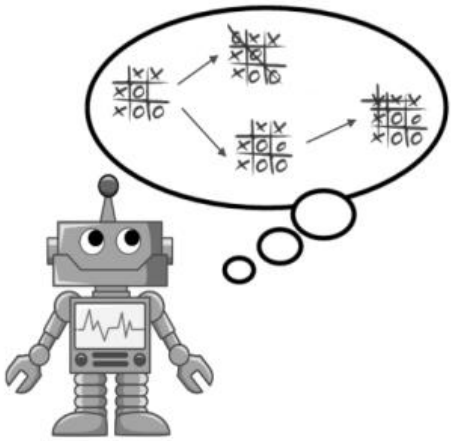


Uma breve introdução a Inteligência Artificial para Programadores

Fabio Galuppo, M.Sc.

http://fabiogaluppo.com e http://simplycpp.com/

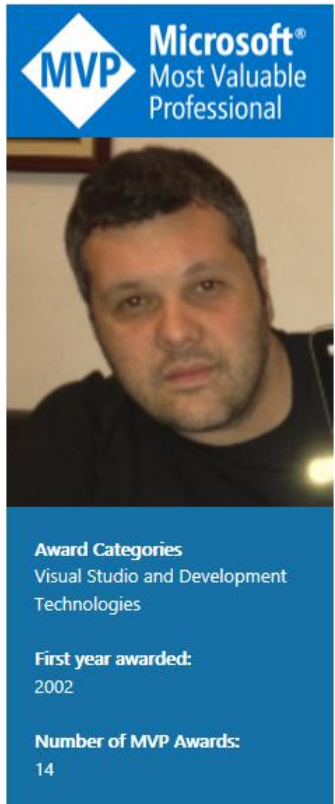
fabiogaluppo@acm.org

@FabioGaluppo

Microsoft MVP Visual Studio and Development Technologies

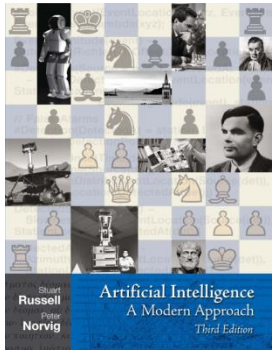
https://mvp.microsoft.com/en-us/PublicProfile/9529

http://bit.ly/mvpconf2018



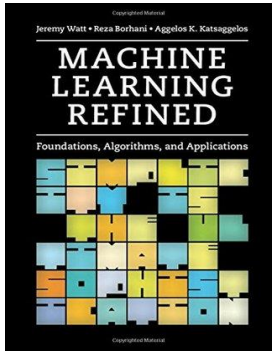
O que é Inteligência Artificial?
O que é *Machine Learning*?

- AI: ***Systems that act rationally***
  - that which is expected to ***maximize goal achievement***, given the available information



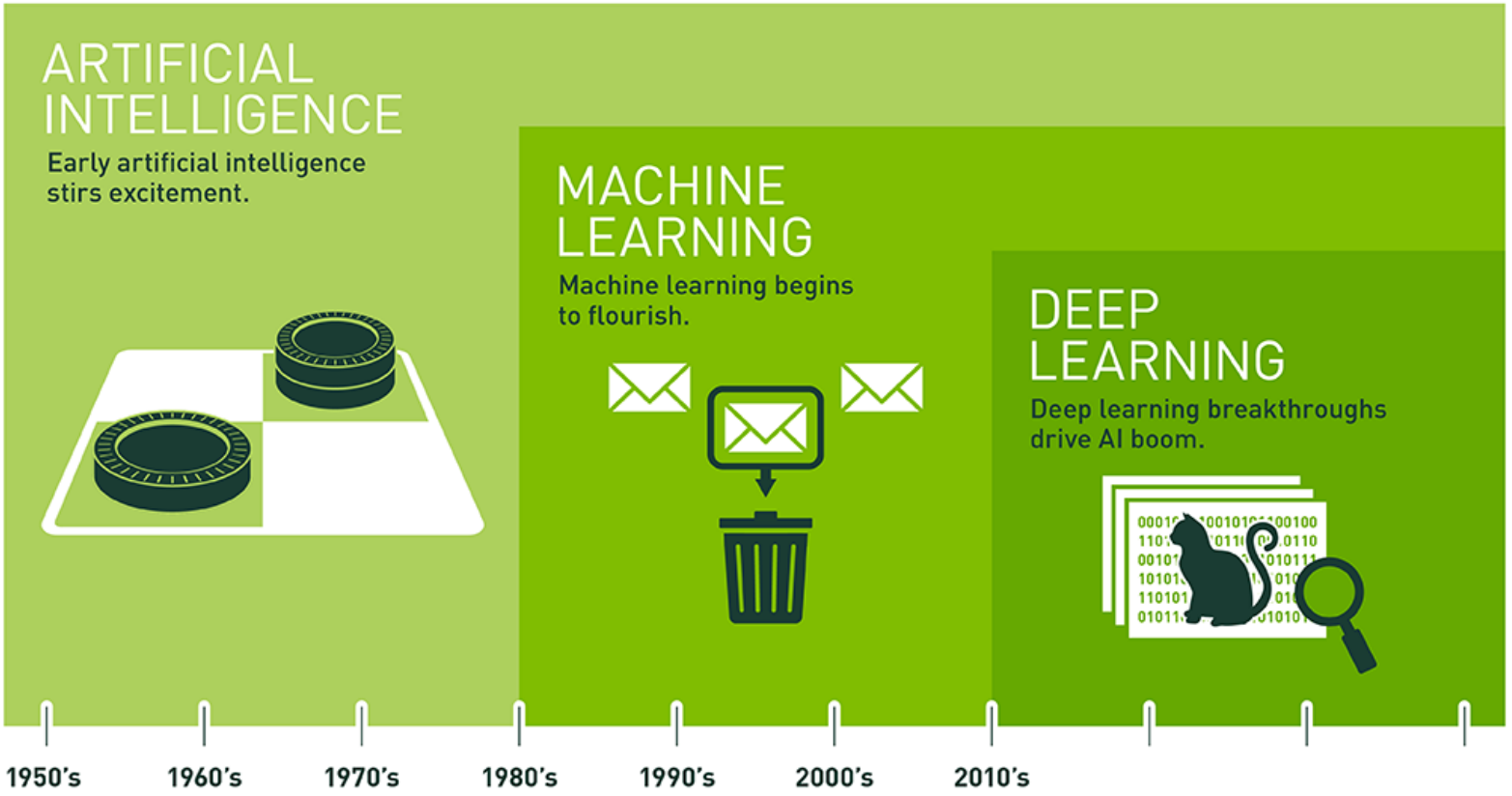
Artificial Intelligence: A Modern Approach

