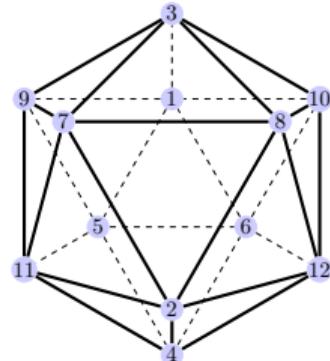


Metaheuristicas - 7 - Simulated Annealing

Alexandre Checoli Choueiri

05/03/2023

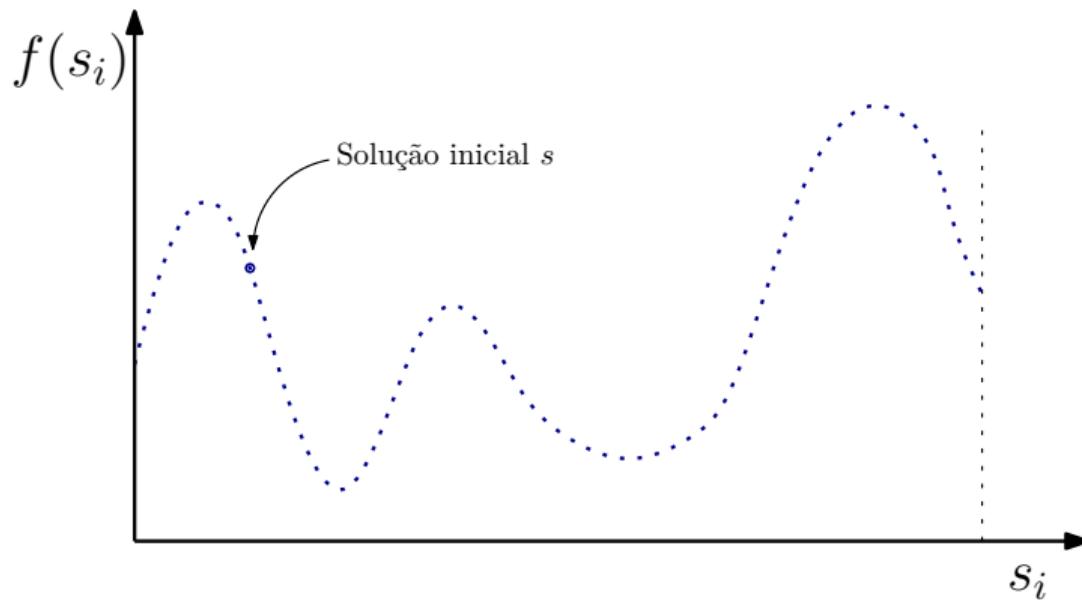


- ① O problema da busca local
- ② Paradigma exploração x intensificação
- ③ Metaheurísticas
- ④ O processo de Annealing
- ⑤ A metáfora no algoritmo
- ⑥ Pseudocódigo
- ⑦ Parâmetros do algoritmo
- ⑧ Atividades

Problemática

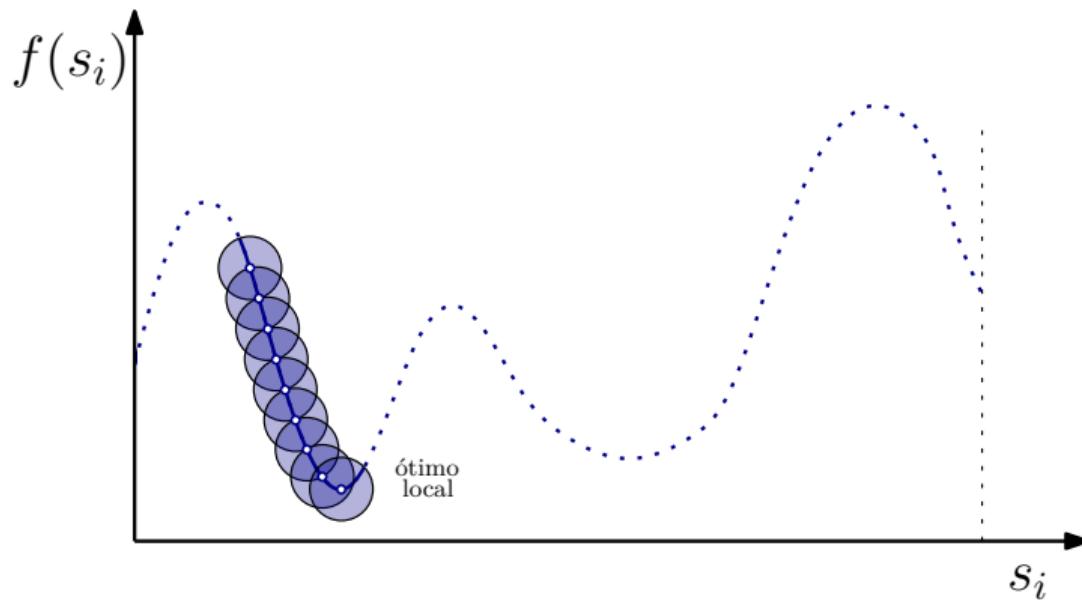
Vimos a primeira e mais simples metaheuristica, chamada busca local. Esta classe de algoritmos, no entanto, possui algumas limitações.

Relembrando a busca local



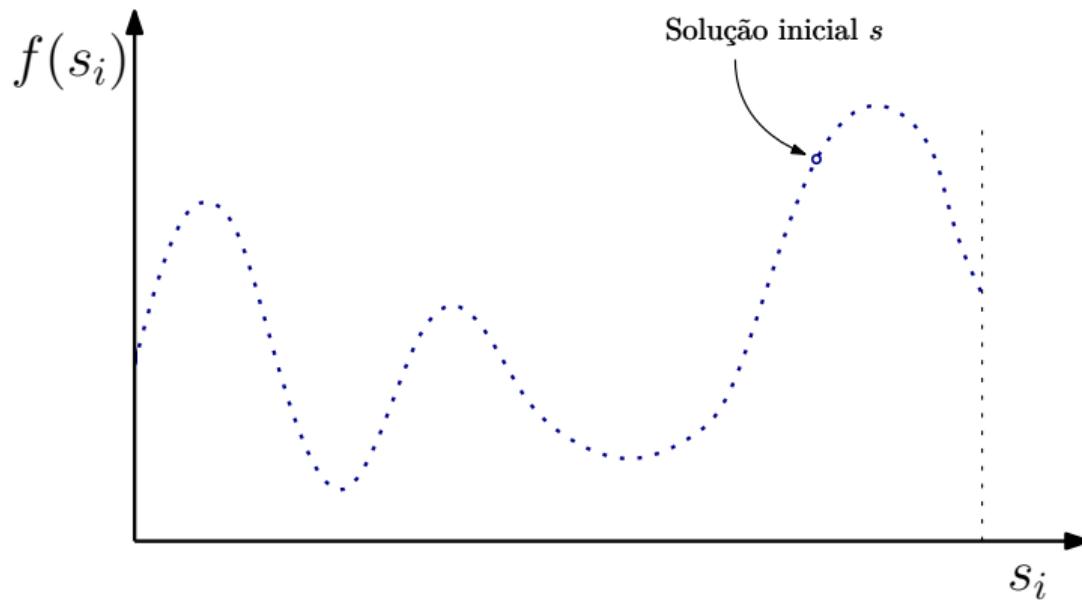
Na **busca local**, partimos de uma solução inicial, e **intensificamos** a busca até chegar em um ótimo local.

Relembrando a busca local



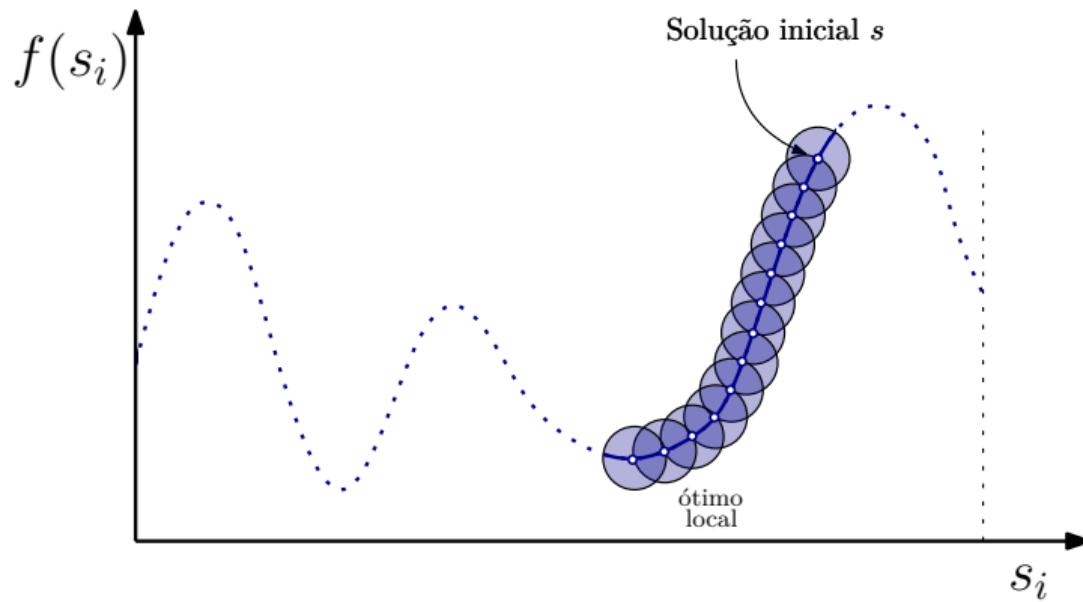
Na **busca local**, partimos de uma solução inicial, e **intensificamos** a busca até chegar em um ótimo local.

