



1920 | 2020

# Satyrus III

**Pedro Maciel Xavier**

sob orientação de Priscila Machado Vieira Lima

**UNIVERSIDADE FEDERAL DO RIO DE JANEIRO  
PESC/COPPE/UFRJ**



Foto : Minerva

# Sumário

## 1 Fundamentos

## 2 O Compilador

- Mapeamento
- Penalidades
- Restrições

## 3 Exemplos de Modelagem

- Coloração de Grafos

- TSP

- Caixeiro Pintor

## 4 Instruções

- Instalação

- Uso

## 5 Conclusão



# Fundamentos



# O Compilador

# Mapeamento

$$H(F) = 0$$

$$H(V) = 1$$

$$H(\neg p) = 1 - H(p)$$

$$H(p \wedge q) = H(p) H(q)$$

$$H(p \vee q) = H(p) + H(q) - H(p) H(q)$$

# Penalidades

No contexto do Satyrus, temos uma equação de energia a minimizar dada por

$$\begin{aligned}\mathbb{E} &= \mathbb{E}_{\text{opt}} + \mathbb{E}_{\text{int}} \\ &= \sum_i H(\varphi_i) + \sum_j \lambda_j H(\neg \varphi_j)\end{aligned}$$

onde  $\lambda_j$  é a penalidade associada à  $j$ -ésima restrição de integridade e  $H(\cdot)$  é o mapeamento. Portanto, é possível escrever um problema modelado pelo Satyrus como

$$\begin{aligned}&\text{minimizar } f(\mathbf{x}) + \boldsymbol{\lambda} \cdot g(\mathbf{x}) \\ &\text{sujeito a } \mathbf{x} \in \{0, 1\}^n\end{aligned}$$

# Penalidades

Resta saber se é possível escrever  $f(\mathbf{x}) + \lambda \cdot g(\mathbf{x}) = \mathbf{x}^T \mathbf{Q} \mathbf{x}$ . Como, por construção, tanto  $f(\mathbf{x})$  quanto  $g(\mathbf{x})$  são polinômios nas componentes de  $\mathbf{x}$ , isso pode ser feito ao aplicar uma redução dos termos com três ou mais variáveis.



# Restrições

## 1 Integridade

```
1 (int) constraint_X[1]:  
2   forall {i = [1:n]}  
3   exists {j = [1:n]}  
4   unique {k = [1:n]}  
5   (x[i] & y[j]) -> z[k];
```

## 2 Otimalidade

```
1 (opt) constraint_Y: exists {i = [1:n]} c[i];
```

# **Exemplos de Modelagem**

# Coloração de Grafos

# TSP

# Caixeiro Pintor

# Instalação

## 1 Através do *Python Package Index* (PyPI)

```
$ pip install satyrus  
$ satyrus -help  
$ python -m satyrus -help
```

## 2 Por meio do código-fonte

```
$ git clone https://github.com/pedromxavier/Satyrus3  
$ cd Satyrus3  
$ python setup.py install
```


# Uso

```
$ satyrus problem.sat -o {text, csv, gurobi, dwave, ...}
```


# Conclusão



# Referências

 **Fred Glover, Gary Kochenberger, Yu Du**  
*Quantum Bridge Analytics I: A Tutorial on Formulating and Using QUBO Models.*

 **D-Wave Systems**  
*Problem-Solving Handbook*  
[https://docs.dwavesys.com/docs/latest/c\\_handbook\\_3.html](https://docs.dwavesys.com/docs/latest/c_handbook_3.html)

 **Endre Boros, Peter L. Hammer**  
*Pseudo-Boolean optimization*  
Elsevier, Discrete Applied Mathematics, 2002.