- ***Machine learning*** is a rapidly growing field of study whose primary concern is the ***design and analysis of algorithms*** which ***enable computers to learn***.



Machine Learning Refined

AI, *ML*, and *Deep Learning*

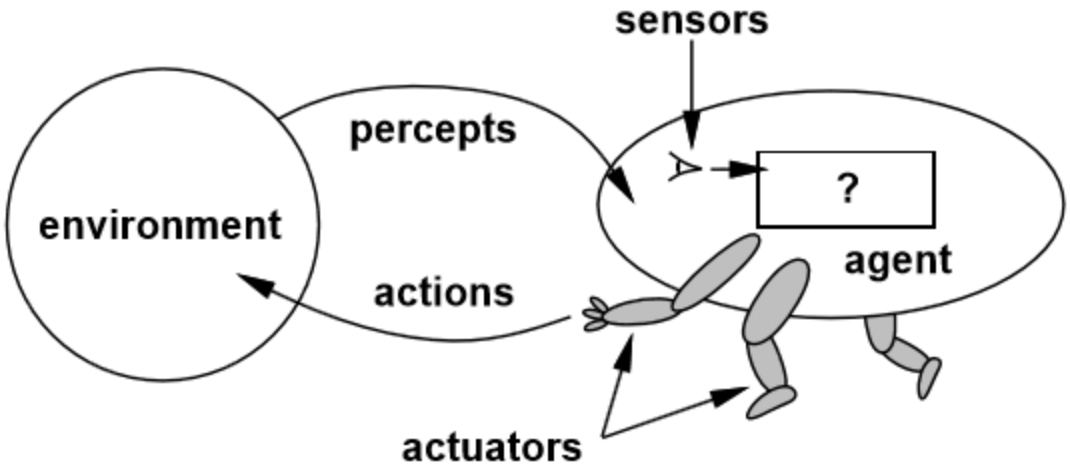


Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/

Intelligent Agents

- ***Agents*** interact with environments through ***actuators*** and ***sensors***



  - The ***agent function*** describes what the agent does in all circumstances

```
function ReactiveAgent(State) : Action
  perception ← perception(state)
  action ← action(perception)
  return action
```