Relembrando a busca local



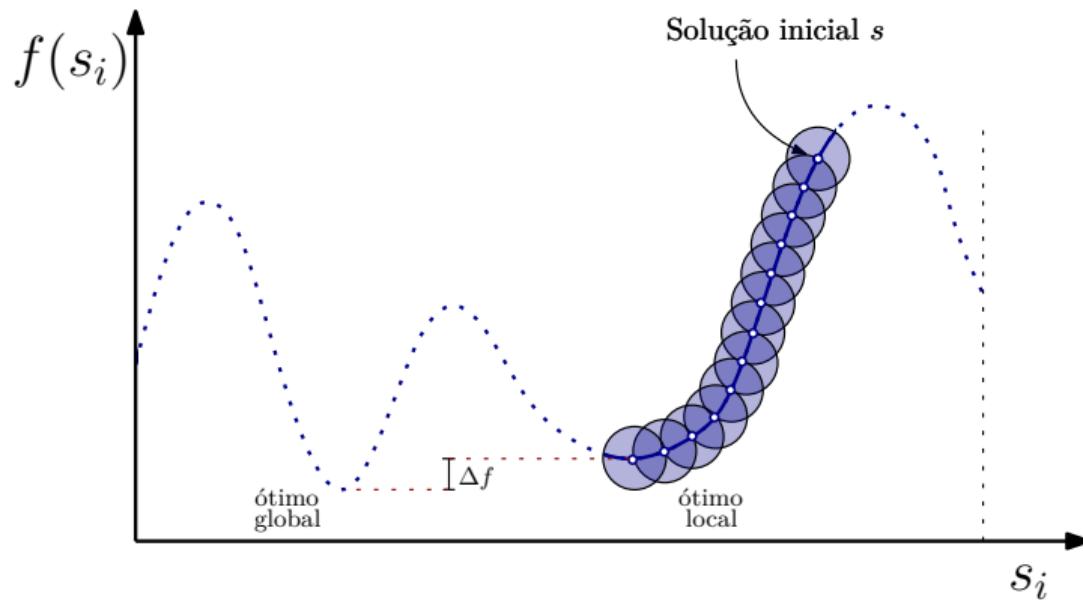
O que aconteceria se a solução inicial fosse como mostrado acima?

Relembrando a busca local - Problemática



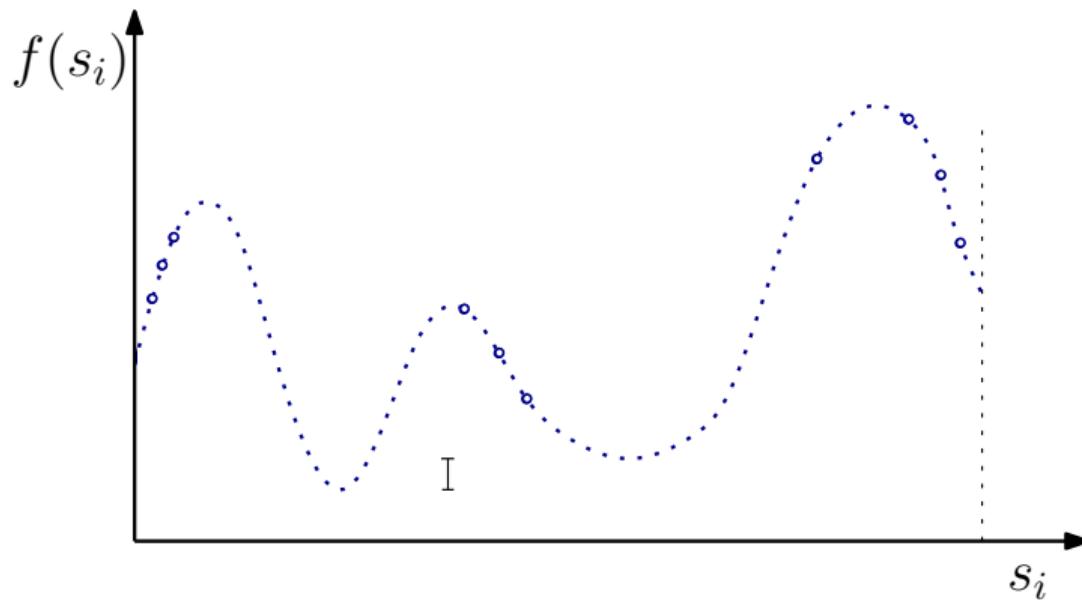
Após a intensificação, a busca chega a um ótimo local que **não é o ótimo global**.

Relembrando a busca local - Problemática



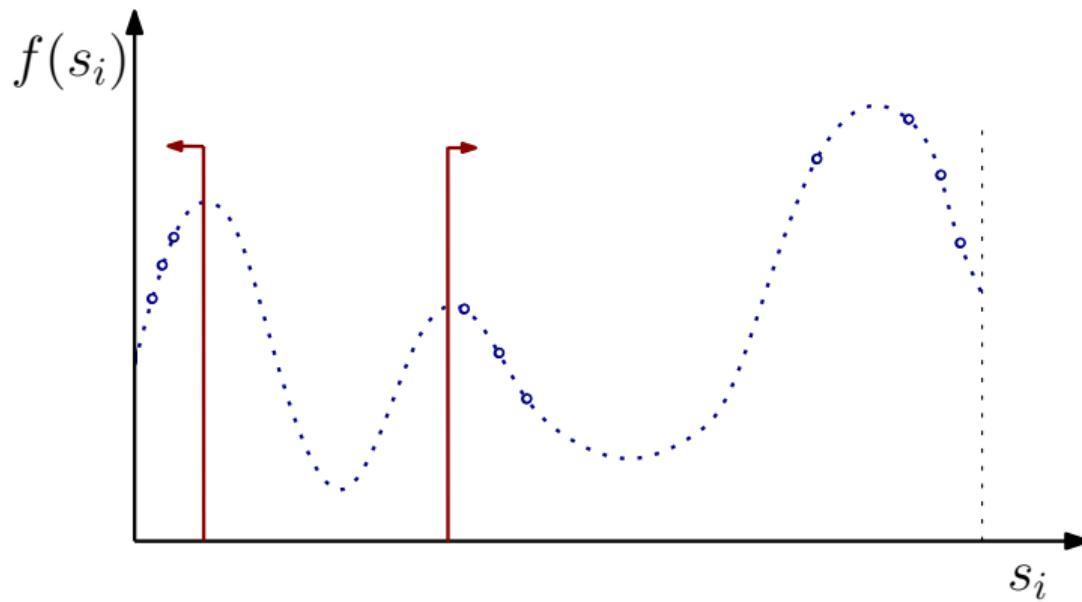
Após a intensificação, a busca chega a um ótimo local que **não é o ótimo global**.

Relembrando a busca local - Problemática



Com isso podemos inferir que a **busca local** é muito sensível à escolha da solução inicial.

Relembrando a busca local - Problemática

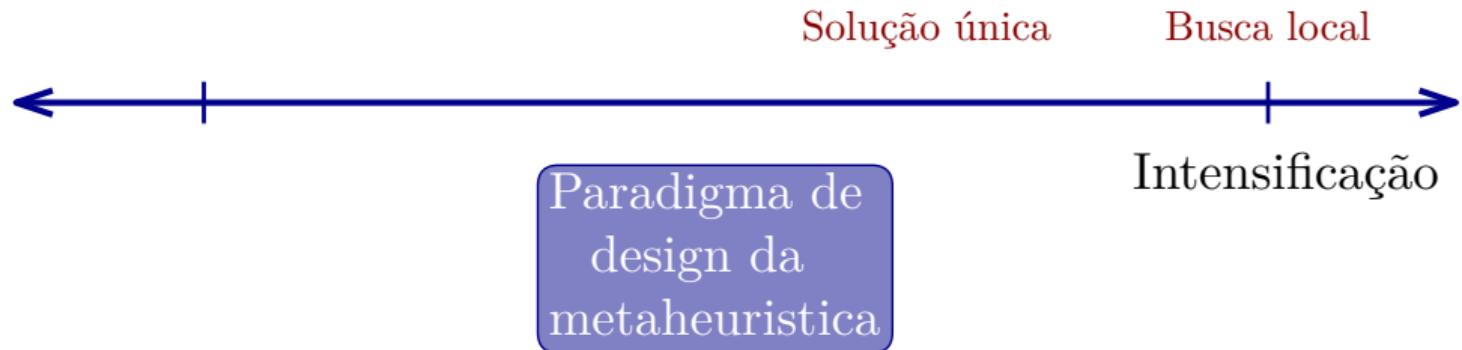


Qualquer solução inicial gerada nas áreas delimitadas acima não chegará ao ótimo global.

Paradigma exploração x intensificação

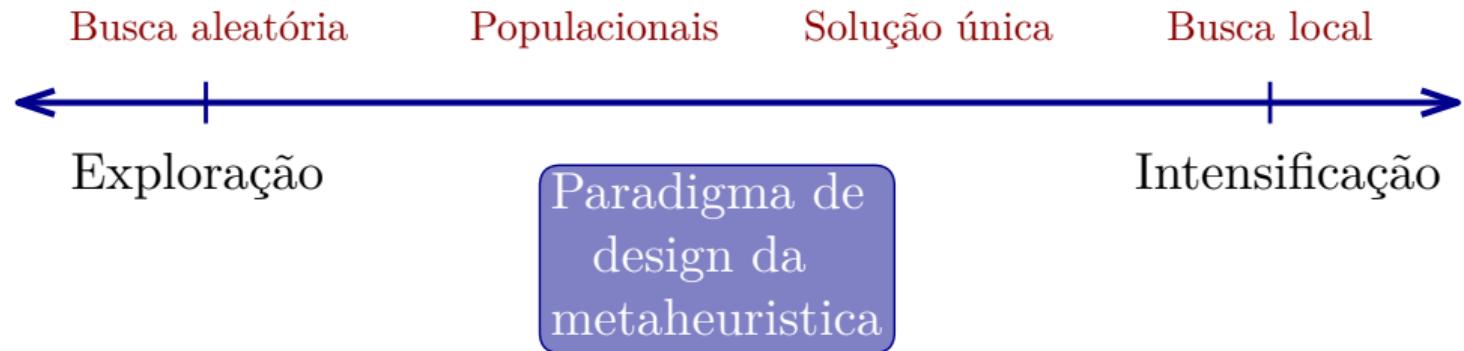
Essa intuição gera o principal paradigma das metaheurísticas, chamado de
exploração x intensificação
do *fitness landscape*.

