

Provérbio Chinês:

“Escute e se esqueça, veja e se lembre, faça e compreenda”

“Hill Climbing” e “Simulated Annealing”

1. Resolução de Problemas via Métodos de Busca

- Sob a perspectiva de computação evolutiva a ser abordada neste curso, um *problema* pode ser entendido como uma coleção de informações a partir das quais algo deverá ser extraído ou inferido.
- Exemplos:
 - Função numérica a ser otimizada: $f(x) = x^3 + x + 1$
 - Sequenciamento de tarefas: dado um conjunto de máquinas, operadores, jornadas de trabalho, etc., qual configuração leva a uma melhor distribuição tarefa/operador?

- O processo de *resolução do problema* corresponderá a tomada de ações (passos), ou sequências de ações (passos), que levam a um desempenho desejado, ou melhoram o desempenho relativo de *soluções candidatas*.
- Este processo de procura por um desempenho desejado ou um melhor desempenho é denominado de *busca*.
- Um *algoritmo de busca* terá como entrada um problema e retornará como saída uma solução.
- Neste caso, uma ou mais soluções candidatas podem ser utilizadas no processo de busca. Os métodos que utilizam mais de um indivíduo no processo de busca são denominados de *métodos populacionais* (p.ex., algoritmos evolutivos).
- O primeiro passo na resolução de um problema é a *formulação do problema*, que irá depender das informações disponíveis.

PROBLEMA 1: Qual a idade dos meus três filhos? Informações disponíveis:

- Todos os meus 3 filhos fazem aniversário no mesmo dia.
 - O produto da idade deles é 36.
 - A soma da idade deles é igual ao número de janelas do prédio da esquina.
 - Meu filho mais velho possui olhos azuis.
- Três conceitos são fundamentais na resolução de problemas: *representação*, especificação de um *objetivo*, e definição de uma *função de avaliação* (Michalewicz & Fogel, 2000).
 - *Escolha de uma representação*
 - Corresponde a codificação de soluções candidatas, que por sua vez sofrerão algum tipo de manipulação. Sua interpretação irá implicar no *espaço de busca* e sua *dimensão*. O espaço de busca é definido pela sua configuração (estado) inicial e pelo conjunto de possíveis configurações (estados).

- *Especificação de um objetivo*
 - Descrição de uma meta. Trata-se de uma expressão (e não uma função) matemática que descreve o problema.
- *Definição de uma função de avaliação*
 - Função que retorna um valor específico indicando a qualidade (relativa) de uma solução candidata particular, dada a representação adotada. Trata-se geralmente de um mapeamento do espaço de soluções candidatas, dada a representação adotada, para um conjunto de números, onde cada elemento do espaço de soluções candidatas possui um valor numérico indicativo de sua qualidade. Geralmente o objetivo sugere uma função de avaliação particular.
- **Exemplo:** Suponha que seu objetivo seja maximizar a seguinte função de uma única variável: $g(x) = 2^{(-2((x-0.1)/0.9))^2} \text{sen}^6(5\pi x)$, $x \in [0,1]$ (Goldberg, 1989)

- *Representação*: suponha também que você irá representar a variável x utilizando uma cadeia binária de comprimento l . Sendo assim, o espaço de busca corresponde a todas as cadeias binárias de comprimento l , e portanto possui dimensão 2^l .
- *Objetivo*: $\max_x g(x), x \in [0,1]$.
- *Função de avaliação*: a avaliação da própria função serve para indicar a qualidade relativa dos candidatos a solução.

PROBLEMA 2: A escolha de um espaço de busca apropriado é fundamental. Seja o seguinte problema. Existem 06 palitos iguais em uma mesa e o objetivo é construir 04 triângulos equiláteros onde o comprimento de cada lado é igual ao comprimento de cada palito. Construa os triângulos.

