

Computação Bio-inspirada: os engenheiros da natureza

Fabrício Olivetti de França

02 de fevereiro de 2020



1. Introdução
2. Aplicações nas Engenharias
3. Aplicações Divertidas

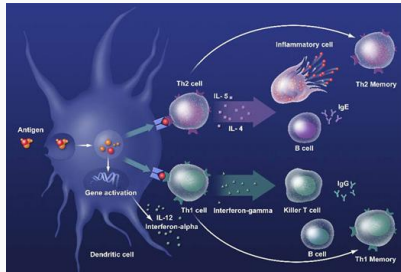
Introdução

The reasonable man adapts himself to the world; the unreasonable one persists in trying to adapt the world to himself. Therefore all progress depends on the unreasonable man.

– George Bernard Shaw, *Maxims for Revolutionists*

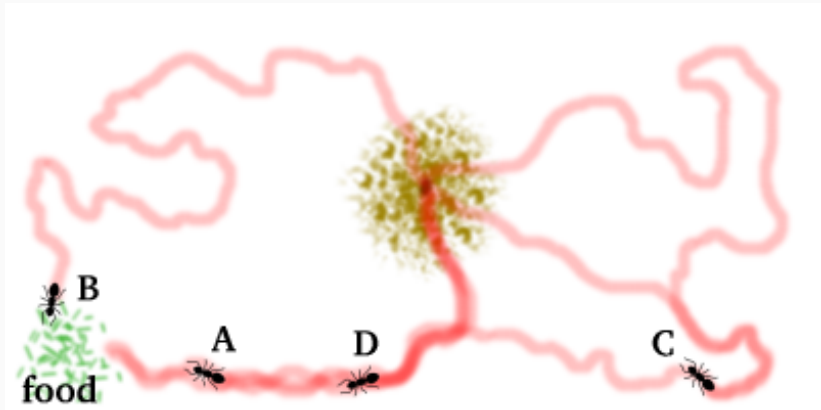
Engenheiros da Natureza

Muitos problemas desafiadores da engenharia e da computação são encontrados e solucionados de forma similar na natureza.

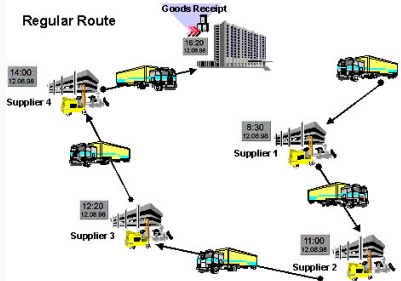
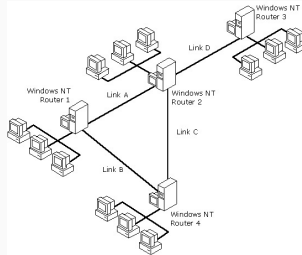


Engenheiros da Natureza

As formigas encontram o menor caminho entre o ninho e o alimento.

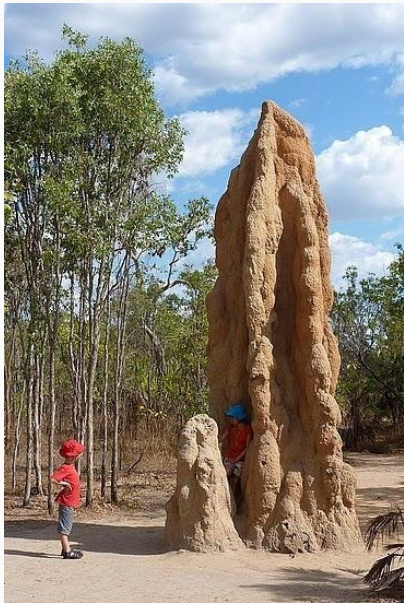


Engenheiros da Natureza

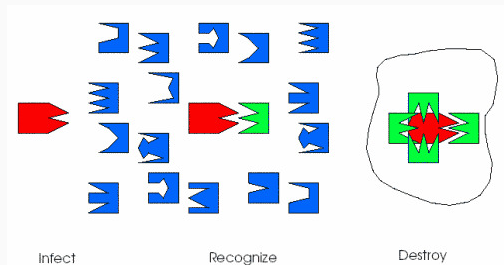


Os **cupins** conseguem construir uma estrutura que seja sólida, conserve água, seja naturalmente aquecida pelo sol e estável.