Paradigma exploração x intensificação



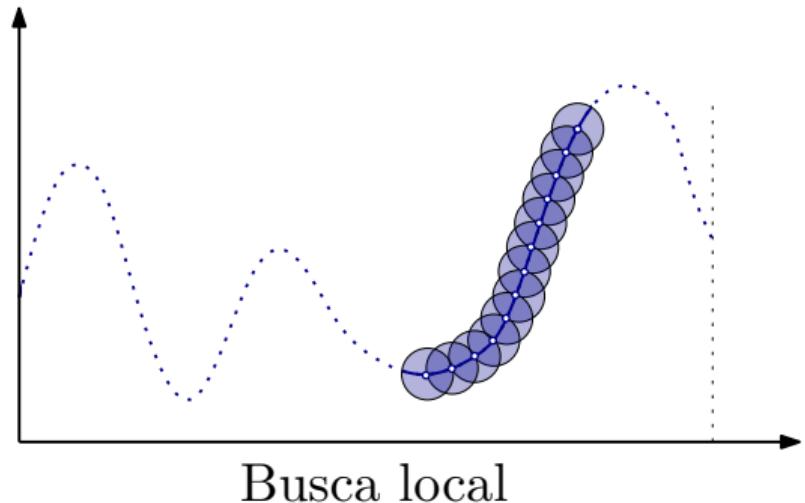
A **intensificação** está ligada a exploração intensiva de uma vizinhança (a maior intensificação é dada pela busca local).

Paradigma exploração x intensificação

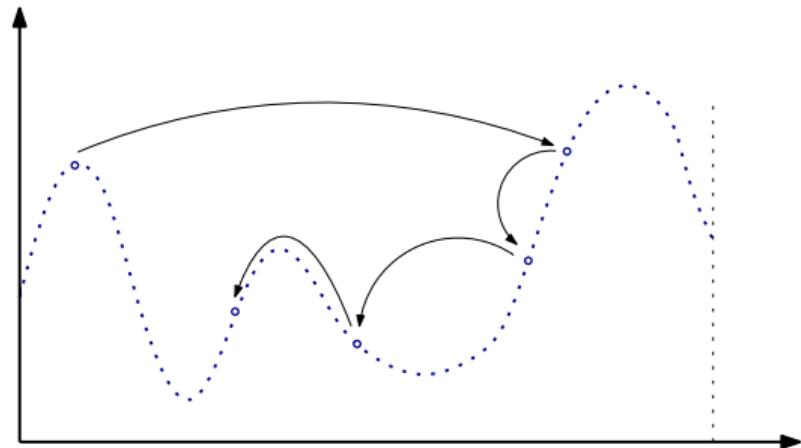


A **exploração** está ligada a capacidade de explorar diferentes regiões do *fitness landscape*. O extremo da exploração é a geração aleatória de soluções, ou seja, a **busca aleatória**.

Paradigma exploração x intensificação



Busca local



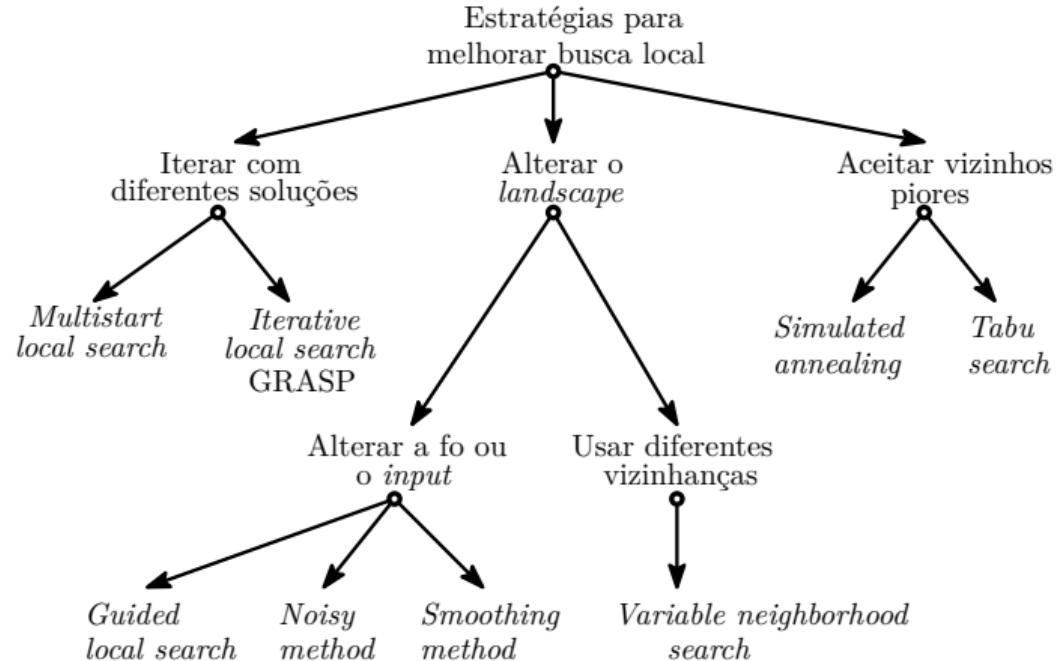
Busca aleatória

Paradigma exploração x intensificação

Conclusão

1. O design de qualquer metaheuristica deve ponderar o paradigma **exploração x intensificação**. É necessário conseguir **intensificar** a busca em algumas regiões ao mesmo tempo que uma grande região do *fitness landscape* é **explorado**.
2. Os dois conceitos são conflitantes, de forma que uma boa metaheuristica faz um compromisso entre eles.
3. A **intensificação** está relacionada a melhoria continua de uma solução por meio de uma vizinhança.
4. O principal objetivo da **exploração** é evitar que a busca fique **presa em ótimos locais**.

Formas de evitar ótimos locais



De forma geral, as diferentes abordagens para se evitar ótimos locais são mostradas acima, bem como os algoritmos mais representativos.

Inspiração para metaheurísticas

Os criadores de metaheurísticas perceberam que uma inspiração para o *design* de algoritmos (e o paradigma **exploração** x **intensificação**) eficientes pode vir diretamente de associações com fenômenos e comportamentos da própria natureza. Alguns exemplos são mostrados abaixo:

1. **Recozimento simulado**: Imita o comportamento de resfriamento de materiais.
2. **Colônia de formigas**: Imita o comportamento das formigas.
3. **Algoritmos genéticos**: Simula a evolução de uma população ao longo de gerações, com mutação, nascimentos, mortes, etc.

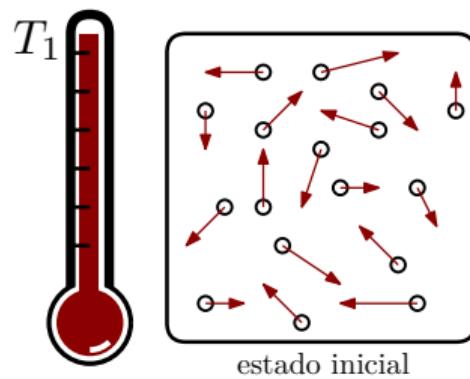


Simulated Annealing

Um desses algoritmos é o chamado **Simulated Annealing** - SA (recozimento simulado, têmpora simulada). O algoritmo é baseado nos princípios de mecânica estatística, e tem a seguinte associação com um fenômeno físico:

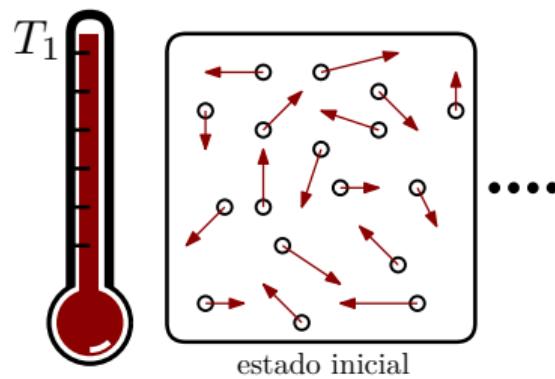
O processo de **Annealing** visa melhorar as características de um material por meio do **aquecimento** e posterior **refrigeração** a uma taxa controlada. A cada redução de temperatura os átomos vão se rearranjando em um estado de menor energia. Ao atingir a temperatura mais baixa e em equilíbrio, o material chega a uma estrutura atômica "perfeita" (com energia = 0). Se o processo for executado de forma correta, o novo material tem boas propriedades físicas.

Simulated Annealing



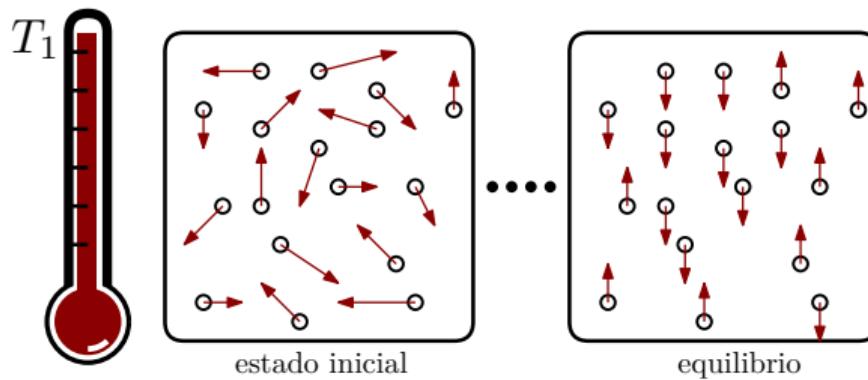
Inicialmente a estrutura é submetida a uma temperatura alta, de forma que os átomos ficam com muita energia (E).