1.1 Definição de um Problema de Busca

- Dado um espaço de busca S e uma região F factível deste espaço, $F \subseteq S$, encontre $x \in F$ tal que
$$eval(x^*) \leq eval(x), \forall x \in F$$
- Trata-se assim, de um problema de *minimização*, onde valores menores de x são considerados de qualidade superior.
- O ponto x^* que satisfaz a condição acima é dito ser o *ótimo global* ou *mínimo global* do problema.
- Ao contrário do ótimo global, uma solução $x \in F$ é dita ser um *ótimo local* em relação a uma vizinhança N de um ponto y , se e somente se
$$eval(x) \leq eval(y), \forall y \in N(x),$$
onde $N(x) = \{y \in S : dist(x,y) \leq \varepsilon\}$, $dist$ é uma função que determina a distância entre x e y , e ε é uma constante positiva.

- A função de avaliação define uma *superfície de resposta*, que será posteriormente denominada de *superfície de fitness* (ou *adaptação*), semelhante a uma topografia de vales e picos.
- Sendo assim, a *determinação de soluções ótimas para um problema corresponde a uma busca por picos (assumindo maximização) em uma superfície de adaptação*
- É importante ter em mente que esta superfície pode apresentar uma grande quantidade de picos, platôs, vales, etc., o que dificulta o processo de busca e a determinação de ótimos locais e globais.
- Métodos de busca eficientes devem ser capazes de fornecer um equilíbrio entre dois objetivos aparentemente conflitantes: busca local (*exploitation*) e exploração do espaço de busca (*exploration*).

2. Hill Climbing (Subida da Montanha)

- *Hill climbing* é um método de *busca (local)* que utiliza um procedimento de *melhora iterativa (iterative improvement)*.
- A estratégia é aplicada a um único ponto x (solução candidata) no espaço de busca

2.1 Hill Climbing Padrão

- Seu *algoritmo padrão* (ou *simples*) – *standard (simple) hill climbing* – pode ser descrito como segue:
 - Inicialize (aleatoriamente) o ponto x na região factível do problema.
 - A cada iteração, um novo ponto x' é selecionado aplicando-se uma pequena perturbação no ponto atual, ou seja, selecionando-se um ponto x' que esteja na vizinhança de x , $x' \in N(x)$.
 - **Se** este novo ponto apresenta um melhor valor para a função de avaliação, **então** o novo ponto torna-se o ponto atual.

- O método é terminado quando nenhuma melhora significativa é alcançada, um número fixo de iterações foi efetuado, ou um objetivo foi atingido.

PROBLEMA 3: Dada a descrição do algoritmo de subida da montanha simples, cite algumas de suas principais características sob o ponto de vista de um método de busca.

2.2 Hill Climbing Iterativo

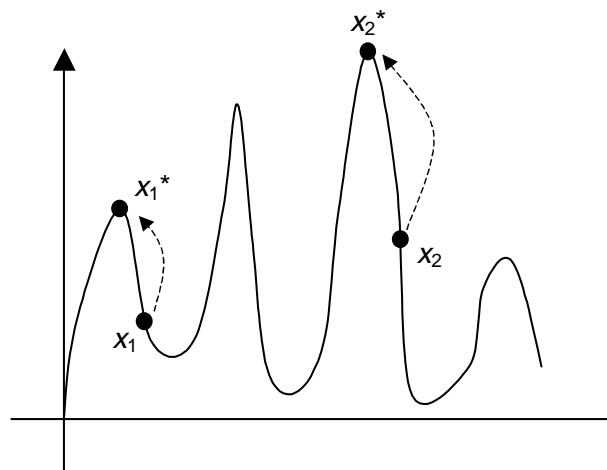
- Como pode ser verificado, o algoritmo de hill climbing simples realiza uma busca local em torno de seu ponto inicial.
- No intuito de aliviar esta limitação, o algoritmo também pode ser implementado automaticamente partindo-se de várias condições iniciais distintas e armazenando a melhor solução obtida; cria-se assim, uma espécie de memória para o hill climbing.

