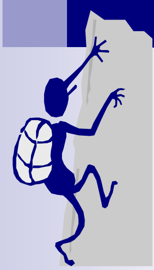
Métodos modernos de pesquisa e Optimização



Victor Lobo

Tópicos

- Introdução
- Métodos matemáticos “clássicos”
- Método de Monte Carlo
- Hill- Climbing
- Simulated Annealing
- Algoritmos Genéticos
- Tabu Search

Introdução

- Problema de optimização
  - Dada uma função
    f(x)
  - encontrar o seu óptimo (máximo ou mínimo)

Problema de pesquisa
- Seja um *ponto inicial*
- Encontrar o óptimo da função *f(x)*

Problema de pesquisa em paralelo
- Seja um conjunto de *pontos iniciais*
- Encontrar o óptimo da função *f(x)*

Propriedades de *f(x)*

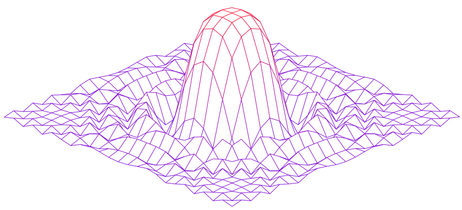
- Domínio
  - $\Re^n$
  - I^n
  - Sub conjunto de $\Re^n$ ou de I^n
  - Símbolos
- Propriedades de *f(x)*
  - Diferenciável
  - Não diferenciável

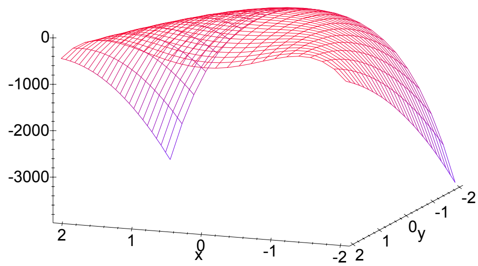
- Optimização Matemática
  - Gradiente
- Optimização com restrições
  - Multiplicadores de Lagrange
- Optimização Inteira
  - Investigação operacional
- Métodos heurísticos
  - Hill- Climbing
  - Simulated Annealing
  - Algoritmos Genéticos
  - Tabu Search

Exemplos

$$f(x,y)=\frac{\sin\left(x^2+y^2\right)}{\left(x^2+y^2\right)}$$

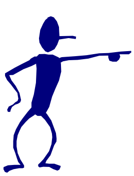
$$g(x,y)=-100\left(x^2-y\right)+(1-x)^2$$





Outros Exemplos

- Problema das N-Rainhas
  - Qual a função de optimização
  - Problema de satisfação de restrições



$$\begin{cases} X_i=\{1,\cdots,N\} \\ X_j \neq X_i \quad \forall j \neq i \\ X_j \neq X_i \pm (j-i) \quad \forall j > i \\ i,j \in \{1,\cdots,N\} \end{cases}$$

- "Assignment Problem"
  - Um conjunto de n pessoas é capaz de realizar n tarefas. O custo da pessoa i fazer a tarefa j é c_{ij}. Encontrar a atribuição de tarefas $(t_1,\ldots,t_n)$que minimize o custo

$$mínimo \sum_{i=1}^{n} c_{it_i}$$

Mais exemplos

"0-1 Knapsack problem"

Um conjunto de "n" itens deve ser empacotado numa mochila com capacidade de C unidades. Existem v_i unidades de cada item "i" e usa c_i unidades de capacidade. Determine o subconjunto I de items que podem ser empacotados de modo a maximizar