Simulated Annealing



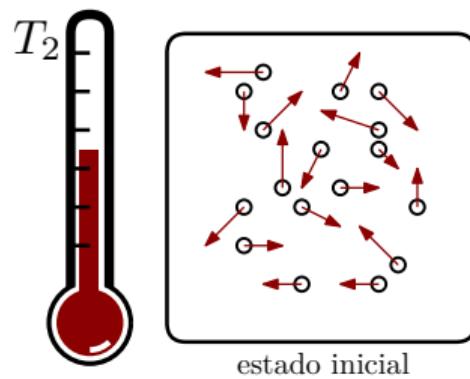
Inicialmente a estrutura é submetida a uma temperatura alta, de forma que os átomos ficam com muita energia (E). Após algum tempo (mantido na mesma temperatura), a estrutura atinge um estado de equilíbrio.

Simulated Annealing



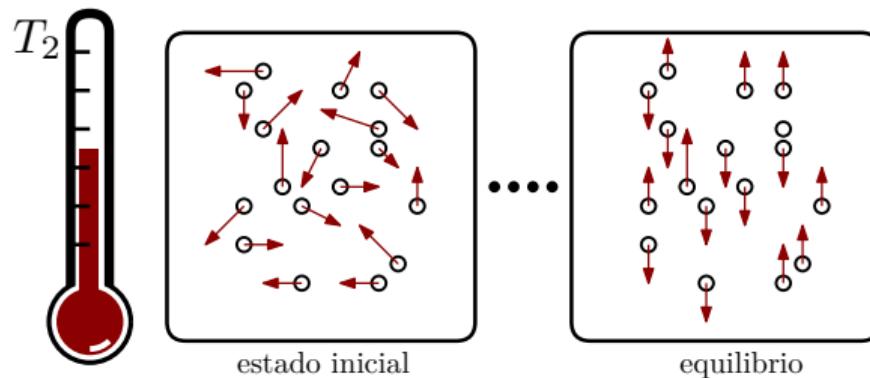
Inicialmente a estrutura é submetida a uma temperatura alta, de forma que as átomos ficam com muita energia (E). Após algum tempo (mantido na mesma temperatura), a estrutura atinge um estado de equilíbrio.

Simulated Annealing



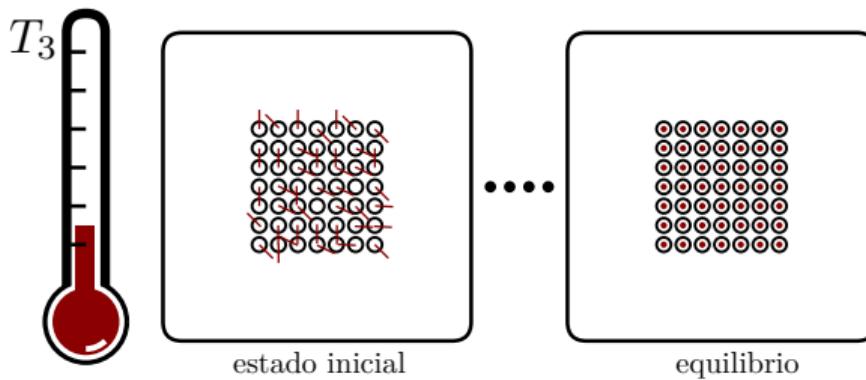
Após o equilíbrio ser atingido, a **temperatura é reduzida**, e o processo é repetido até que um novo estado de equilíbrio de energia seja atingido.

Simulated Annealing



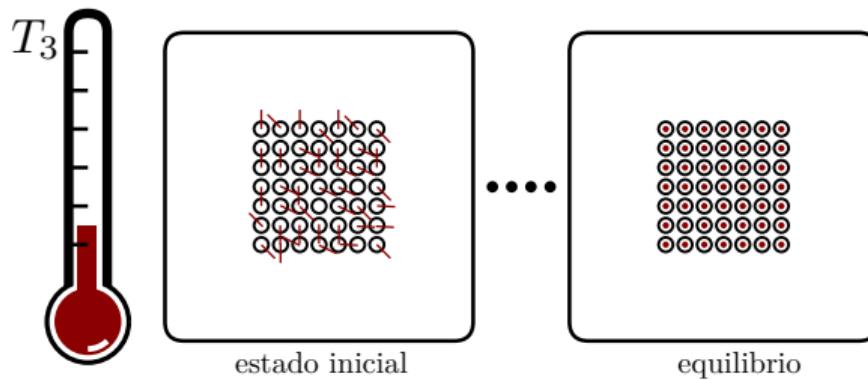
Após o equilíbrio ser atingido, a temperatura é reduzida, e o processo é repetido até que um novo estado de equilíbrio de energia seja atingido.

Simulated Annealing



O processo é repetido até que uma **temperatura** final seja atingida. Se o processo foi feito de forma controlada, a **estrutura gerada mantém uma configuração de energia minima (min E)**.

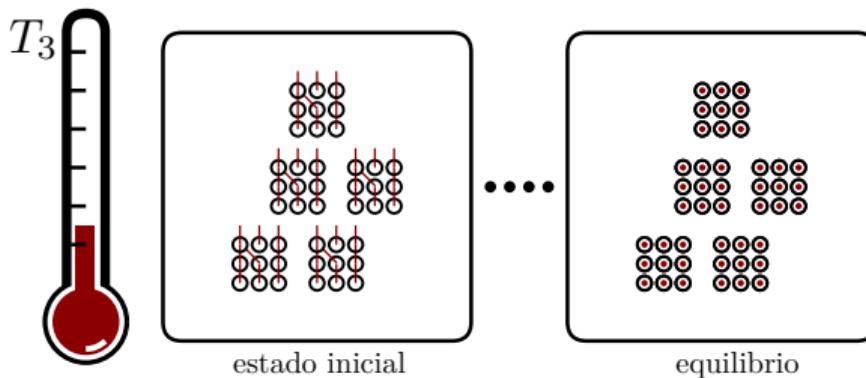
Simulated Annealing



Isso garante boas características mecânicas para o material.

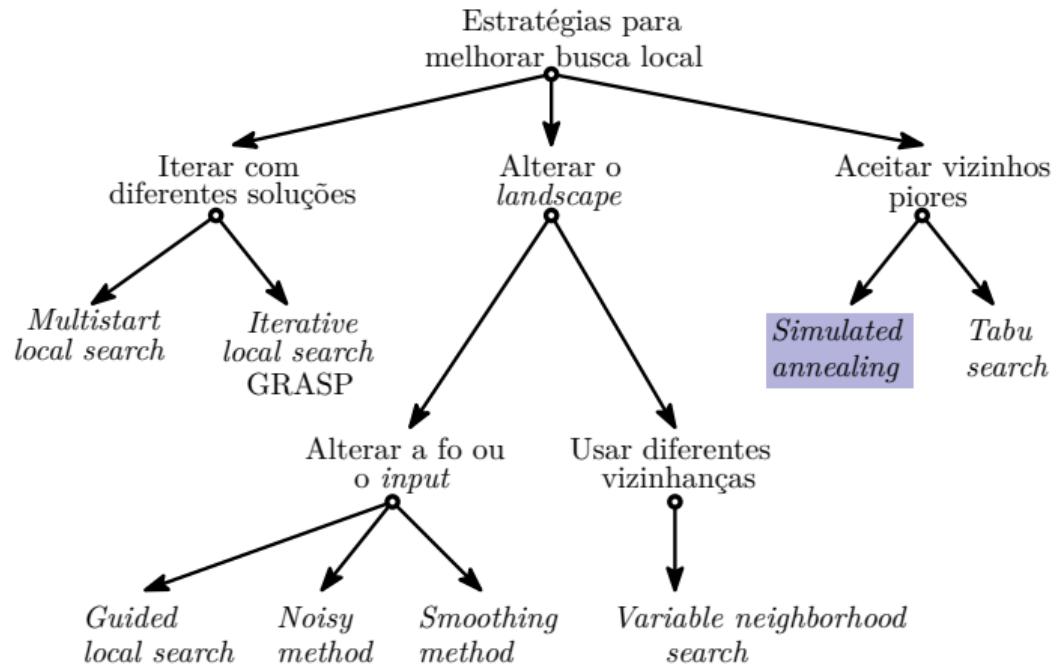
1. Conformabilidade
2. Ductibilidade

Simulated Annealing



Se a **temperatura** for reduzida de forma muito brusca, ou o estado de equilíbrio não for atingido, o material final pode apresentar imperfeições na sua estrutura atômica.