- Este procedimento é denominado de *hill climbing iterativo – iterated hill climbing*:
 - Inicialize a melhor solução encontrada até o momento (variável melhor)
 - Para cada condição inicial, faça
 - Inicialize (aleatoriamente) um ponto x na região factível do problema.
 - A cada iteração, um novo ponto x' é selecionado aplicando-se uma pequena perturbação no ponto atual, ou seja, selecionando-se um ponto x' que esteja na vizinhança de x , $x' \in N(x)$.
 - Se este novo ponto apresenta um melhor valor para a função de avaliação, então o novo ponto torna-se o ponto atual.
 - O método é terminado quando nenhuma melhora significativa é alcançada, um número fixo de iterações foi efetuado, ou um objetivo foi atingido.
 - Se x é melhor do que melhor, então $\text{melhor} \leftarrow x$.

2.3 Hill Climbing Estocástico (Probabilístico)

- O algoritmo padrão de hill climbing também pode ser modificado de forma a permitir uma escolha probabilística da solução candidata x' .
- Esta probabilidade depende da qualidade relativa entre x' e x , ou seja, da diferença entre os valores retornados pela função de avaliação quando aplicada a x' e x .
- Este procedimento é denominado de *hill climbing estocástico – stochastic hill climbing*:
 - Inicialize (aleatoriamente) o ponto x na região factível do problema.
 - A cada iteração, um novo ponto x' é selecionado aplicando-se uma pequena perturbação no ponto atual, ou seja, selecionando-se um ponto x' que esteja na vizinhança de x , $x' \in N(x)$.
 - Selecione x' com probabilidade $p = (1/(1+\exp[(eval(x)-eval(x'))/T]))$.
 - O método é terminado quando nenhuma melhora significativa é alcançada, um número fixo de iterações foi efetuado, ou um objetivo foi atingido.

- O parâmetro T é mantido fixo durante o processo de busca.



PROBLEMA 4: Atribua alguns valores para $eval(x)$ e $eval(x')$ e verifique a influência de T na probabilidade p de seleção do candidato x' . Sugestão: monte uma tabela relacionando os diversos elementos de p .

3. Simulated Annealing (Recozimento Simulado)

- O algoritmo de *simulated annealing* proposto por Kirkpatrick et al. (1983) foi inspirado pelo processo de *recozimento (annealing)* de sistemas físicos.
- O algoritmo de *simulated annealing* é baseado no procedimento de Metropolis et al. (1953) originalmente proposto como uma estratégia de determinação de estados (configurações) de equilíbrio de uma coleção de átomos a uma dada temperatura.
- As idéias básicas têm suas origens em *termodinâmica estatística*, ramo da física responsável por apresentar previsões teóricas sobre o comportamento de sistemas macroscópicos baseados nas leis que governam seus átomos.
- Kirkpatrick et al. (1983) perceberam que existe uma semelhança entre o procedimento de recozimento estudado com o algoritmo de Metropolis e processos de *otimização combinatorial*.

- Uma analogia com o recozimento de sólidos poderia fornecer uma estrutura para o desenvolvimento de um algoritmo genérico de otimização capaz de escapar de ótimos locais.
- O algoritmo de *simulated annealing* foi desenvolvido baseado em um procedimento utilizado para levar um material a seu estado de *equilíbrio máximo*, ou seja, a um estado de *energia mínima*.
 - Um determinado material é inicialmente aquecido a uma alta temperatura, de forma que ele derreta e seus átomos possam se mover com relativa liberdade.
 - A temperatura desta substância derretida é lentamente reduzida de forma que, a cada temperatura, os átomos possam se mover o suficiente para adotarem uma orientação mais estável.
 - Se a substância derretida for resfriada apropriadamente, seus átomos serão capazes de atingir um estado de equilíbrio máximo (energia mínima), produzindo um cristal.

- Caso contrário, um vidro ou outra substância (quebradiça) semelhante será produzida.
- Ao processo de aquecimento seguido de um resfriamento lento denominamos de *recozimento (annealing)*.