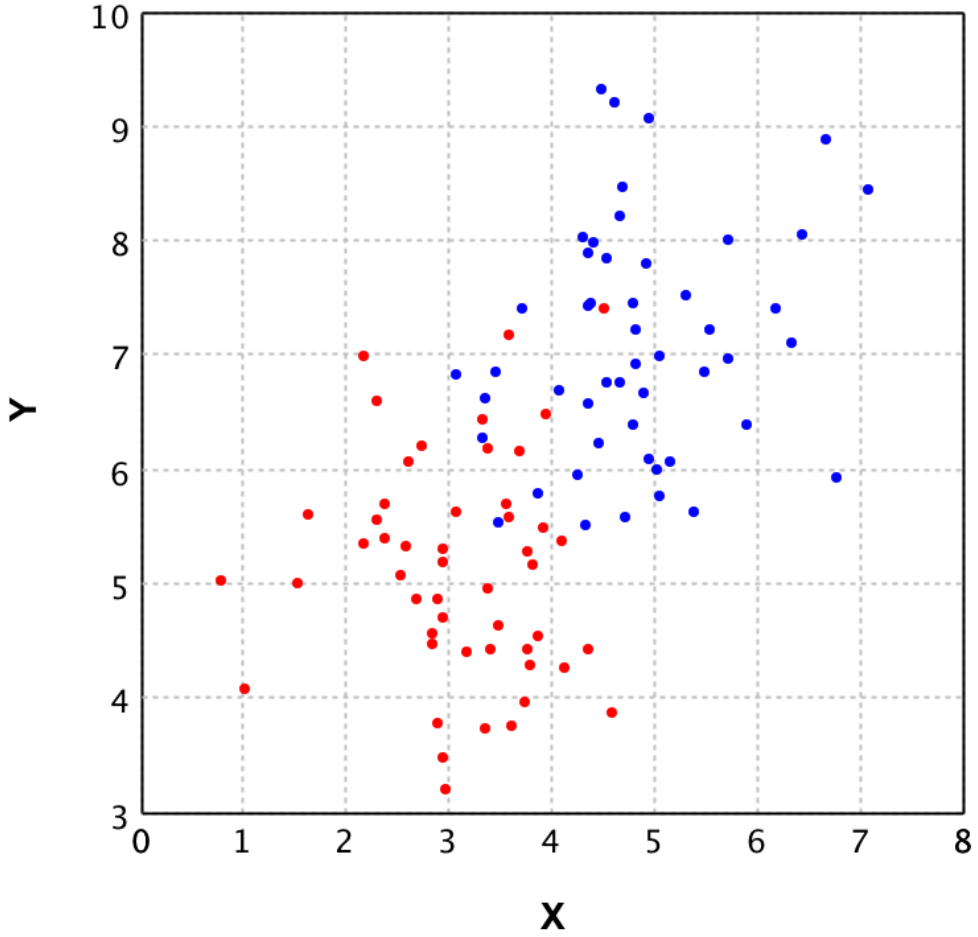
http://aima.eecs.berkeley.edu/slides-pdf/chapter02.pdf

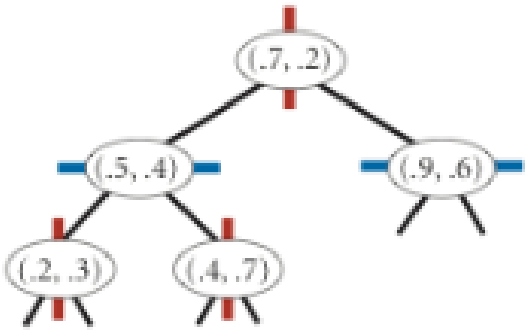
Tipos de algoritmos de *Machine Learning*

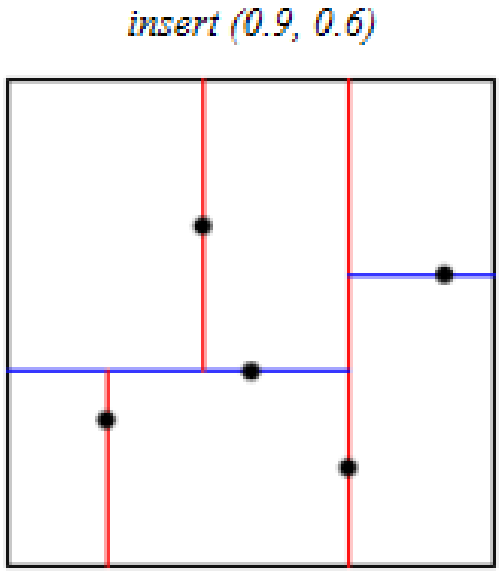
- ***Supervised Learning***
  - algorithms make predictions based on a set of examples
    - ***Classification***
    - ***Regression***
    - ***Forecasting***
- ***Unsupervised Learning***
  - It is asked to discover the intrinsic patterns that underlies the data
    - ***Clustering***
    - ***Dimension reduction***

https://blogs.sas.com/content/subconsciousmusings/2017/04/12/machine-learning-algorithm-use/

KdTree e Classificação







http://coursera.cs.princeton.edu/algs4/assignments/kdtree.html

https://xyclade.github.io/MachineLearning/#labeling-isps-based-on-their-downupload-speed-k-nn-using-smile-in-scala

