

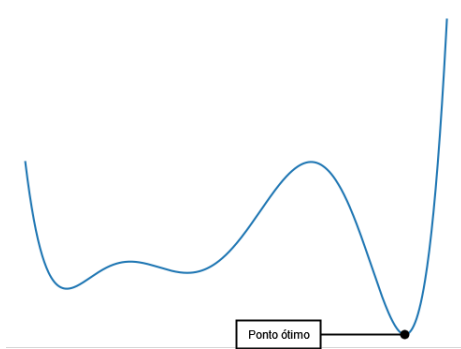
Simulated Annealing

Matheus Marinho, Guilherme Freire, Italo Costa

November 15, 2024

O que é otimização?

- Ação ou efeito de otimizar;
- Produzir condições apropriadas para o melhor desenvolvimento de algo;
- Encontrar a melhor solução dentro de um espaço de soluções;
- Processo de busca dos argumentos que resultam no valor extremo de uma ou mais funções que representem o problema.



Exemplos práticos

- Busca de caminho mais curto em mapas;
- Minimização de custos;
- Maximização de lucros;
- Design de circuitos;
- Planejamento logístico.

Desafios da otimização

Simulated Annealing: Analogia Física

O recozimento de metais (também chamado de recristalização) é um processo de alteração de propriedades de um material metálico por aquecimento e resfriamento lento. Quando um material é submetido a um processamento a frio, sua estrutura cristalina é deformada e surgem vários pontos de tensão. Os cristais deformados têm mais energia que os não deformados, por causa da desorganização da estrutura cristalina nas interfaces entre os grãos. Estas estruturas são estáveis nesta temperatura, já que os átomos não têm mobilidade suficiente para alterar a sua organização cristalina. Havendo oportunidade, os átomos irão se deslocar visando um arranjo mais perfeito e regular, de menor energia.

Simulated Annealing: Algoritmo Básico

O método de Recozimento Simulado (simulated annealing) é um tipo de método de busca local de implementação extremamente simples proposto por METROPOLIS et al. (1953), que perceberam que a natureza faz na verdade a minimização da energia da estrutura cristalina quando o material é recozido para remover defeitos de sua estrutura atômica. KIRKPATRICK et al. (1983) estenderam o método de otimização termodinâmica de Metropolis para o problema de otimização combinatorial.

- O estado do sistema termodinâmico corresponde à solução atual do problema;
- A equação de energia para o sistema é a função objetivo;
- O estado de referência é análogo ao mínimo global da função;
- O que evita o algoritmo de ficar preso em um mínimo local é a seleção adequada da programação do recozimento:
 - Temperatura inicial;
 - Número de iterações do algoritmo com a mesma temperatura;
 - Estratégia de redução de temperatura;
 - Definição de uma estrutura de vizinhança.

A partir de um ponto no espaço de soluções, calcula-se um novo ponto vizinho do atual. Se a energia (valor da função objetivo) é menor neste novo ponto, este passa a ser nosso ponto atual. Se a energia é maior neste novo ponto, ele não é automaticamente descartado. Há uma certa probabilidade de ele ser aceito como o novo ponto atual, e esta probabilidade é tão maior quanto maior for o parâmetro temperatura ou quanto menor for a diferença de energia entre os dois pontos.

Queremos reduzir o valor da energia, mas ainda assim aceitamos como nova solução um ponto que seja pior que o anterior? O motivo para esta aparente contradição é simples: para que se vá de um mínimo local a um mínimo global da estrutura acessando apenas pontos vizinhos, é necessário que, pelo menos em alguns pontos ao longo desta trajetória, a energia do sistema aumente. Em outras palavras, a única maneira de ir de um mínimo local para o mínimo global (ou para outro mínimo local) é aumentando a energia do sistema.

Redução de temperatura

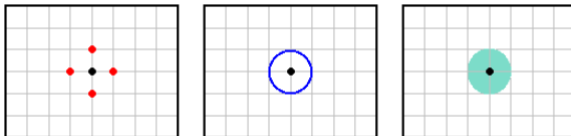
- Qualquer função monótona decrescente pode ser usada como função de resfriamento. Uma escolha muito usual é a redução linear da temperatura, fazendo $T^{k+1} = \alpha T^k$. O parâmetro α varia tipicamente entre 0,7 e 0,95. Funções exponenciais de resfriamento são também utilizadas na literatura da área.
- Como uma regra geral, o número de temperaturas empregadas (n_1) deve ser pequeno – algo entre 10 e 20 costuma ser suficiente. Já o número de iterações com a mesma temperatura (n_2) deve ser um pouco maior, para que se dê tempo suficiente ao algoritmo para buscar o espaço de soluções com aquela temperatura

Critério de aceitação

- O critério proposto por Metropolis consiste em avaliar a diferença de energia entre a solução atual e a nova solução e calcular $\Delta \equiv E^{k+1} - E^k$. Se Δ é negativo, isto significa que a solução atual é melhor que a anterior, e esta última é substituída. Se Δ é positivo, a probabilidade de esta solução de maior energia substituir a anterior é dada por $p = e^{-\frac{\Delta}{T}}$.
- Como implementar este critério probabilístico de aceitação?
Primeiramente é sorteado um número randômico r a partir de uma distribuição uniforme no intervalo $[0, 1]$. Considere que p representa a probabilidade de a transição para este novo ponto ser aceita. A probabilidade do número sorteado r ser menor ou igual do que p , também contido no intervalo $[0, 1]$, é $(100p)\%$.
- Logo, se $r < p$ a transição é aceita. Do contrário, este ponto é descartado e um novo vizinho do ponto atual é calculado

