

## O problema da mochila

### Conclusão

Toda a etapa de modelagem computacional deve ser feita em conjunto, não de forma sequencial: as estruturas de dados limitam a forma de se realizar os cálculos, e a definição de como calcular pode também restringir as estruturas de dados utilizadas.



## Atividade I - em sala

- 1 O problema da mochila também possui instancias de treino. Algumas são fornecidas [neste repositório](#) (também estão disponíveis na [página da disciplina](#) ). Desenvolva o código para ler os dados das instancias e os armazene nas ED definidas.
- 2 Desenvolva um código para ler as instancias do TSP que estão no [página da disciplina](#). Tanto as de coordenadas quanto as de matrizes (as instancias são do [TSPLib](#)).

## Atividade II

- 1 Responda as 4 questões de modelagem computacional para cada um dos seguintes problemas (não precisa estar implementado):
  - 2.1 O problema do caixeiro viajante (TSP).
  - 2.2 O problema de roteirização de veículos (VRP)
  - 2.3 O problema de localização de facilidades.
  - 2.4 O problema de designação.
  - 2.5 O problema de sequenciamento de produção com uma máquina.
  - 2.6 **DESAFIO:** O problema de carregamento 2D.

## Atividade III

- 1 Com base em tudo o que foi visto na aula e nos exemplos, escolha um problema de otimização para trabalhar durante o resto da disciplina. A resolução deste problema usando as técnicas desenvolvidas será uma parte da nota da disciplina. O trabalho pode ser feito em grupos de até 3 pessoas (sem problemas se for individual também). Ao longo das aulas, de acordo com os problemas escolhidos, eu vou direcionando o que deverá ser feito para a apresentação final. Escreva todas as questões de implementação para o problema escolhido:

- 1.1 Estruturas de dados (parâmetros e solução).

- 1.2 Leitura de instâncias (verifique se já existe algum repositório online disponível).

- 1.3 Cálculo de factibilidade/função objetivo

**Escreva um relatório com essas definições.**

- 2 Considerando o seu problema, crie o código para ler as instancias do problema e armazenar nas estruturas de dados.