Engenheiros da Natureza



O nosso [sistema imunológico](#) consegue reconhecer o que faz parte de nosso corpo e o que é nocivo.



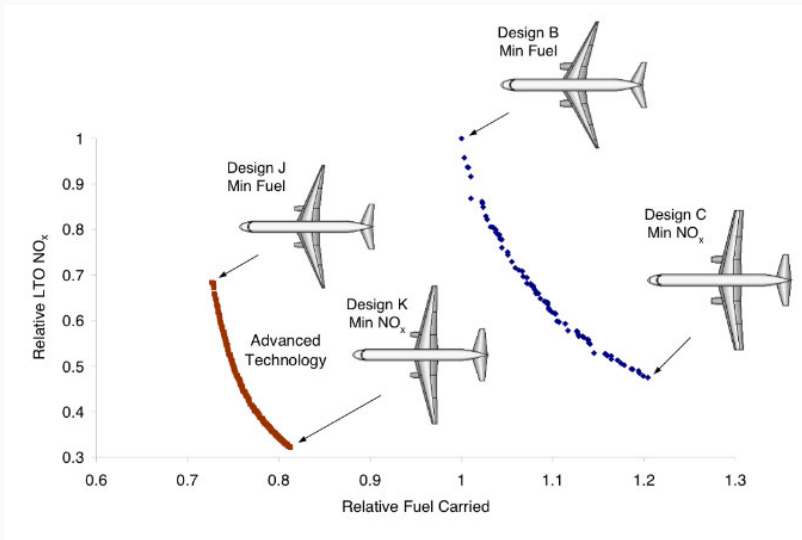
Engenheiros da Natureza



Os pássaros adotam uma [estratégia de vôo](#) para economizar energia, sobreviver ataques e encontrar o destino.



Engenheiros da Natureza



Engenheiros da Natureza

Fungos conseguem expandir criando uma rede de transportes que é eficiente e resistente a ataques.



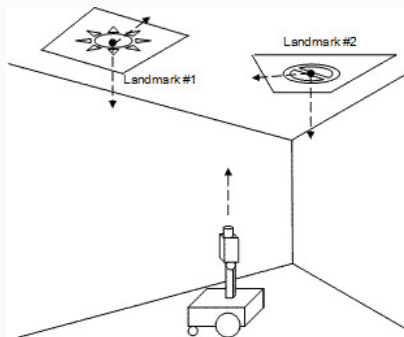
Engenheiros da Natureza



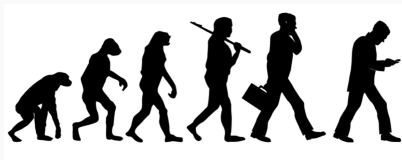
Engenheiros da Natureza

As abelhas criaram um **idioma próprio** utilizando *dança* em que conseguem informar a direção e distância do ninho até flores polinizadas.





O próprio **processo de evolução** é uma otimização da nossa adaptabilidade.



A evolução consegue resolver um problema de caixa preta! Mas por que isso é útil?

Como conseguem?

Apesar de agirem seguindo regras simples e individuais, a união de suas ações levam a **emergência** da solução.



Como conseguem?

Essas regras simples são conhecidas como **heurísticas** .

Procedimento intuitivo para encontrar uma solução com o conhecimento disponível.

Como conseguem?

As heurísticas individuais são coordenadas por uma heurística para a emergência da solução!

Esses algoritmos são classificados como **meta-heurística** .

O que realmente fazem?

Na prática todos esses problemas resolvidos pela natureza são problemas de otimização.

Dada uma função-objetivo e suas restrições, encontre a solução factível que retorne o maior valor para essa função.

Assumindo que a função mede a qualidade.

O que realmente fazem?

Os problemas da engenharia se resumem em:

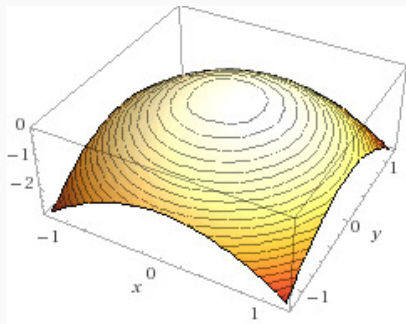
1. formulação matemática
2. otimizar a função-objetivo
3. verificar a solução obtida

Descrever o problema como uma função matemática e inequações que representam restrições.