Formas de evitar ótimos locais



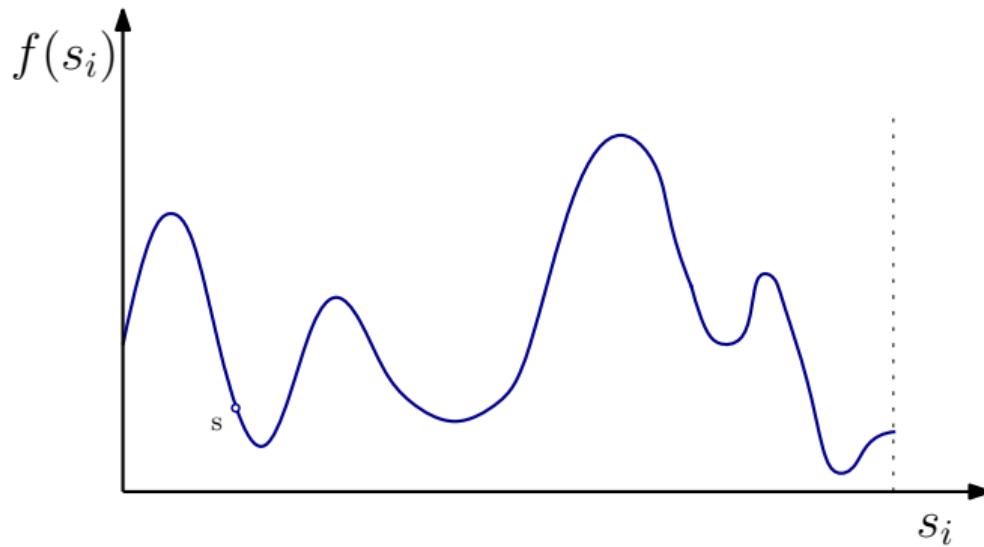
Como isso é usado no algoritmo?

A ideia de que altas temperaturas geram mais movimento nos átomos é usada para **evitar a convergência prematura em ótimos locais**. Com altas temperaturas existe uma probabilidade **P** alta de aceitarmos um movimento na vizinhança que não melhora a solução atual. Ao longo do algoritmo a temperatura vai abaixando e a probabilidade de aceitar uma solução pior é reduzida.

Se a redução de temperatura for muito acelerada, a busca ficará presa em um **ótimo local**, equivalentemente ao material que foi resfriado muito rapidamente e os átomos terminaram ficaram com **imperfeições** no final do processo de *Annealing*.

Simulated Annealing

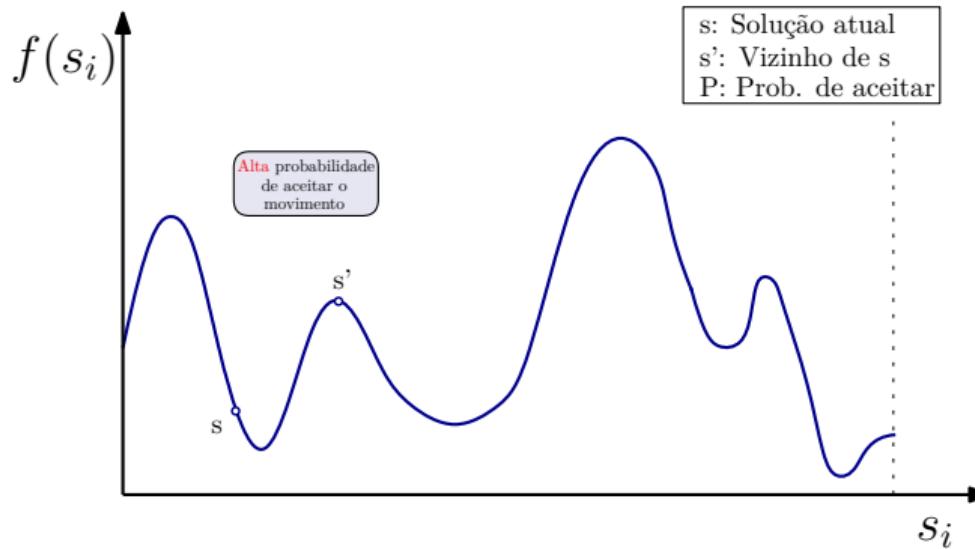
No *fitness landscape*



Considerando a solução s . O algoritmo gera uma solução vizinha de forma aleatória.

Simulated Annealing

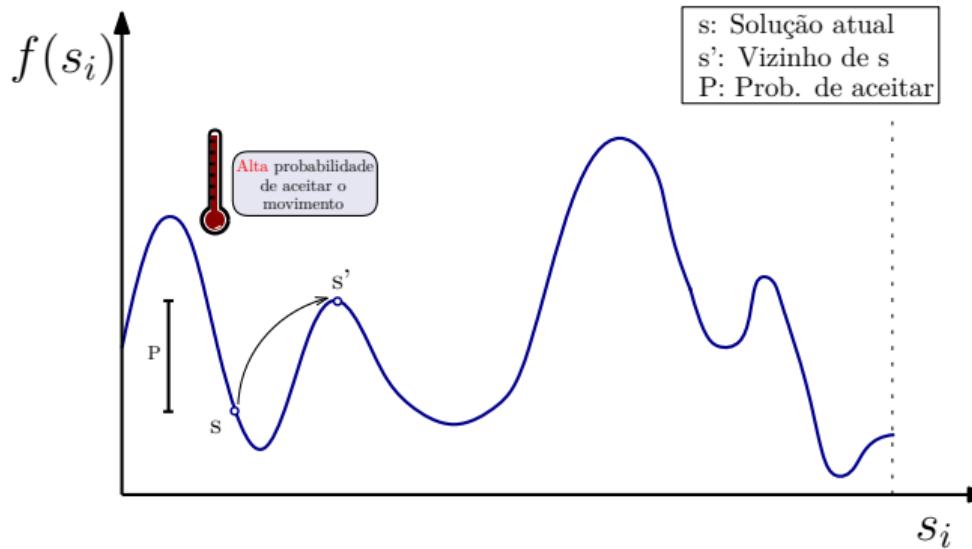
No *fitness landscape*



Quando a solução s' é pior do que a solução atual, calculamos uma probabilidade de aceitá-la mesmo assim.

Simulated Annealing

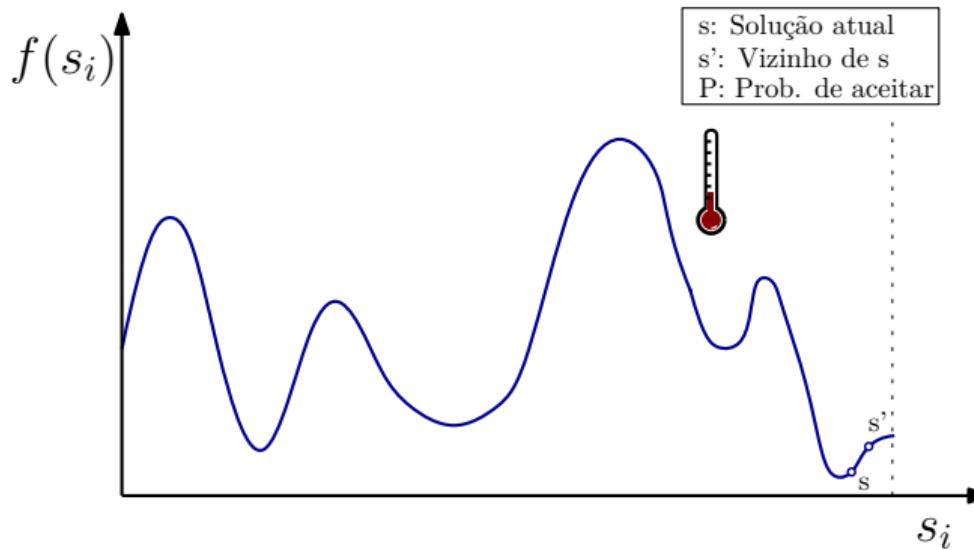
No *fitness landscape*



Essa probabilidade é influenciada pela temperatura (altas temperaturas aumentam a prob.), e pela diferença entre os valores objetivo ΔE .

Simulated Annealing

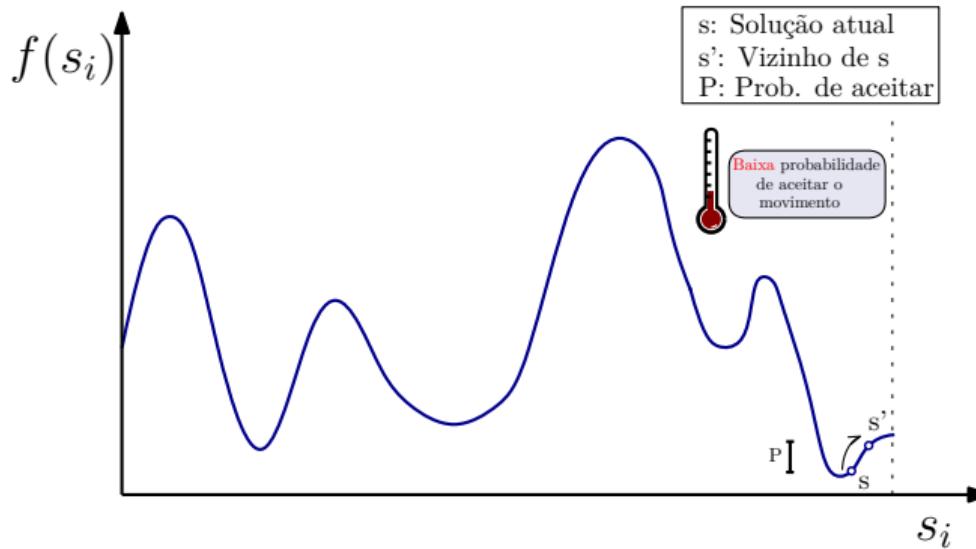
No *fitness landscape*



Já quando a temperatura é reduzida a probabilidade de aceitar um movimento pior é menor.