$$\text{máximo} \sum_{i \in I} v_i$$

tal que

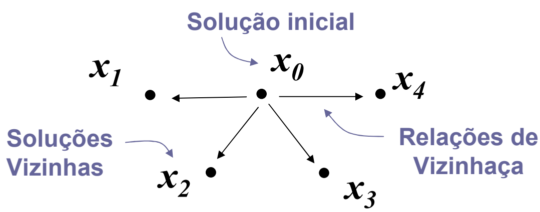
$$\sum_{i \in I} c_i \le C$$

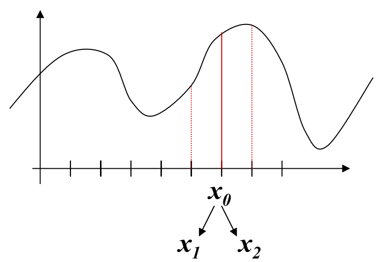
Mais outro exemplo

- Coloração de um grafo
  - Um grafo é definido por um conjunto de nós, alguns dos quais estão ligados entre si através de arcos. Dois nós ligados por um arco são designados por adjacentes. O problema da coloração de um grafo é atribuir cores a cada um dos nós de tal modo que dois nós adjacentes não tenham a mesma cor.
  - "O objectivo é encontrar o número mínimo de cores capazes de colorir um grafo."

Uma nova terminologia

- Estado → Solução
- Conjunto dos descendentes → Vizinhança
- Espaço de estados → Espaço de soluções





Codificação dos estados e operadores

- Domínios em $\Re^n$
  - Codificação: vector com um ponto em $\Re^n$
  - Cálculo dos descendentes
    - Orientada: Método do gradiente
    - Não orientada: Adicionar vector aleatório por exemplo gaussiano
- Domínios simbólicos
- Problema das N-rainhas
  - Exemplos de codificação
    - Vector de inteiros de 1 a N sem repetições
  - Exemplo do operador
    - Mudar duas das posições seleccionados aleatoriamente

Método do gradiente

- Seja uma função $f(x_1, \ldots, x_n)$ derivável.

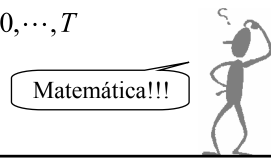
$$X = X_0 \pm \eta \nabla f(X)\big|_{X=X_0}$$

- O mínimo de $f(x_1, \ldots, x_n)$ é dado por

$$\begin{cases} x_i^{t+1} = x_i^t - \eta \dfrac{\partial f(X)}{\partial x_i}\Bigg|_{X = X^t} \\ i = 1,\cdots,n \wedge t = 0,\cdots,T \end{cases}$$

- O máximo de $f(x_1, \ldots, x_n)$ é dado por

$$\begin{cases} x_i^{t+1} = x_i^t + \eta \dfrac{\partial f(X)}{\partial x_i}\Bigg|_{X = X^t} \\ i = 1,\cdots,n \wedge t = 0,\cdots,T \end{cases}$$



Método do Gradiente

Problema: *maximizar f(X) em que f(X) é derivável*

Seleccionar uma solução inicial $X_0 \in \Re^n$

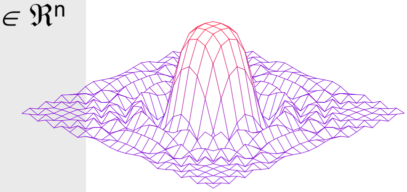
repita

$$X = X_0 + \eta \nabla f(X)\big|_{X=X_0}$$

se *f(X) > f(X_0)* **então** X_0 = *X*

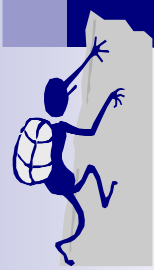
até *critério de paragem*

X_0 é a solução.



Hill-Climbing

(ou stochastic hill-climbing)



Hill-Climbing

Problema: maximizar f(s)

Seleccionar uma solução inicial $s_0 \in S$

repita

  Seleccionar aleatoriamente $s \in N(s_0)$ /* $N(s_0)$ é a vizinhança de s0 */

  se $f(s) > f(s_0)$

    então $s_0 =$ s; Contador = 0;

  senão Contador = Contador + 1

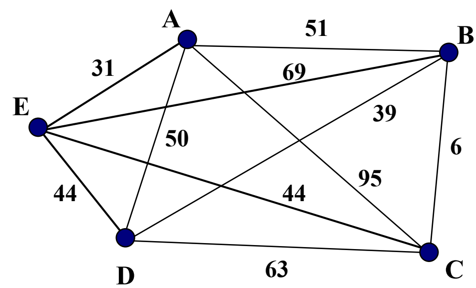
até critério de paragem

s_0 é a solução.