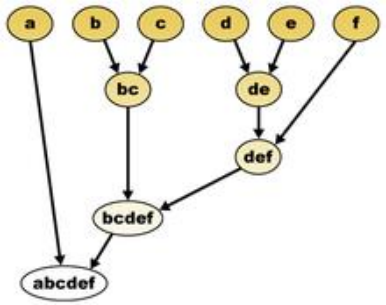
Dendrograma e Clusterização

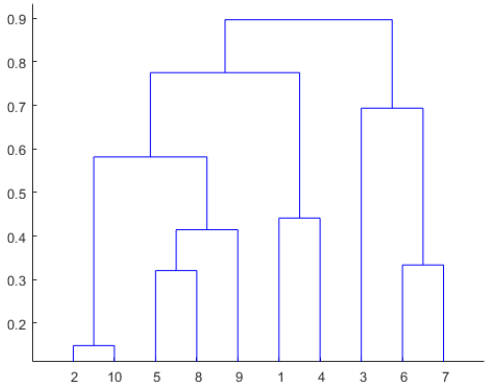
```
let book1 = { Author = list2sortedset(["Donald E. Knuth"]); Title = list2sortedset(["The Art of Computer Programming: Volume 1"]) };;
let book2 = { Author = list2sortedset(["Donald E. Knuth"]); Title = list2sortedset(["The Art of Computer Programming: Volume 2"]) };;
let book3 = { Author = list2sortedset(["Donald E. Knuth"]); Title = list2sortedset(["The Art of Computer Programming: Volume 3"]) };;
let book4 = { Author = list2sortedset(["Donald E. Knuth"]); Title = list2sortedset(["Concrete Mathematics: A Foundation for Computer Science"]) };;
let book5 = { Author = list2sortedset(["Ronald L. Graham"]); Title = list2sortedset(["Concrete Mathematics: A Foundation for Computer Science"]) };;
let book6 = { Author = list2sortedset(["Oren Patashnik"]); Title = list2sortedset(["Concrete Mathematics: A Foundation for Computer Science"]) };;
let book7 = { Author = list2sortedset(["Bjarne Stroustrup"]); Title = list2sortedset(["A Tour of C++"]) };;
let book8 = { Author = list2sortedset(["Bjarne Stroustrup"]); Title = list2sortedset(["The C++ Programming Language"]) };;

mergeTopDown (list2mutablelist([book1; book2; book3; book4; book5; book6; book7; book8])) mergeableBooks mergeBooks;;
```

1: {Donald E. Knuth, Oren Patashnik, Ronald L. Graham} {Concrete Mathematics: A Foundation for Computer Science}
2: {Donald E. Knuth} {The Art of Computer Programming: Volume 1, The Art of Computer Programming: Volume 2, The Art of Computer Programming: Volume 3}
3: {Bjarne Stroustrup} {A Tour of C++, The C++ Programming Language}

```
mergeTopDown (list2mutablelist([book4; book5; book6; book1; book2; book3; book7; book8])) mergeableBooks mergeBooks;;
```

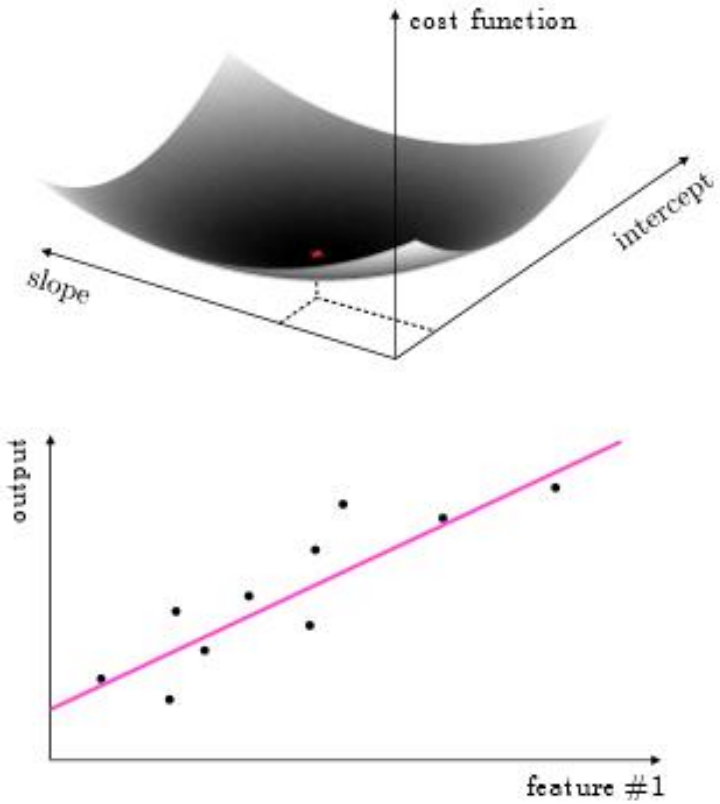
1: {Donald E. Knuth} {Concrete Mathematics: A Foundation for Computer Science, The Art of Computer Programming: Volume 1, The Art of Computer Programming: Volume 2, The Art of Computer Programming: Volume 3}
2: {Oren Patashnik, Ronald L. Graham} {Concrete Mathematics: A Foundation for Computer Science}
3: {Bjarne Stroustrup} {A Tour of C++, The C++ Programming Language}