Simulated Annealing

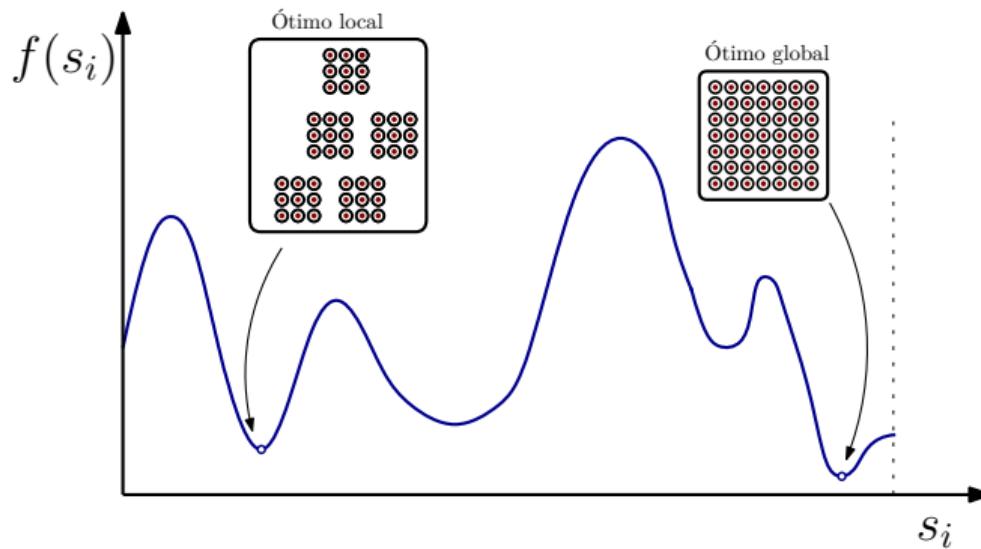
No *fitness landscape*



Já quando a temperatura é reduzida a probabilidade de aceitar um movimento pior é menor.

Simulated Annealing

No *fitness landscape*



Na comparação com o fenômeno físico, os ótimos locais são materiais com imperfeições e o ótimo global é o material com menor quantidade de energia e boas características.

Pseudocódigo

```
s = GeraSolucaoInicial()                                ▷ Gera uma solução inicial
T = Tmax
while T ≥ Tmin do
    while Equilibrio não atingido do
        Gera vizinho aleatório s'                      ▷ Gera vizinho aleatório de s
        ΔE = f(s') - f(s)
        if ΔE ≤ 0 then
            s = s'
        else
            Aceite s' com probabilidade P = e-ΔE/T
        end if
    end while
    T = RefriaTemperatura(T)                            ▷ Reduz a temperatura
end while
return melhor s.                                     ▷ Melhor s deve ser armazenada
```

Simulated Annealing

Questões de design

Para cada temperatura, a busca continua por um período determinado pelo equilíbrio. Além das questões comuns de *design* para as metaheurísticas de solução única (determinação da vizinhança e da solução inicial), as principais decisões do simulated annealing são:

1. **A função de probabilidade para o aceite:** Esse é o principal componente do SA que possibilita piores vizinhos de serem aceitos, mecanismo que tenta **evitar a convergência prematura para ótimos locais**.
2. **A função de resfriamento:** A função de resfriamento determina a temperatura a cada iteração do algoritmo. Têm papel crucial no sucesso do algoritmo.

Simulated Annealing

A função de probabilidade

Vejamos como a busca e seleção dos vizinhos funcionava no algoritmo de **busca local**.

Simulated Annealing

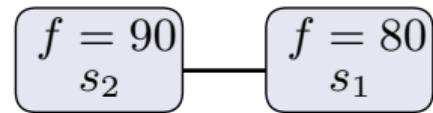
A função de probabilidade

$$\begin{array}{c} f = 80 \\ s_1 \end{array}$$

A partir de uma solução s_1 geramos sucessivos vizinhos.

Simulated Annealing

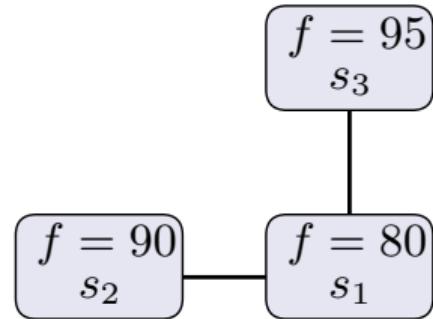
A função de probabilidade



A partir de uma solução s_1 geramos sucessivos vizinhos.

Simulated Annealing

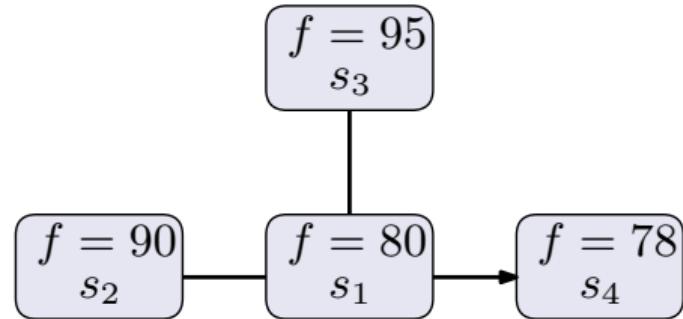
A função de probabilidade



A partir de uma solução s_1 geramos sucessivos vizinhos.

Simulated Annealing

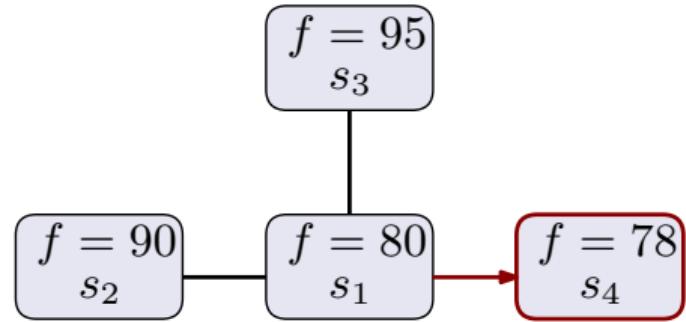
A função de probabilidade



Até o momento em que encontramos um **vizinho melhor** do que a solução atual.

Simulated Annealing

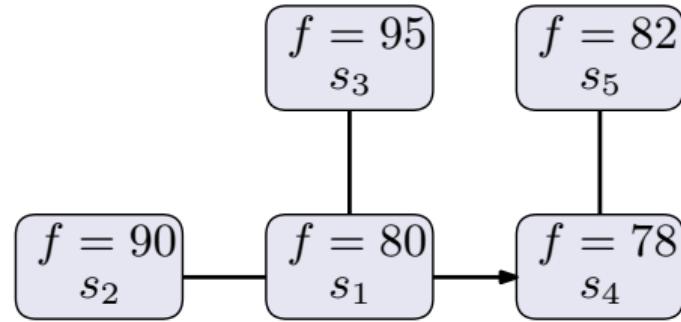
A função de probabilidade



Neste caso fazemos a substituição da solução atual pelo vizinho.

Simulated Annealing

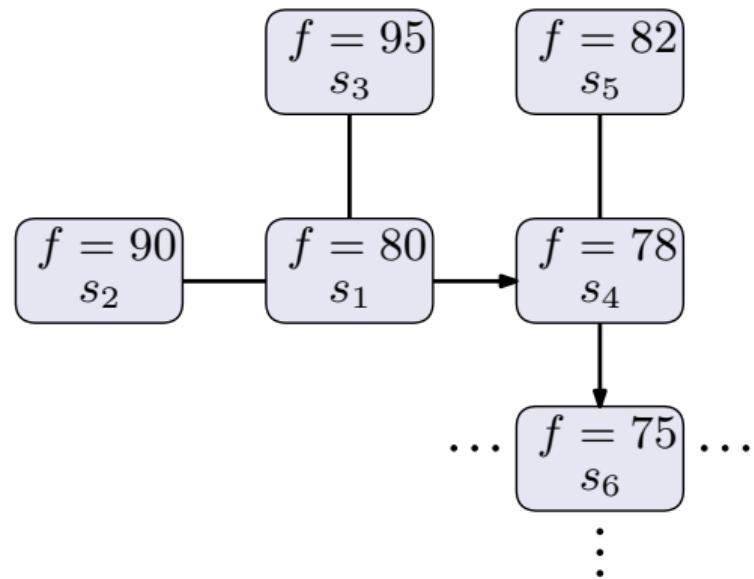
A função de probabilidade



E geramos os vizinhos da nova solução.

Simulated Annealing

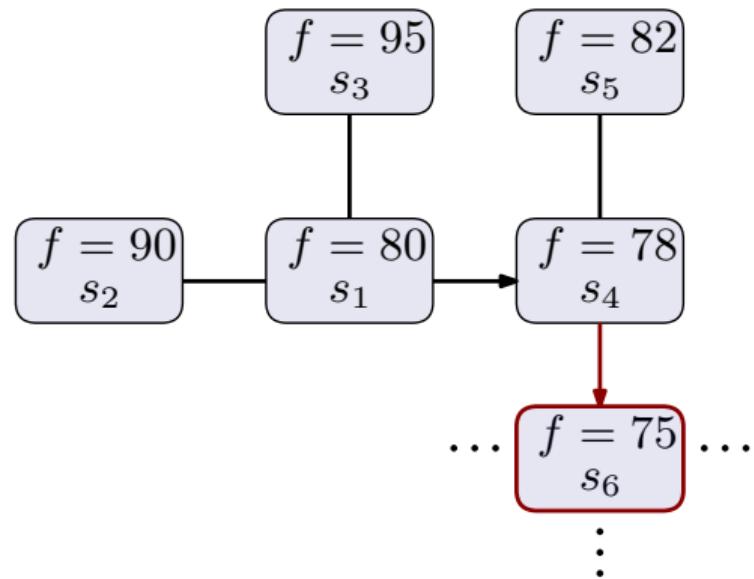
A função de probabilidade



E o processo continua até nenhum vizinho ser melhor do que a solução atual.

Simulated Annealing

A função de probabilidade



E o processo continua até nenhum vizinho ser melhor do que a solução atual.