Exemplo do caixeiro viajante

- Consideremos o espaço de soluções representados pela sequência de 6 letras, em que só a primeira e a última são repetidas.
- O conjunto de vizinhança definida pela troca de duas letras
- Considere o ponto inicial

  ABCDEA



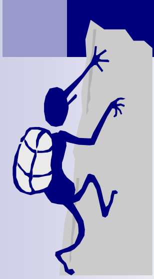
Solução

ADBCEA ou AECBDA

Problemas com o Hill-Climbing

- Pára nas seguintes situações
  - Máximos locais
  - Planaltos
  - Arestas.

Simulated Annealing



Kirkpatrick (1983)

"When optimising a very large system (i.e. a system with many degrees of freedom), instead of "always" going downhill, try to go downhill "most of the time".

Annealing

- Na física da matéria condensada refere-se como "annealing" o processo que se segue:
  - Um sólido num banho quente é aquecido, aumentando a temperatura até um valor máximo. A essa temperatura, todo o material encontra-se na fase líquida e as partículas arrumam-se aleatoriamente
  - A temperatura do banho quente é arrefecida suavemente, permitindo que todas as partículas se arrumem no estado de menor energia dessa estrutura.
  - Em Português:
    - Arrefecimento Simulado, Resfriado Simulado, ...

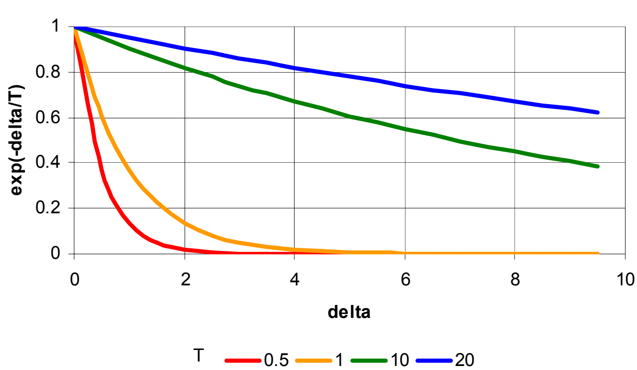
Algoritmo de Metropolis (1953)

- Desenvolvido para simular a evolução de um sistema físico quente que tende para o estado de equilíbrio térmico.
- Em cada passo do algoritmo, um átomo do sistema é sujeito a um pequeno deslocamento aleatório.
- Calcula-se a variação ΔE da energia do sistema.
  - Se ΔE < 0 o deslocamento é aceite. Se não o deslocamento só será aceite com uma probabilidade

$$p(\Delta E) = e^{-\frac{\Delta E}{T}}$$

  - onde T é a temperatura

Função exp(-δ/T)



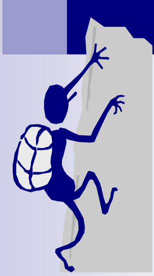
Simulated annealing

Problema: minimizar f(s)

Seleccionar uma solução inicial s0 $\in$ S; uma temperatura inicial T > 0; e uma função de redução de temperatura α

repita
    repita
        Seleccionar aleatoriamente s $\in$N(s0)   /* N(s0) é a vizinhança de s0 */
        δ = f(s) - f(s0)
        se δ < 0 então s0 = s; Contador = 0;
            senão
                seja x um um número aleatório entre 0 e 1
                se x < exp(- δ/T) então s0 = s ; Contador = Contador + 1
    até Contador = Nmax
    T = α(T)
até critério de paragem
s0 é a solução.