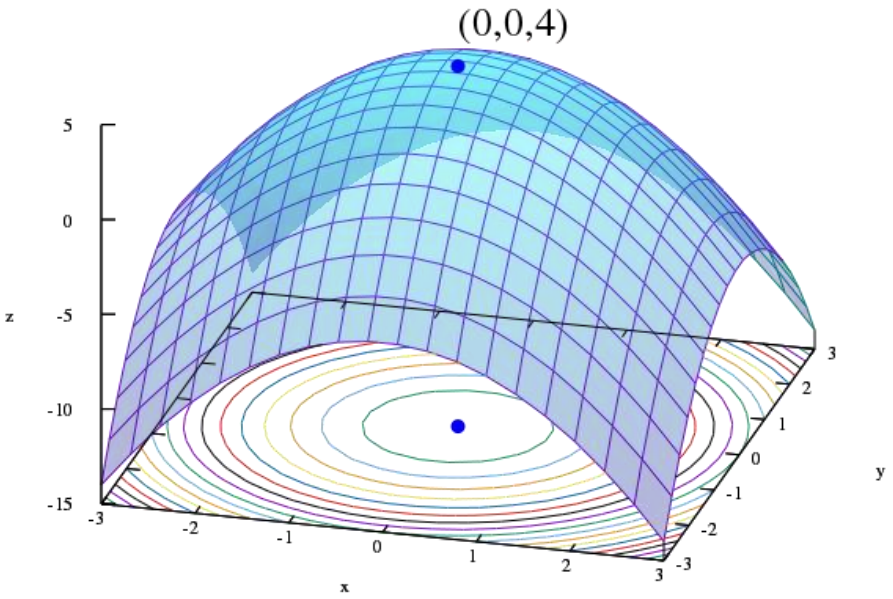


https://pt.wikipedia.org/wiki/Dendrograma

Optimization



minimum of the cost function

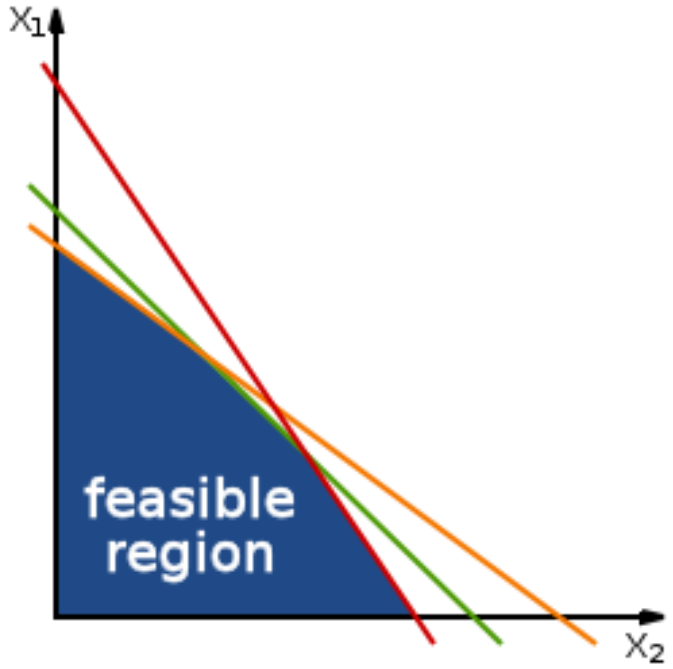


maximum of the paraboloid function

https://en.wikipedia.org/wiki/Mathematical_optimization

Optimization (Linear Programming)

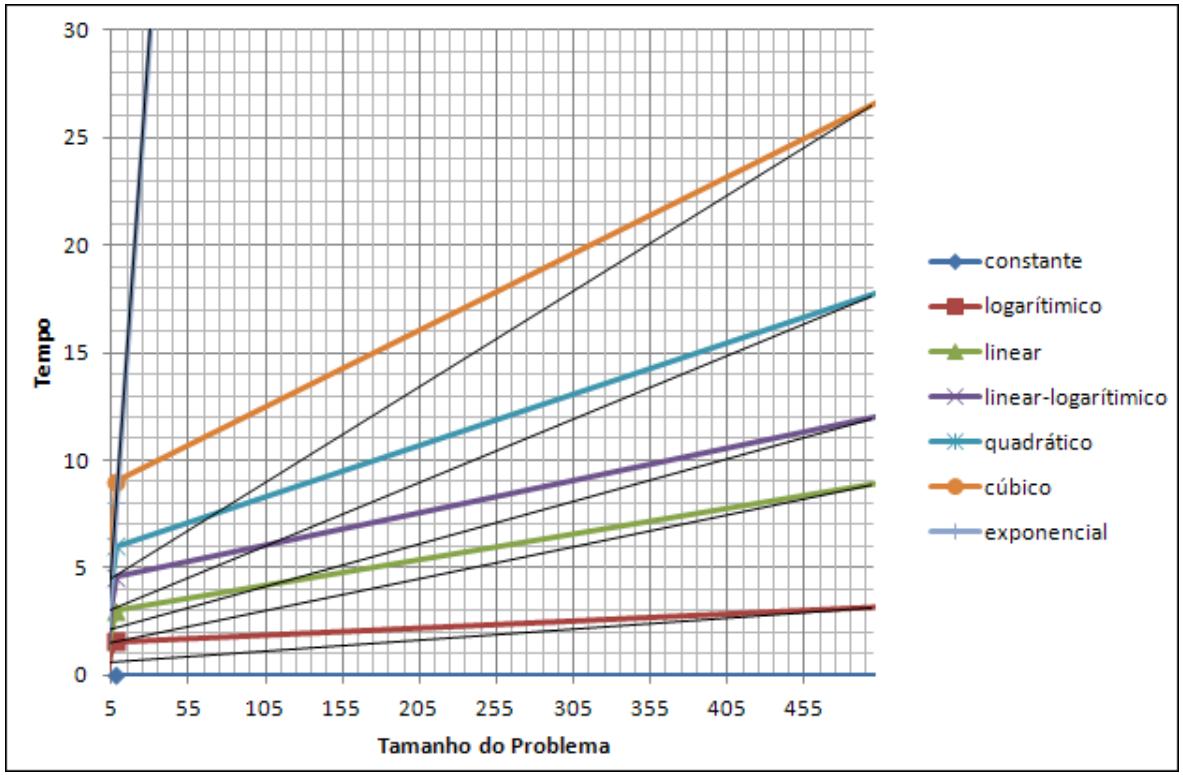
- Linear programming is a method to achieve the ***best outcome*** (such as ***maximum*** profit or ***lowest*** cost) in a mathematical model whose requirements are represented by linear relationships.
  - linear programming is a technique for the ***optimization of a linear objective function***, ***subject to linear equality and linear inequality constraints***.