Simulated Annealing

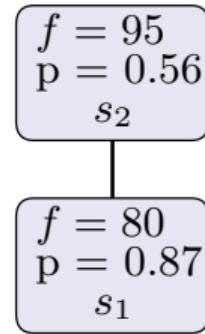
A função de probabilidade

$$\begin{aligned}f &= 80 \\ p &= 0.87 \\ s_1\end{aligned}$$

Já no SA, geramos os vizinhos de forma aleatória.

Simulated Annealing

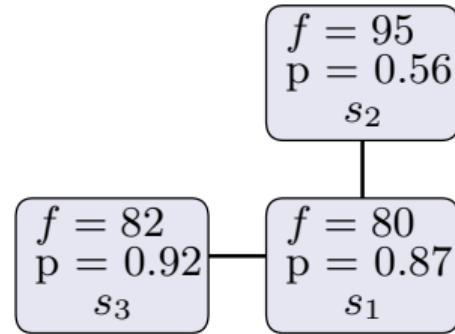
A função de probabilidade



E à cada vizinho calculamos uma probabilidade p de aceitar a solução.

Simulated Annealing

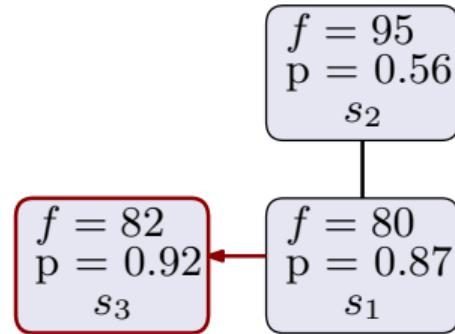
A função de probabilidade



Com essa probabilidade, mesmo soluções piores do que a atual podem ser aceitas.

Simulated Annealing

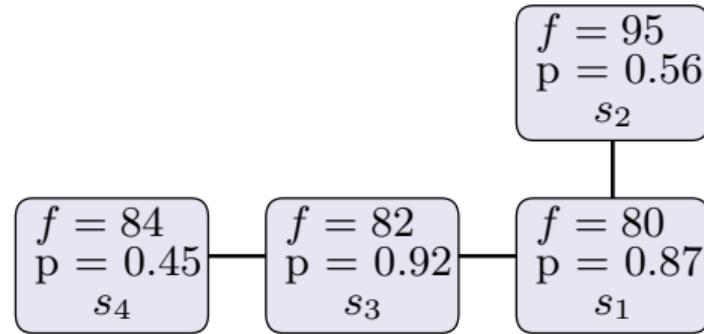
A função de probabilidade



Com essa probabilidade, mesmo soluções piores do que a atual podem ser aceitas.

Simulated Annealing

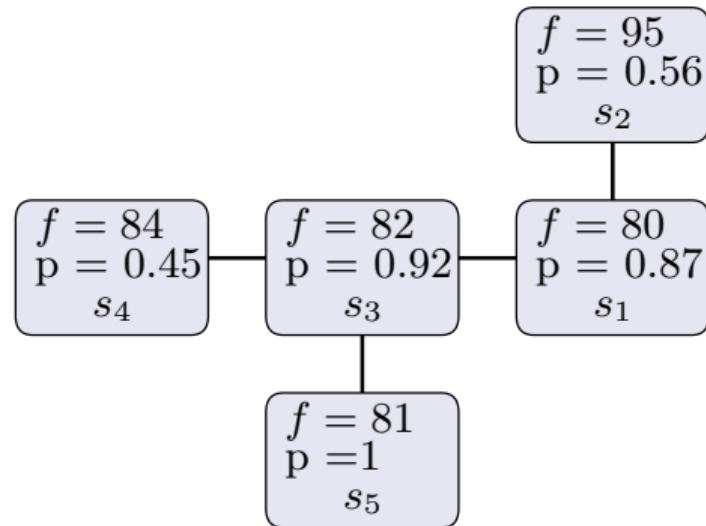
A função de probabilidade



Se a solução for aceita, da mesma forma que na busca local, continuamos gerando seus vizinhos.

Simulated Annealing

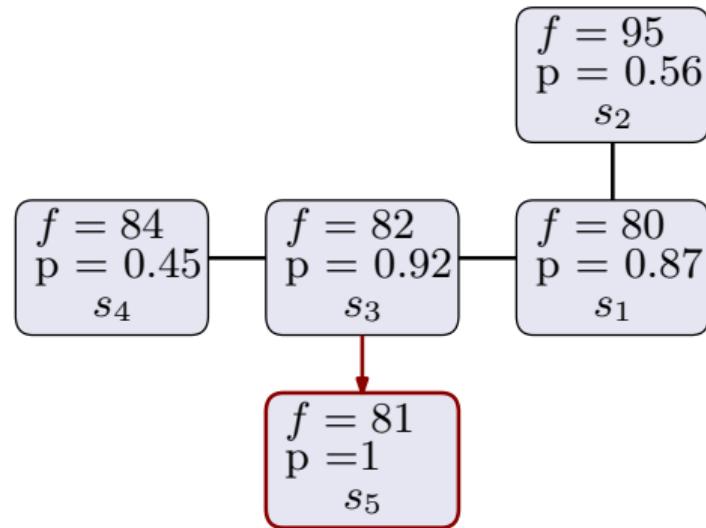
A função de probabilidade



Também da mesma forma que a busca local, se um vizinho é melhor que a solução atual, ele é aceito automaticamente (sem usar o mecanismo de probabilidade).

Simulated Annealing

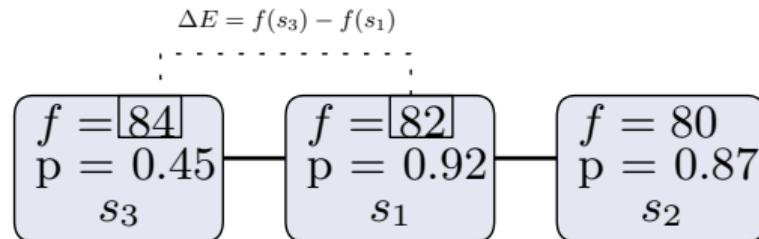
A função de probabilidade



Também da mesma forma que a busca local, se um vizinho é melhor que a solução atual, ele é aceito automaticamente (sem usar o mecanismo de probabilidade).

Simulated Annealing

A função de probabilidade



O cálculo da probabilidade deve ser uma função de 2 parâmetros:

1. **ΔE** : A diferença nos valores objetivos das duas soluções. Quanto maior ΔE , pior é o vizinho em relação a solução atual, de forma que a probabilidade de aceite deve ser menor.
2. **Temperatura**: Quanto maior a temperatura, maior deve ser a probabilidade de aceite de soluções piores.

Simulated Annealing

A função de probabilidade

Como estamos trabalhando com uma probabilidade $P(\Delta E, T)$, os axiomas da probabilidade devem ser satisfeitos. Seja:

$$P(\Delta E, T) = e^{\frac{-f(s') - f(s)}{T}} = e^{\frac{-\Delta E}{T}} \quad (1)$$

Mantido a uma temperatura T constante temos:

$$P(\Delta E) = e^{\frac{-f(s') - f(s)}{T}} = e^{\frac{-\Delta E}{T}} \quad (2)$$

Com os diferentes valores de $\Delta E = \{\Delta E_1, \Delta E_2, \dots\}$

1. $P(\Delta E) \geq 1$
2. $\sum_{\Delta E=-\infty}^{\Delta E=\infty} P(\Delta E) = 1$
3. $P(\Delta E_1 \cup \Delta E_2 \dots) = \sum_{\Delta E=-\infty}^{\Delta E=\infty} P(\Delta E)$

Simulated Annealing

A função de probabilidade

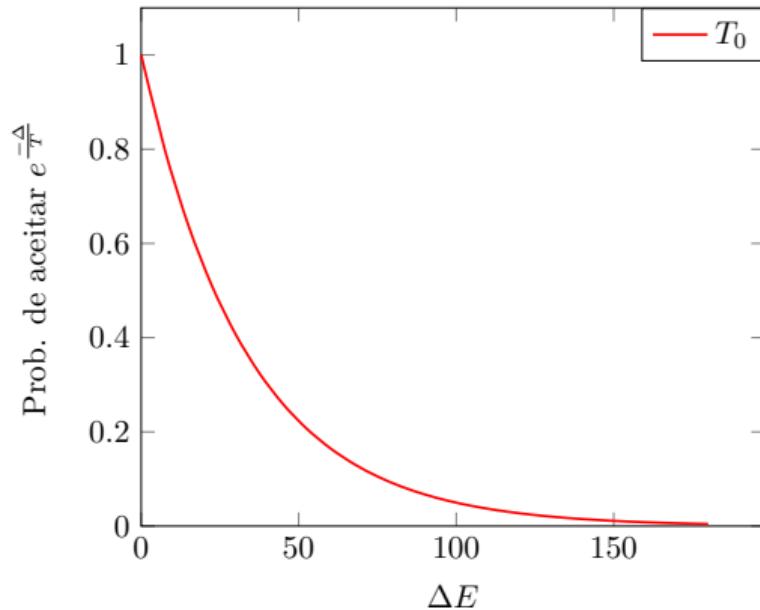
A função de probabilidade

$$P(\Delta E, T) = e^{\frac{-f(s') - f(s)}{T}} = e^{\frac{-\Delta}{T}} \quad (3)$$

é chamada de função de **distribuição de Boltzmann**, e obedece a todos os axiomas, bem como as relações que queremos entre probabilidade, temperatura e valores da função objetivo. Vejamos como a distribuição se comporta com diferentes valores de ΔE .