Necessário utilizar o conhecimento interdisciplinar: física, química, biologia, etc.

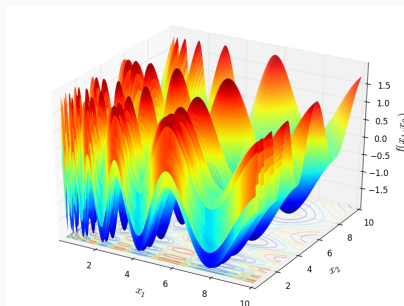
Encontrar a(s) solução(ões) da função-objetivo.

$$f(x) = -x^2 \leftarrow \text{fácil}$$



Encontrar a(s) solução(ões) da função-objetivo.

$$f_{\text{Vincert}}(\mathbf{x}) = - \sum_{i=1}^n \sin(10 \log(x))$$



Mesmo ao encontrar a solução ela pode não ser factível no problema real.

Isso ocorre por imprecisões ou falta de informação na hora de criar a função-objetivo.

Em altas dimensões não conseguimos ver as características dessa função:

ela é contínua? tem superfície suave? tem vários ótimos? é diferenciável?

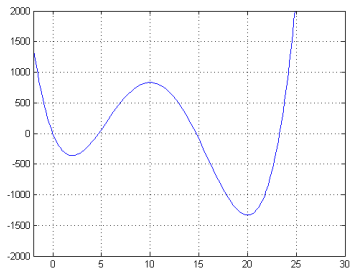
Quando a nossa função é bem determinada podemos usar alguns métodos tradicionais para encontrar a solução.

Algoritmo de Busca

```
x <- solucaoAleatoria
while change do
  forEach x' in vizinhos(x) do
    if f(x') > f(x)
      then x = x'
return x
```

A vizinhança $N(x)$ da solução x é o conjunto de soluções próximas a ela.

Algoritmo de Busca



Algoritmo de Busca

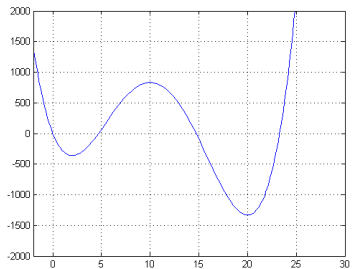
```
x <- solucaoAleatoria
while change do
  x' = x + f'(x)/f''(x)
  if f(x') > f(x)
    then x = x'
return x
```

Mas e se $f(x)$ não for diferenciável??

O poder das soluções da natureza está no uso de diversos agentes fazendo a busca em paralelo.

Dessa forma é possível explorar as diversas regiões do espaço de soluções.

Algoritmo de Busca



Exploração x Exploração

Exploração: o ato de explorar toda a região do espaço de busca.

Exploração: o ato de explorar apenas a região de vizinhança.

```
p <- populaçãoInicial  
while not converged:  
  p' <- altera(p)  
  p <- seleciona(p + p')
```

Processo natural em que um grupo de uma espécie se adapta sucessivamente para:

- sobrevivência
- equilíbrio ecológico
- diversidade

Proposta por Charles Darwin e explica porque certas características se tornam comuns enquanto outras desaparecem.

1. São produzidos mais herdeiros do que o sustentável
2. Características do indivíduo define sua probabilidade de sobrevivência
3. Características são hereditárias

Vamos ilustrar com a luta pela sobrevivência de coelhos em um território com raposas.

Na população de coelhos, alguns são mais espertos e mais rápidos que os outros.

Esses têm maior chance de sobrevivência.

Produzem **mais coelhos**.

Com o cruzamento dos coelhos rápidos e dos espertos na população novas variedades de coelhos surgem:

Coelhos rápidos e burros; lentos e espertos, lentos e burros, **rápidos e espertos**.

Evolução

A natureza joga de vez em quando um “coelho selvagem” nessa população mutando um ou outro filhote dessas crias.

As crias não são cópias exatas dos pais, mas variações aleatórias.



Com o passar das gerações espera-se que a população de coelhos se torne mais rápida e esperta do que a inicial.

Mas as raposas também evoluem. . .