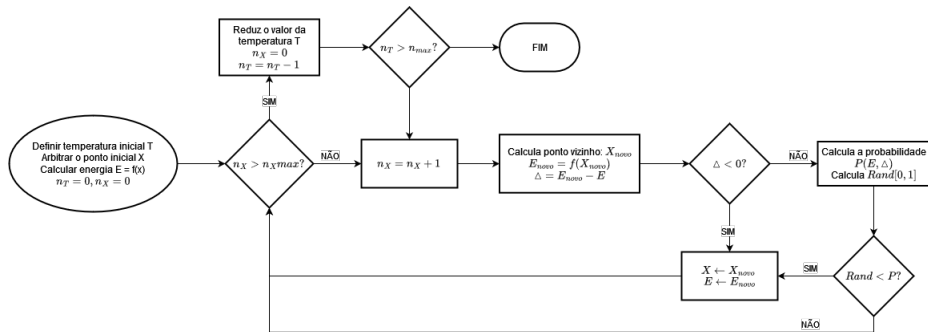
Estrutura de vizinhança

- A definição da estrutura de vizinhança é fundamental para a implementação do algoritmo. Uma estrutura de vizinhança nada mais é do que uma maneira de calcular uma solução próxima à solução anterior. Para um mesmo problema, várias estruturas de vizinhança são cabíveis. A vizinhança de um ponto corresponde ao conjunto de todos os pontos que lhe são vizinhos.
- Podemos calcular um ponto “vizinho” no plano real de várias maneiras. Poderíamos: (i) somar (ou subtrair) uma quantidade fixa δ em uma ou mais direções; (ii) dar um passo de tamanho δ em alguma direção arbitrária; (iii) selecionar qualquer ponto que diste não mais que δ do ponto anterior.



Uma característica do recozimento simulado: não há qualquer garantia de que o último ponto visitado será o melhor ponto já visitado pelo algoritmo. Uma modificação simples e efetiva no algoritmo original consiste em registrar em uma variável à parte este melhor resultado encontrado. Assim, é possível que o seu problema seja resolvido mesmo com uma programação de recozimento pobre.

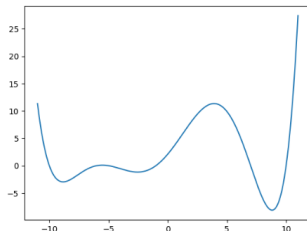
Simulated Annealing: Fluxograma



Exemplo ilustrativo

Para entender o significado dos parâmetros do método, considere o problema de minimização da função $F(x)$ apresentada abaixo.

$$F(x) = \frac{1}{10000}(x+10)(x+6)(x+5)(x+1)(x-7)(x-10) \quad (1)$$



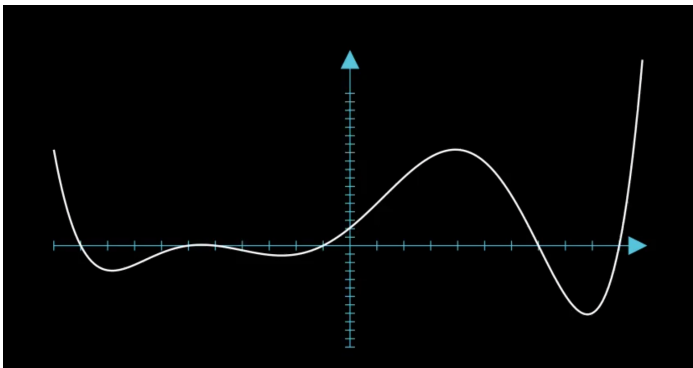
Observe que existem mínimos locais em $x = -8,834$, $x = -2,546$ e $x = 8,817$, sendo este último o mínimo global. A função apresenta máximos locais em $x = -5,518$ e $x = 3,914$.

Para este exemplo foram escolhidos os parâmetros: $X_{inicial} = -11$, $T_{inicial} = 60$, $\alpha = 0,75$, $n_1 = 20$, $n_2 = 50$ e $\delta = 1$

Código utilizado:

```
1 def SA(F, x0, t0, dx, alpha, n1, n2):
2     points = []
3     bests = []
4     rng = np.random.default_rng()
5     x = x0
6     t = t0
7     best = x
8     for i in range(n1):
9         for j in range(n2):
10             a = rng.random() * 2 - 1
11             x1 = viz(x, dx, a)
12             delta = F(x1) - F(x)
13             if x1 > -11 and x1 < 11:
14                 if delta < 0:
15                     x = x1
16                 else:
17                     b = rng.random()
18                     temp = b - np.exp(-delta/t)
19                     if temp < 0:
20                         x = x1
21             if F(x) < F(best):
22                 best = x
23             points.append(x)
24             bests.append(best)
25         t = resfr(t, alpha)
26     return points
```


Trajeto do Algoritmo



Possíveis melhorias

Perguntas?