3.1 Princípios Básicos de Termodinâmica Estatística

- Em termodinâmica estatística, grandes sistemas a uma dada temperatura se aproximam espontaneamente de um estado de equilíbrio, caracterizado por um valor médio de energia que é função da temperatura (• erný, 1985).
 - A simulação da transição para um estado de equilíbrio e a redução da temperatura do sistema, permite a determinação de valores cada vez menores para a energia média do sistema.
- Seja x a configuração atual do sistema, $E(x)$ a energia de x , e T sua temperatura. É importante ressaltar que um estado de equilíbrio não é uma situação estática. Em

equilíbrio, o sistema muda aleatoriamente seu estado de uma configuração a outra de forma que a probabilidade de se encontrar o sistema em uma configuração particular é dada pela distribuição de Boltzmann-Gibbs:

$$P(x) = K \cdot \exp(-E(x)/T) \quad (1)$$

- Assumindo um sistema com uma quantidade discreta de possíveis estados (configurações), o cálculo numérico da energia média do sistema torna-se complicado.
- Entretanto, pode-se utilizar uma simulação de Monte Carlo, como a proposta por Metropolis et al. (1953) para as variações aleatórias de estado de uma configuração a outra do sistema, de forma que, em equilíbrio, a Equação (1) seja satisfeita.

Interpretação da Terminologia do Sistema Físico para o Domínio Computacional

Termodinâmica	Algoritmo de Simulated Annealing
Estado (configuração)	Solução (factível) do problema. Ponto no espaço de busca
Energia	Valor retornado pela função de avaliação
Estado de equilíbrio	Ótimo (local)
Estado de equilíbrio máximo	Ótimo global
Temperatura	Parâmetro de controle
Recozimento	Busca através da redução de T
Distribuição de Boltzmann-Gibbs	Probabilidade de seleção de um novo ponto

3.2 O Algoritmo de Simulated Annealing

- Assuma o problema genérico de minimizar uma função:
 - Inicialize (aleatoriamente) um ponto x na região factível do problema.
 - A cada iteração, um novo ponto x' é selecionado aplicando-se uma pequena perturbação no ponto atual, ou seja, selecionando-se um ponto x' que esteja na vizinhança de x , $x' \in N(x)$.
 - Determine a energia de cada uma destas configurações $E(x)$ e $E(x')$, e verifique a variação na energia do sistema: $\Delta E = E(x') - E(x)$.
 - **Se** $\Delta E \leq 0$, **então** x' torna-se o ponto atual ($x \leftarrow x'$); **senão**, a probabilidade de x' ser aceite como o ponto atual é dada por um caso particular da distribuição de Boltzmann-Gibbs, ou seja, $P(\Delta E) = \exp(-\Delta E/T)$.
 - O método é terminado quando nenhuma melhora significativa é alcançada, um número fixo de iterações foi efetuado, ou a temperatura T atingiu seu valor mínimo.

- O parâmetro T é iniciado com um valor elevado e é lentamente reduzido durante o processo de busca.
 - Geralmente implementa-se um decrescimento geométrico para T ; $T \leftarrow n.T$.
- A sequência de temperaturas e o número de rearranjos dos parâmetros até a chegada a um estado de equilíbrio a cada temperatura é denominado de *sequenciamento de recozimento (annealing schedule)* (Kirkpatrick et al., 1983).

EXERCÍCIO COMPUTACIONAL 1 (EC1): Dada a função de uma única variável

$g(x) = 2^{(-2((x-0.1)/0.9)^2)} \sin^6(5\pi x)$, apresentada na Figura 1, tente determinar seu ótimo global utilizando todos os quatro algoritmos apresentados (hill climbing padrão, hill climbing iterativo, hill climbing estocástico, e simulated annealing). A variável x está definida no intervalo $[0,1]$, $x \in [0,1]$.

Defina uma *representação*, o *objetivo* e uma *função de avaliação*. Dica: utilize como representação $x \in \mathfrak{R}$.

EXERCÍCIO COMPUTACIONAL 2 (EC2): Aplique os mesmos quatro métodos ao problema anterior, mas escolha como representação para a variável x uma cadeia binária m ($m = \langle m_l, \dots, m_2, m_1 \rangle$) de comprimento $l = 22$, correspondente a uma precisão de seis casas decimais.