Simulated Annealing

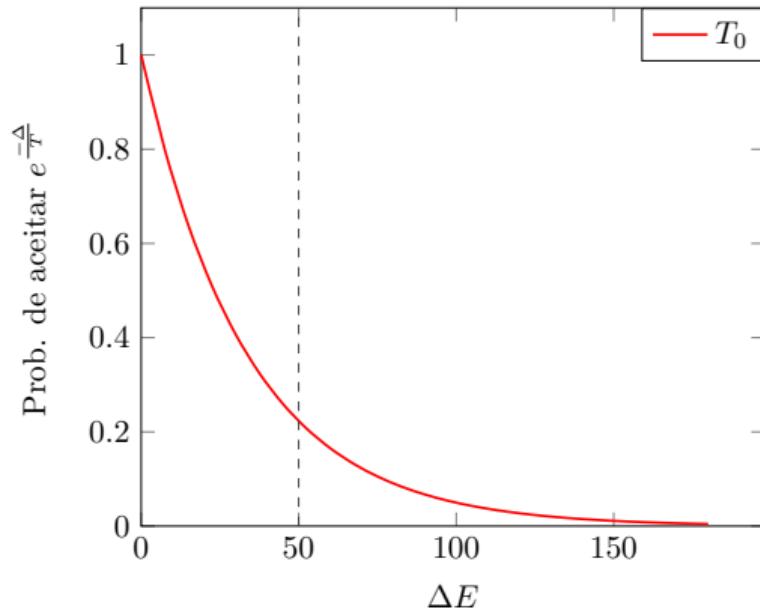
A função de probabilidade



Para uma determinada temperatura, a **distribuição de Boltzmann** tem o seguinte comportamento.

Simulated Annealing

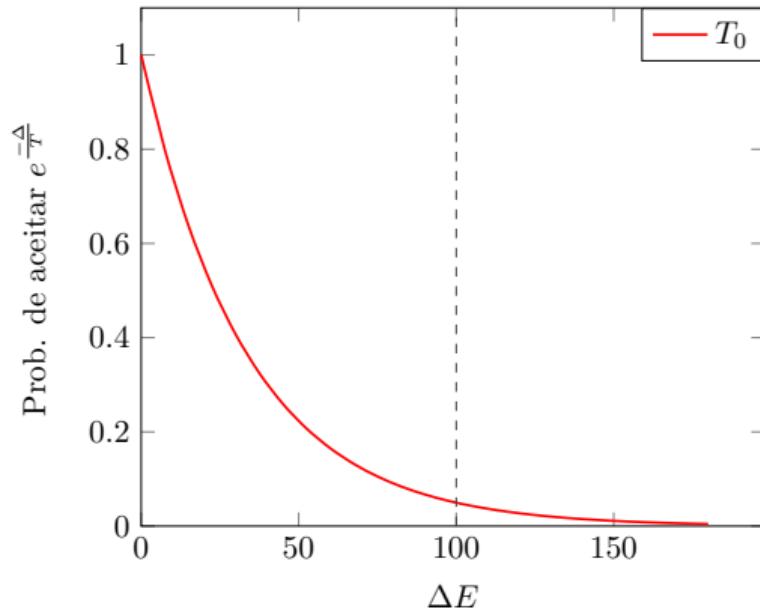
A função de probabilidade



Note que quando o ΔE é baixo (a solução vizinha é um pouco pior do que a atual) a probabilidade de aceite é alta.

Simulated Annealing

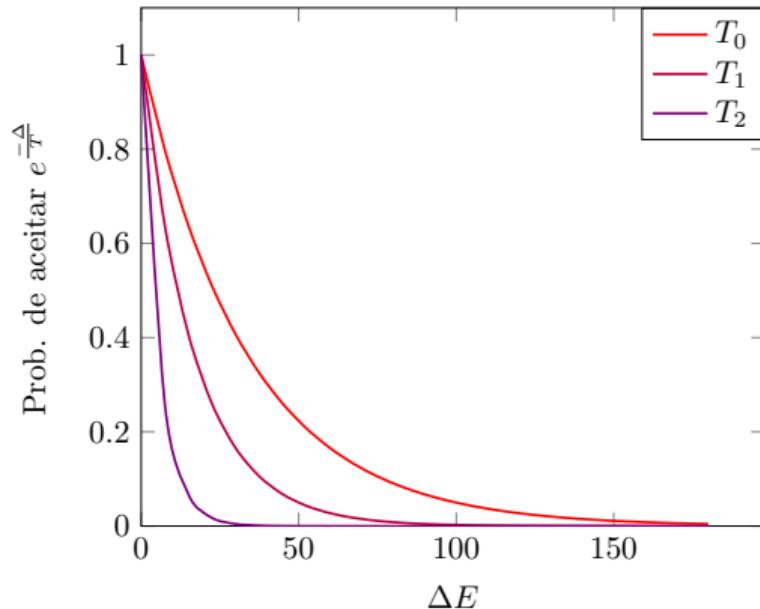
A função de probabilidade



Já quando ΔE aumenta a probabilidade de aceite diminui. Soluções um pouco piores do que a sol. atual são mais prováveis de serem aceitas do que as muito piores.

Simulated Annealing

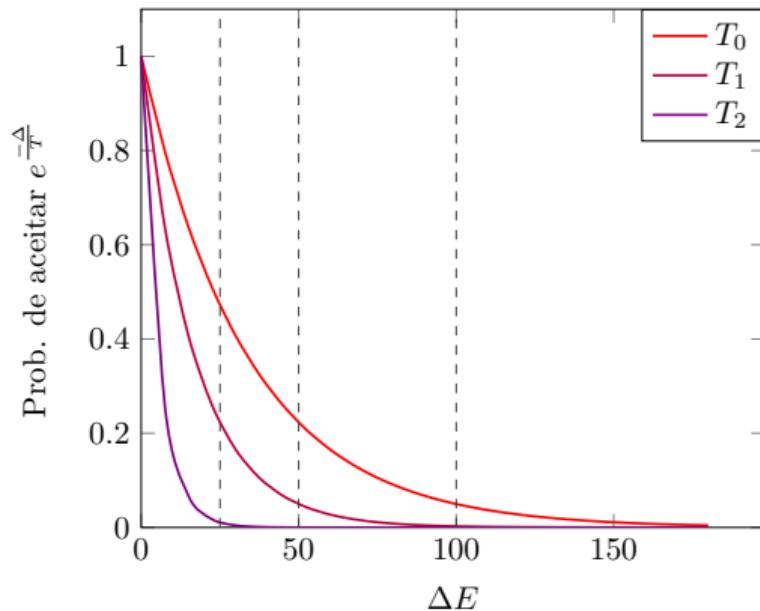
A função de probabilidade



Acima conseguimos ver a diferença da distribuição para diferentes temperaturas, com $T_0 > T_1 > T_2$.

Simulated Annealing

A função de probabilidade



Considerando o mesmo ΔE , para temperaturas maiores a probabilidade de aceite é maior.

Simulated Annealing

A função de probabilidade - conclusão

Conclusões

A distribuição de Boltzmann exibe o comportamento que precisamos (isso não impede que outras distribuições sejam utilizadas).

PERGUNTA: Como implementamos esse mecanismo de aceite/rejeição de uma determinada probabilidade?

Simulated Annealing

Esquema de resfriamento

Como a distribuição de probabilidade, existem muitos esquemas de resfriamento que podemos usar no algoritmo. As questões que devem ser definidas em um esquema são:

1. A temperatura inicial T_{max}
2. A temperatura final T_{min}
3. Estado de equilíbrio
4. Função de resfriamento da temperatura

Simulated Annealing

Esquema de resfriamento

Existem diversos estudos e divergências em relação aos 3 primeiros elementos, geralmente os cálculos levam em consideração o tamanho da instância. Por exemplo, problemas maiores precisam de mais iterações para chegar a um estado de equilíbrio, etc...de forma que vamos "testar" esses parâmetros..

Também existem diversas **funções de resfriamento da temperatura**, usaremos a **função geométrica**, em que a temperatura da próxima iteração T_{i+1} é calculada a partir da temperatura atual T_i com a seguinte fórmula:

$$T_{i+1} = T_i \alpha \quad (4)$$

Com $\alpha \in (0, 1)$. Testes mostram que α deve estar entre 0.5 e 0.99.

Simulated Annealing

Esquema de resfriamento

ATENÇÃO

O **ajuste dos parâmetros** (T_{max} , T_{min} , ...) do algoritmo é uma etapa crucial para o bom desempenho do SA. **Vá com calma!** Comece com valores que não gerem muitas iterações e aumente gradativamente, até encontrar o "equilíbrio" entre qualidade de soluções e tempo computacional despendido para encontrá-las.

Exemplo - TSP

EXEMPLO: Implemente um algoritmo SA para a resolução do TSP. Use as instâncias do TSPlib para verificar a sua eficácia. Lembre de rodar diversos testes para calibrar os parâmetros do algoritmo. Os resultados foram melhores ou piores do que a busca local? Salve todos em um arquivo de *benchmark* para comparação futura.

Atividade I

- 1 Qual é o problema dos algoritmos de busca local? Dê uma sugestão de como podemos melhorar o desempenho deles.
- 2 O que é o **paradigma intensificação x exploração** em metaheuristicas? Qual é o exemplo extremo de intensificação? E de exploração?
- 3 Explique a relação do algoritmo de *Simulated Annealing* com o processo de *Annealing* (recozimento) de materiais.
- 4 Aponte quais componentes do SA são responsáveis pela **intensificação** e pela **exploração** do *fitness landscape*.
- 5 Quais são os parâmetros do algoritmo SA? Como podemos determiná-los?

Atividade II

- 1 Considerando o seu problema escolhido para a disciplina. Encontre 2 artigos científicos que usam SA (ou usam SA como uma parte de outro algoritmo) para resolver o problema. Entenda o algoritmo proposto e identifique suas particularidades:
 - 1.1 Como a sol. inicial é construída?
 - 1.2 Qual é a vizinhança utilizada?
 - 1.3 Como são definidos os parâmetros do SA, função de resfriamento, prob., etc?
- 2 Implemente um algoritmo de SA para o seu problema escolhido.