Algoritmo Evolutivo

```
p <- populaçãoInicial
enquanto não convergir faça
  pais <- selecionaPais(p)
  filhos <- reproduz(pais)
  xmen <- muta(filhos)
  p <- seleciona(p + xmen)
```

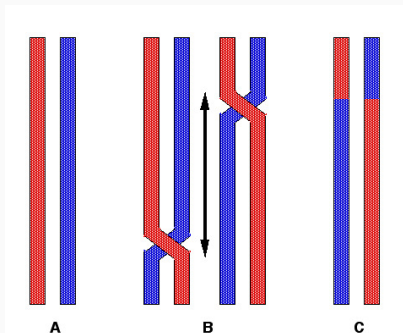
Cada indivíduo da população representa uma solução, eles são representados computacionalmente como mais conveniente.

Ex.: vetor de variáveis.

Denominamos essa representação de **cromossomo** .

Reprodução

A reprodução simula a reprodução das espécies em que ocorre troca de material genético.



Com uma representação vetorial:

`pai1 = [3 0 2 | 5 2 1]`

`pai2 = [5 1 2 | 5 0 3]`

`filho1 = [3 0 2 5 0 3]`

`filho2 = [5 1 2 5 2 1]`

A mutação promove a **novidade** no nosso conjunto de soluções.
Previne que todos os indivíduos sejam iguais.

A mutação no algoritmo é uma perturbação aleatória da solução:

`pai = [3 0 *2* 5 2 1]`

`filho = [3 0 *4* 5 2 1]`

Uma vez que temos a população de pais e filhos, temos que selecionar aqueles que sobreviverão.

Lembrando que na natureza os mais aptos tem maior **chance** de sobrevivência.

Para verificar as soluções mais aptas, usamos a própria função-objetivo (**fitness**). Quem tem maior valor é mais apto.

Fazemos então uma seleção probabilística de tal forma que aqueles com maior fitness tenham maior chance.

- **Algoritmos Genéticos** : proposto inicialmente por Holland com objetivo de estudar os fenômenos de adaptação.
- **Estratégias Evolutivas**: introduzido por Rechenberg com o intuito de otimizar parâmetros de uma função não-linear.
- **Programação Evolutiva**: representação de uma solução como uma máquina de estado finito, proposta por Fogel et al.
- **Programação Genética**: intuito de evoluir programas de computador, proposta por Koza.

http://rednuht.org/genetic_cars_2/

<http://alteredqualia.com/visualization/evolve/>

Colônia de Formigas

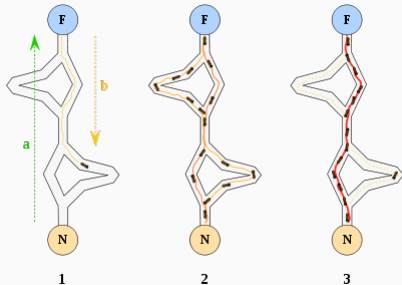
Foi observado que as formigas eram capazes de encontrar o menor caminho entre o ninho e uma fonte de alimento.

Inicialmente o caminho aparentava desordem, mas com o passar do tempo elas seguiam um mesmo caminho.



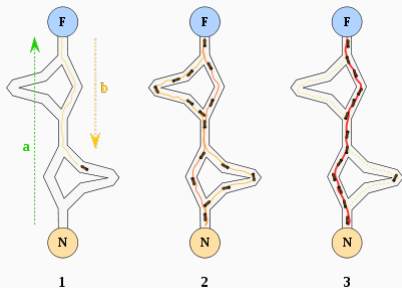
Colônia de Formigas

Inicialmente elas não têm informação alguma e andam aleatoriamente.



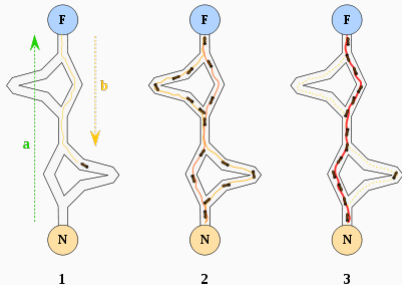
Colônia de Formigas

Ao encontrar um alimento, elas retornam ao ninho depositando feromônio pelo caminho.



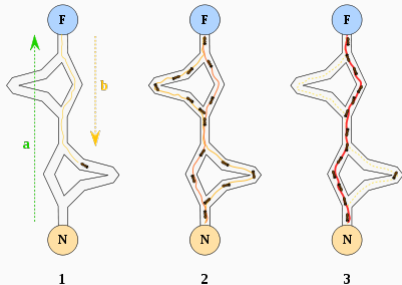
Colônia de Formigas

Os menores caminhos tem uma probabilidade maior de ter mais feromônio.