https://en.wikipedia.org/wiki/Linear_programming

Intractability

- Um problema é considerado intratável quando não existe um algoritmo conhecido que o resolva deterministicamente em tempo polinomial.



- Este tipo de problema é denominado NP.
  - Aquele que possui tempo polinomial não determinístico.

O Problema do Caixeiro Viajante (PCV)

- Problema de Minimização
  - Encontrar o menor ciclo para um conjunto de cidades a serem visitadas obrigatoriamente e retornando a origem
  - Uma de suas instâncias considera a função objetivo como a distância euclidiana entre as cidades
    - Problema NP, inerentemente intratável
- Intratabilidade
  - Simétrica = $\frac{\Gamma(N)}{2}$
  - Assimétrica = $\Gamma(N)$

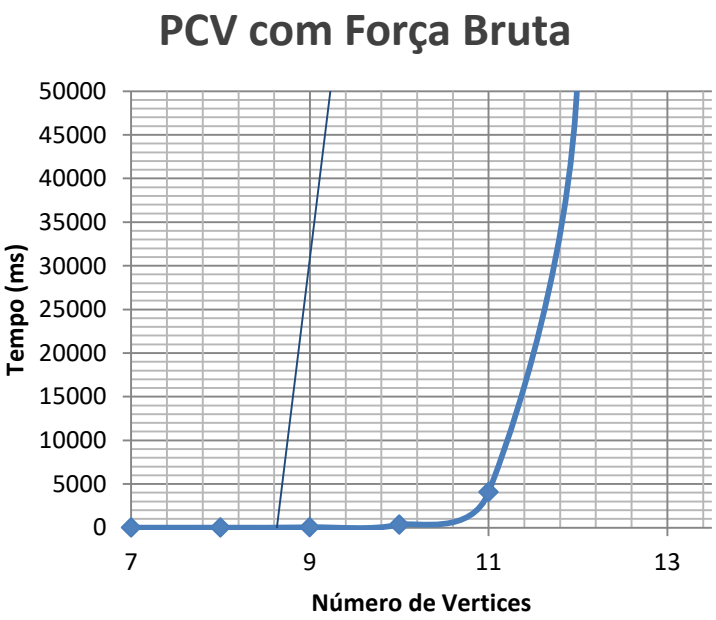
```
In[17]:= Gamma[48] / 2

Out[17]= 129 311 620 755 584 090 321 482 177 576 805 989 984 598 816 194 560 000 000 000

In[18]:= Gamma[48]

Out[18]= 258 623 241 511 168 180 642 964 355 153 611 979 969 197 632 389 120 000 000 000

In[19]:= Gamma[49]

Out[19]= 12 413 915 592 536 072 670 862 289 047 373 375 038 521 486 354 677 760 000 000 000
```

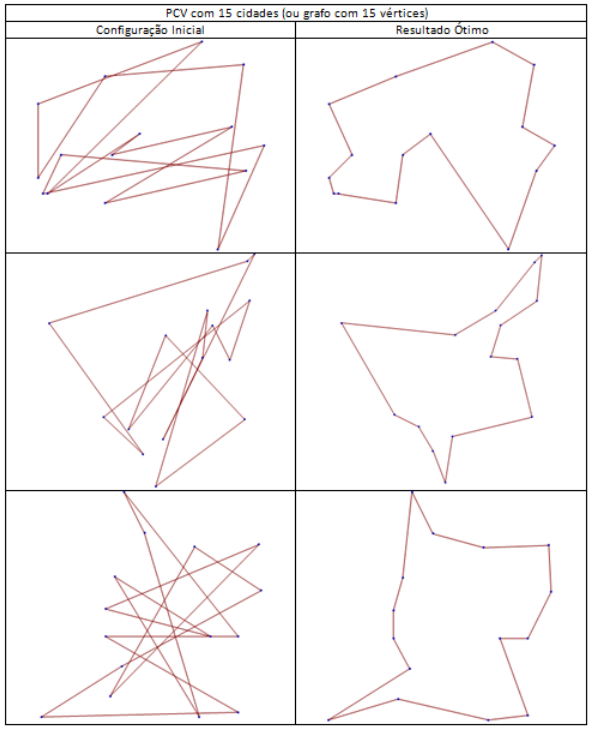
O Problema do Caixeiro Viajante (PCV)