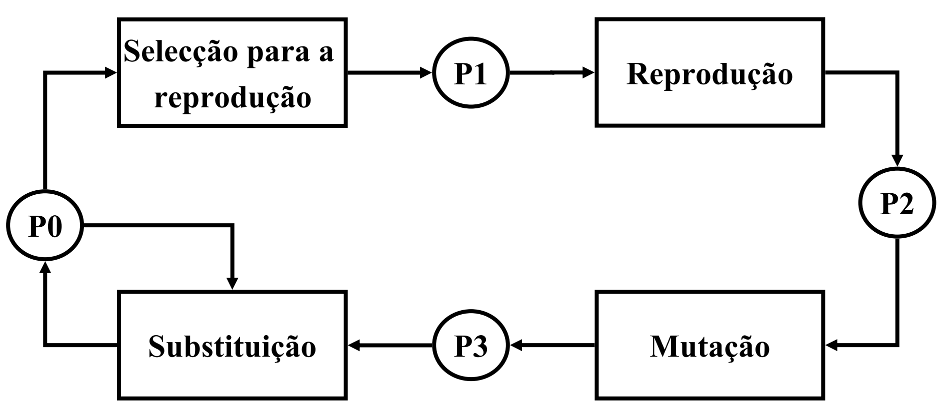
Algoritmos Genéticos



Algoritmos Genéticos

- Baseado na simulação da dinâmica de populações
- A pesquisa é baseada em populações
- Terminologia
  - *População - conjunto de descrições de indivíduos*
  - *Cromossoma - descrição de um indivíduo*
  - *Gene - Posição dentro do cromossoma*
  - *Alelo - Valor existente no gene*
  - *fitness - medida de adaptação do individuo ao meio ambiente*
- Parte da área de *"Evolutionary Computation"*

Esquema básico



Operadores

- Selecção para reprodução
  - Uniforme
  - Roleta
  - Integral
  - Torneio
- Reprodução ou Cross-over

  Depende do problema
  - Cruzamento 1 ponto
  - Cruzamento de n pontos
- Mutação

  Depende do problema
  - Inversão
  - Troca de dois genes
- Substituição
  - Completa
  - Parcial com selecção
    - *Uniforme*
    - *Roleta*
    - *Torneio*

Selecção para a reprodução

- A hipótese de um indivíduo ser seleccionado para a reprodução é função do seu fitness
- Roleta
  - Escolha aleatória e directamente proporcional ao seu fitness
- Integral
  - Respeita a muito rigidamente o fitness relativo
- Torneio
  - Dois indivíduos seleccionados aleatoriamente disputam um torneio. O melhor passa.

Um exemplo muito simples

- Encontrar o máximo da função $f(x) = x^2$ no domínio [0, 31]
- Qual a função de fitness?
  - $f(x)$
- Como codificar?
  - Utilizaremos uma codificação binária de 5 bits
- Exemplo de cromossomas
  - [00000] $\rightarrow$ x = 0
  - [01100] $\rightarrow$ x = 12
  - [11101] $\rightarrow$ x = 29

Processo de selecção para o exemplo (1ª geração)

Desc.	Crom.	X	$f(x)$	$\frac{f_i}{\sum f}$	$\frac{f_i}{\bar{f}}$	Selec.
1	01101	13	169	0.14	0.58	1
2	11000	24	576	0.49	1.97	2
3	01000	8	64	0.06	0.22	0
4	10011	19	361	0.31	1.23	1
Soma				1.00	4.0	
Média				0.25	1.0	
Máximo				0.49	1.97	

Cruzamento

- 1 Ponto de cruzamento
  - Sejam dois cromossomas de dimensão “N”. Seleccione-se aleatoriamente um ponto de corte do cromossoma (1…(N-1)). Cada um dos dois descendentes recebe informação genética de cada um dos pais

Exemplo

Cr 1 - 11101001

Cr 2 - 10101101

Seja o ponto de cruzamento 4

Cr 1 - 11101|001

Cr 2 - 10101|101

Descendentes

Desc 1 - 11101|101

Desc 2 - 10101|001

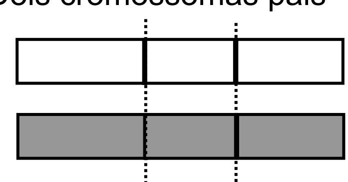
Cruzamento - Outro exemplo

- 2 pontos de cruzamento

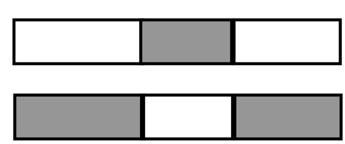
  Semelhante ao caso anterior mas agora com a escolha de dois pontos de corte

- Exemplo

  Dois cromossomas pais



  Dois descendentes



Processo de cruzamento para o exemplo (1ª para 2ª geração)