Colônia de Formigas

Coletivamente o menor caminho receberá cada vez mais feromônio e se tornará o único caminho.



```
enquanto não convergir faça  
  para cada formiga:  
    solução  <- constroiSolução(feromônio)  
    feromônio <- atualiza(feromônio)
```

Para construir uma solução a formiga utiliza a informação do feromônio como probabilidade:

$$p_{xy}^k = \frac{\tau_{xy}^\alpha \eta_{xy}^\beta}{\sum_z \tau_{xz}^\alpha \eta_{xz}^\beta}$$

O feromônio é atualizado proporcionalmente a qualidade da solução.

$$\tau_{xy} = (1 - \rho)\tau_{xy} + \sum_k \tau_{xy}^k$$

$$\Delta\tau_{xy}^k = \begin{cases} Q/L_k & \text{if ant } k \text{ uses curve } xy \text{ in its tour} \\ 0 & \text{otherwise} \end{cases}$$

<https://www.youtube.com/watch?v=SJM3er3L6P4>

<https://www.youtube.com/watch?v=qaiyydNp4TU>

O funcionamento do cérebro também serve como fonte de inspiração.

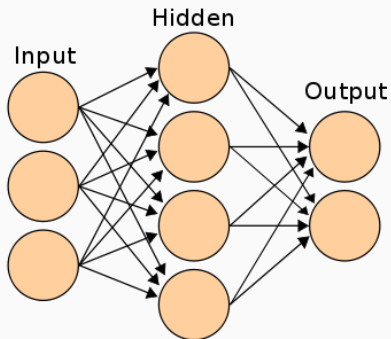
Se for possível simular o funcionamento dele no computador, o computador será capaz de aprender?



As redes neurais artificiais modelam o aprendizado através de um grafo que simula o processamento da informação.

Cada nó desse grafo é um neurônio e as arestas são utilizadas para transmitir informação de um neurônio a outro.

Redes Neurais Artificiais



Uma rede neural aprender através de exemplos com saídas conhecidas e atualiza o peso para que essa se aproxime do que é esperado.

Tendo a entrada e a saída bem determinada existe uma fórmula fechada para atualizar os pesos: **retropropagação** .

<https://www.youtube.com/watch?v=ocB8uDYXtt0>

<https://www.youtube.com/watch?v=vI8BS2Jikn8>

Aplicações nas Engenharias

As equações de Navier-Stokes são usadas para modelar o fluxo de ar em torno das asas de um avião. Mas elas são computacionalmente caras para avaliar e otimizar. É possível usar uma Rede Neural para criar uma aproximação dessa função.

Predição de estrutura de RNA

Redistribuição de Rede de Distribuição de energia para Smart Grid

Encontrar a escala ótima para enfermeiros em um hospital,
maximizando a satisfação dos funcionários

Predição de estruturas cerâmicas e ligas de metal minimizando entropia

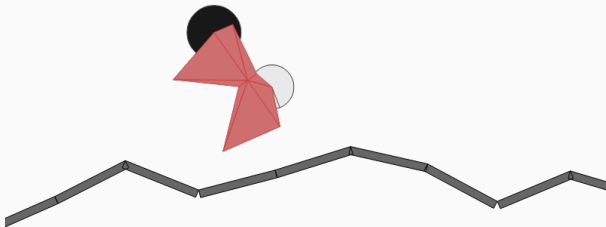
Roteamento dinâmico utilizando Algoritmo de Colônia de Formigas

Configuração automática de sensores para controle e tolerância a falhas de sistemas de suspensão

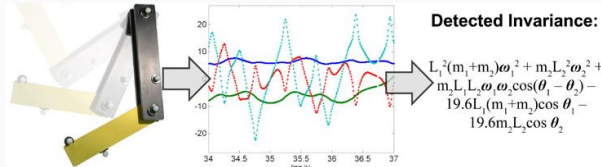
Aplicações Divertidas

Evoluindo um Carro

http://rednuht.org/genetic_cars_2/



https://www.youtube.com/watch?v=56zz_EIAP7o



Jogar Mega Man

<https://www.youtube.com/watch?v=yIxIYfZ5wrc>



Corrida de Carros

<https://www.youtube.com/watch?v=1dvSnvGzekg>



<https://www.youtube.com/watch?v=HZjwcDGgJgU>