Complexidade

	Tempo (ms)
Número de Cidades	Força Bruta
13	743691
12	53093
11	4056
10	331
9	39
8	2
7	1

PCV com 15 Cidades		
	Força Bruta	
Solução Ótima	Tempo (ms)	Tempo (h)
359,399	165340592	45,928
317,232	165590540	45,997
368,79	165517424	45,977

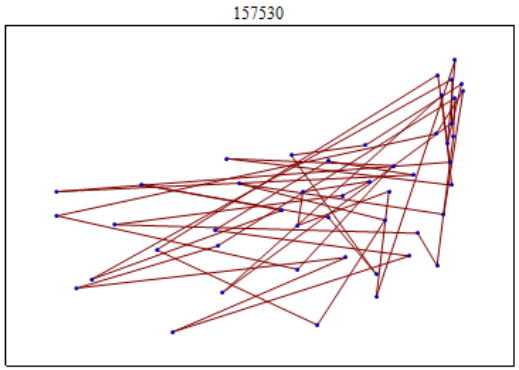




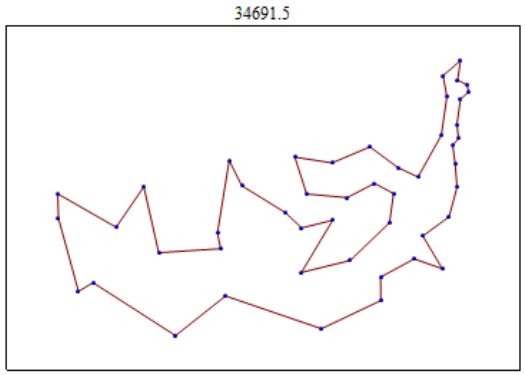
$47!/2 = 1{,}29311 \times 10^{59}$ $47! = 2{,}58623 \times 10^{59}$ $48! = 1{,}24139 \times 10^{60}$

Solução com composição de algoritmos em paralelo

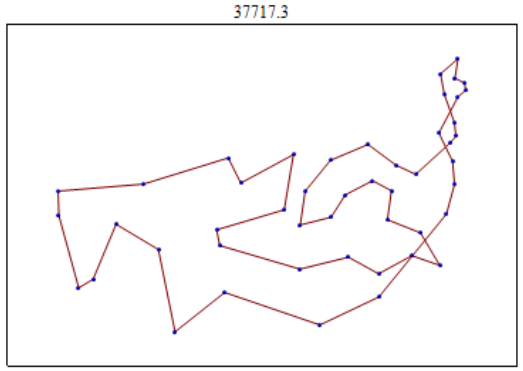
att48.tsp (48 cidades)



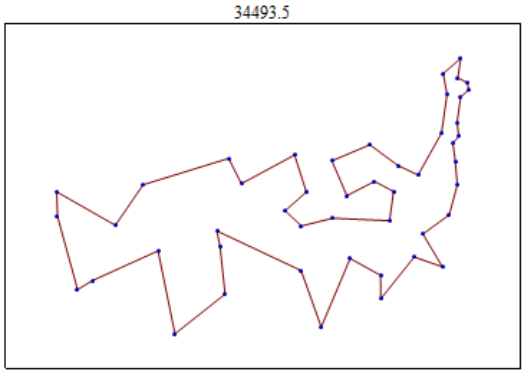
Estado Inicial



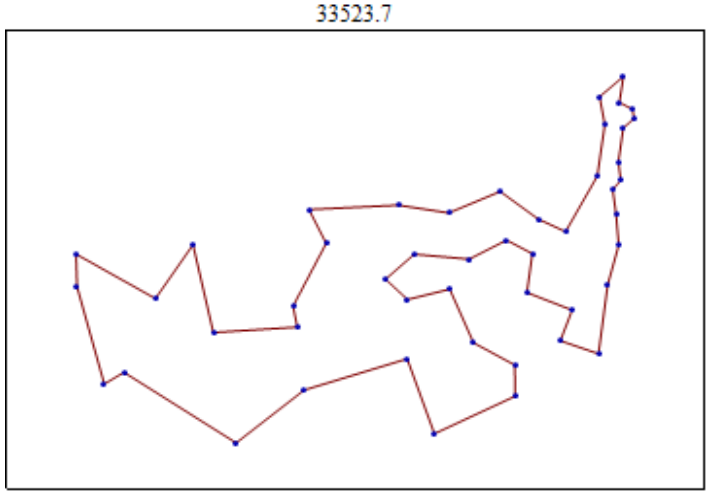
2-opt (8 tarefas)



SA (8 tarefas)