Cromos.	Cônjuge	Ponto de Cruzamento	Nova População	Valor x	f(x)
0 1 1 0 1	2	4	0 1 1 0 0	12	144
1 1 0 0 0	1	4	1 1 0 0 1	25	625
1 1 0 0 0	4	2	1 1 0 1 1	27	729
1 0 0 1 1	3	2	1 0 0 0 0	16	256
Soma					1754
Média					439
Máximo					729

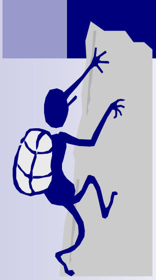
Processo de selecção para o exemplo (2ª geração)

Desc.	Crom.	X	*f(x)*	$\frac{f_i}{\sum f}$	$\frac{f_i}{\bar{f}}$	Selec.
1	0 1 1 0 0	12	144	0,08	0,33	0
2	1 1 0 0 1	25	625	0,36	1,43	1
3	1 1 0 1 1	27	729	0,42	1,66	2
4	1 0 0 0 0	16	256	0,15	0,58	1
Soma				1.00	4.0	
Média				0,25	1	
Máximo				0,42	1,66	

Efeito de passagem de uma geração para outra

- 1ª Geração:
  - 24, 19, 13, 8
- 2ª Geração
  - 27, 25, 16, 12
- .
- .
- n-ésima Geração (c/ mutação)
  - 31,31,30,27

Tabu Search

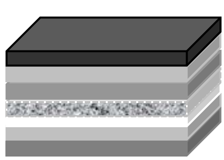


Tabu Search

- Algoritmo de pesquisa com um único ponto
- Tem memória
  - Memoriza os últimos movimento
  - Tabela de Tabu
- Proposto por Fred Glover (1986, 1990)

Exemplo do "Tabu Search"

- Pretende-se construir um módulo de material isolante composto por 7 camadas de diferentes materiais



- Codificação

2	5	7	3	4	6	1

- Operador de vizinhança
  - Trocar dois módulos entre si

2	5	7	3	4	6	1

Tabela de Tabu

	2	3	4	5	6	7
1						
2						
3						
4						
5						
6						

Iteração 0

2	5	7	3	4	6	1

f(X) = 10

2	4	7	3	5	6	1

f(X) = 16

Tabela de Tabu

	2	3	4	5	6	7
1						
2						
3						
4						
5						
6						

N(X)	Δf
5,4	6
7,4	4
3,6	2
2,3	0
4,1	-1

Iteração 1

2	4	7	3	5	6	1

f(X) = 16

2	4	7	1	5	6	3

f(X) = 18

Tabela de Tabu

	2	3	4	5	6	7
1						
2						
3						
4				3		
5						
6						

N(X)	Δf
3,1	2
2,3	1
3,6	-1
7,1	-2
6,1	-4

Iteração 2

2	4	7	1	5	6	3

f(X) = 18

4	2	7	1	5	6	3

f(X) = 14

Tabela de Tabu

	2	3	4	5	6	7
1		3				
2						
3						
4				2		
5						
6						

N(X)	Δf	
1,3	-2	T
2,4	-4	
7,6	-6	
4,5	-7	T
5,3	-9	

Iteração 3 - Aspiração

4	2	7	1	5	6	3

f(X) = 14

5	2	7	1	4	6	3

f(X) = 20

Tabela de Tabu

	2	3	4	5	6	7
1		2				
2			3			
3						
4				1		
5						
6						

N(X)	Δf	
4,5	6	T
5,3	2	
7,1	0	
1,3	-3	T
2,6	-6	

Iteração 4

5	2	7	1	4	6	3

f(X) = 20

Tabela de Tabu

	2	3	4	5	6	7
1		1				
2			2			
3						
4				3		
5						
6						

N(X)	Δf	
7,1	0	
4,3	-3	
6,3	-5	
5,4	-6	T
2,6	-8	

Bibliografia

- Colin R, Reeves, Modern Heuristic Techniques for Combinatorial Problems, McGraw-Hill
- David E. Goldberg, Genetic Algorithms in search Optimization & Machine Learning, Addison Wesley