- O mapeamento de uma cadeia binária para um número real pode ser feito em dois passos:

- Converta a cadeia binária $m = \langle m_l, \dots, m_2, m_1 \rangle$ da base 2 para a base 10:

$$\langle m_l, \dots, m_2, m_1 \rangle_2 = \left(\sum_{i=0}^{l-1} m_i \cdot 2^i \right)_{10} = x'$$

- Encontre o correspondente valor real (\mathfrak{R}) para x : $x = x_{\min} + x' \cdot \frac{x_{\max} - x_{\min}}{2^l - 1}$, onde

$$x_{\min} = 0 \text{ e } x_{\max} = 1, x \in [x_{\min}, x_{\max}] \text{ (domínio da variável).}$$

A função de avaliação é a mesma do problema anterior após a decodificação de m para seu valor real x como descrito acima.

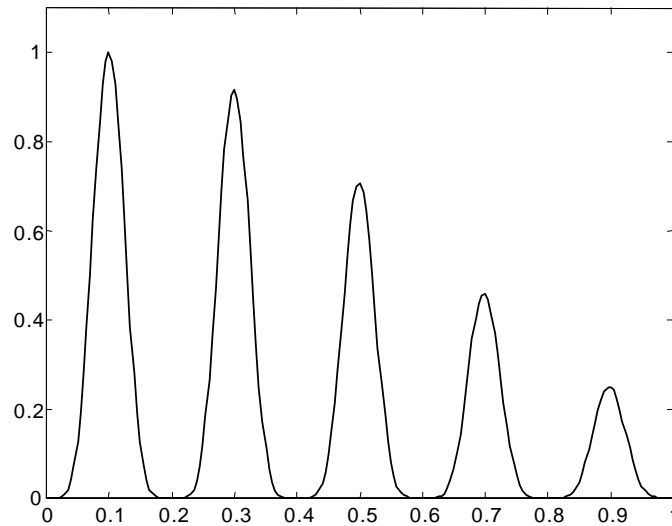


Figura 1: Função $g(x) = 2^{-(2((x-0.1)/0.9)^2)} \sin^6(5\pi x)$ a ser maximizada.

DISCUSSÃO: Discuta o desempenho individual e relativo dos métodos para cada representação.

4. Bibliografia Citada e Recomendada

- Aarts, E. & Korst, J. (1989), *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*, John Wiley & Sons, 1989.
- erný, V. (1985), “Thermodynamical Approach to the Travelling Salesman Problem: An Efficient Simulation Algorithm”, *J. of Optim. Theory and App.*, 45(1), pp. 41-51.
- Goldberg, D. E. (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Reading, Massachusetts.
- Kirkpatrick, S. (1984), “Optimization by Simulated Annealing - Quantitative Studies”, *J. Stat. Phys.*, 34, pp. 975-986.
- Kirkpatrick, S., Gelatt, C. D. Jr., & Vecchi, M. P. (1983), “Optimization by Simulated Annealing”, *Science*, 220, 671-680.
- Metropolis, N., Rosenbluth, A.W., Rosenbluth, M. N., Teller, A.H. and Teller, E. (1953), “Equations of State Calculations by Fast Computing Machines”, *J. Chem. Phys.*, 21, pp. 1087- 1092.
- Michalewicz, Z. & Fogel, D. B. (2000), *How To Solve It: Modern Heuristics*, Springer-Verlag, Berlin.
- Pincus, M. (1970), “A Monte Carlo Method for the Approximate Solution of Certain Types of Constrained Optimization Problems”, *Oper. Res.*, 18, pp. 1225-1228.
- Russell, S. J. & Norvig, P. (1995), *Artificial Intelligence A Modern Approach*, Prentice Hall, New Jersey, USA.

Material de apoio disponível na página do Prof. Fernando (www.dca.fee.unicamp.br/~vonzuben)