SA ∘ 2-Opt (8 tarefas)
1 geração



SA ∘ 2-Opt (8 tarefas)
10 gerações – resultado ótimo
encontrado na geração 7

```
GENERATION:7
START SOLUTION:
[1](6734, 1453) : [8](7265, 1268) :
GraphPlot[{0 -> 1, 1 -> 2, 2 -> 3,
CANDIDATE SOLUTIONS:
[1](6734, 1453) : [16](6107, 669) :
[1](6734, 1453) : [8](7265, 1268) :
[1](6734, 1453) : [40](6271, 2135)
[1](6734, 1453) : [16](6107, 669) :
[1](6734, 1453) : [16](6107, 669) :
[1](6734, 1453) : [8](7265, 1268) :
[1](6734, 1453) : [22](6101, 1110)
[1](6734, 1453) : [8](7265, 1268) :
CANDIDATE LENGTHS:
34993.4
34344.2
34155.6
33831.7
34229.1
33600.6
34594
33523.7
CANDIDATE LENGTHS (ORDERED):
33523.7
33600.6
33831.7
34155.6
34229.1
34344.2
34594
34993.4
SELECTED SOLUTION:
[1](6734, 1453) : [8](7265, 1268) :
GraphPlot[{0 -> 1, 1 -> 2, 2 -> 3,
6051 ms
```

http://tede.mackenzie.br/jspui/handle/tede/3340?mode=full

Conclusão

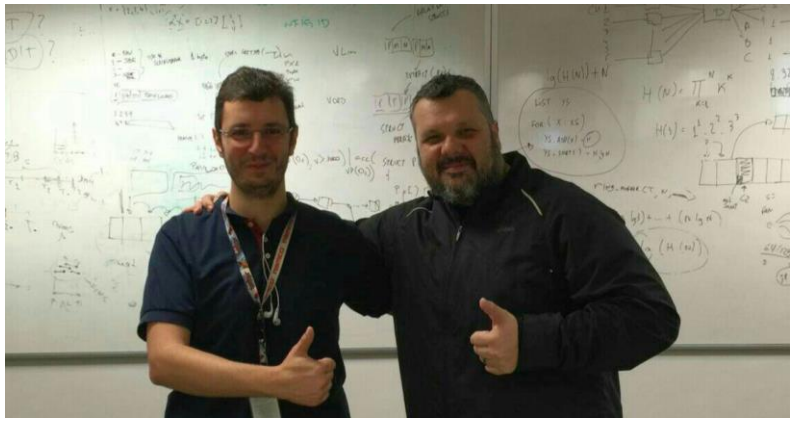
- Existem diversas técnicas e abordagens de problemas através da Inteligência Artificial e *Machine Learning*. É desafiador dominar todos eles
- A maioria dos problemas podem ser vistos como problemas de otimização
- Focar o aprendizado em Algoritmos e Estrututra de Dados (este é o diferencial!)
- Escrever (muito) código, entender os dados
  - "Our world, …, will soon be 'reduced to data'."
  - "… and among the winners will be those who code."
    - https://news.microsoft.com/apac/features/brief-update-future-bots-code-hot-tubs/
- Alguns campos da Matemática são essenciais

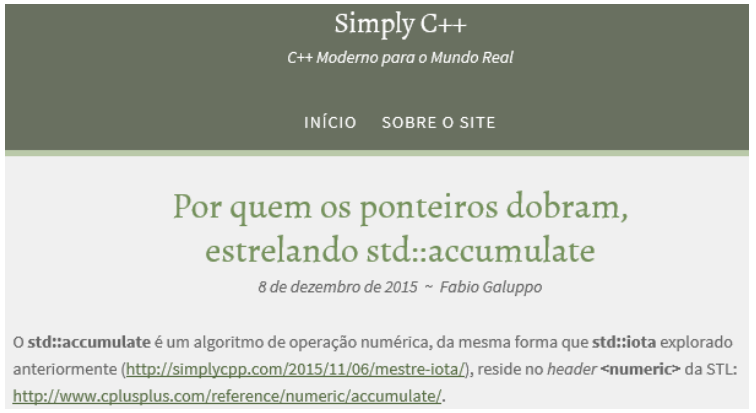
Se quiser saber mais sobre:

C++
Programação Genérica
Iterators
Policy-based Design
Algoritmos
...

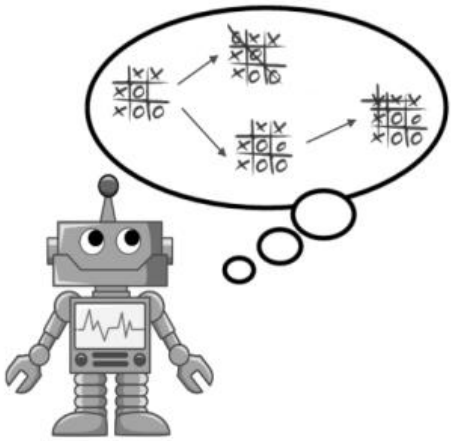
visite:

www.simplycpp.com





http://www.simplycpp.com



Uma breve introdução a Inteligência Artificial para Programadores

Fabio Galuppo, M.Sc.

http://fabiogaluppo.com e http://simplycpp.com/

fabiogaluppo@acm.org

@FabioGaluppo

Microsoft MVP Visual Studio and Development Technologies

https://mvp.microsoft.com/en-us/PublicProfile/9529

http://bit.ly/mvpconf2018