

UM PROCESSO NÃO DETERMINÍSTICO NATURAL PARA OTIMIZAÇÃO DE FUNÇÕES

Amarildo de Vicente¹, Rafaela Greici da Motta Camicia²

1-Professor do Centro de Ciências Exatas e Tecnológicas da UNIOESTE; 2-Aluna do curso de especialização em Matemática Aplicada da UNIOESTE

Resumo - Este trabalho visa apresentar um algoritmo não determinístico, que está sendo denominado Árvore da Montanha, destinado à obtenção do ponto de mínimo global de uma função f não linear positiva. O algoritmo é baseado no comportamento de uma planta fictícia que agrega características de diversas outras plantas no seu processo de reprodução. Para diversas funções testadas ele apresentou um bom desempenho, produzindo resultados bastante satisfatórios.

Palavras-Chave: Otimização, algoritmo não determinístico, minimização.

A NATURE NONDETERMINISTIC PROCESS FOR FUNCTIONS OPTIMIZATION

Abstract- This work pretends to present a nondeterministic algorithm, that is being denominated Tree Mountain, destined to obtain the minimum global point of a positive nonlinear function. The algorithm is based on comportment of a fictitious plait that aggregates characteristics of several others plants in its reproduction process. For several functions tested it had presented a good performance, with the production of results very satisfactory.

KeyWord: Optimization, nondeterministic algorithm, minimization

1. INTRODUÇÃO

A programação não linear há bastante tempo é alvo de muitas pesquisas, principalmente no que diz respeito à consecução de algoritmos destinados a encontrar o ótimo global de uma função. Nesta busca surgiram vários algoritmos não determinísticos, como Algoritmos Genéticos (Vicente, 1999), *Ant Colony Optimization* (Dorigo, 1996), etc. Neste trabalho está sendo proposto um novo algoritmo não determinístico. Este algoritmo, que está sendo denominado Árvore da Montanha, é baseado no processo de renovação de uma planta que se desenvolve em regiões montanhosas. Esta planta atinge a idade de reprodução aos três anos e chega a cerca de 3 metros de altura, sendo que sua copa adquire cerca de dois metros de diâmetro. Durante sua florada o processo de polinização é feito por meio de insetos e, principalmente, pelo vento. O fato de o vento ser o principal agente polinizador se dá porque suas flores não são muito atrativas para a maioria dos insetos, pois possui

uma quantidade muito pequena de néctar e é pobre em coloração. Quando uma planta está em uma região alta, os grãos de pólen gerados por ela, em sua maioria, tendem a ser levados para as outras plantas que estão em partes mais baixas, ou ficam retidos na própria planta que os gerou. Evidentemente uma parte deles é conduzida para partes mais altas, trabalho que é basicamente feito pelos isentos. A Figura 1 ilustra o exposto.

Por causa destas características, as plantas mais novas, na sua maioria, nascem nas partes que estão abaixo de suas matrizes, fazendo com que sua população convirja para os vales.

A idéia do algoritmo é simular este comportamento para obter o ponto de mínimo global de uma função não linear positiva f .

Para implementar este algoritmo são necessários os conceitos matriz de distâncias e de soma de ponto com vetor, que serão apresentados a seguir. Estes conceitos são úteis para a produção

de um ponto aleatório em uma bola $B(O, r)$ qualquer do \mathbf{R}^n

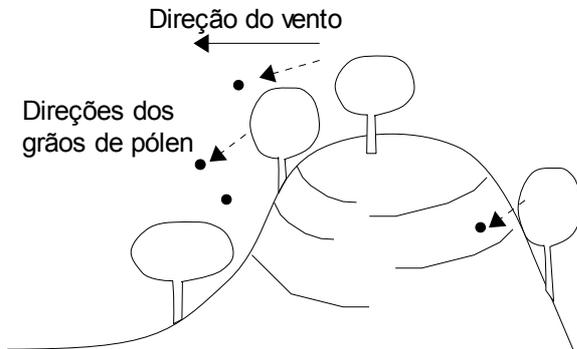


Figura 1 - Deslocamentos de grãos de pólen

Definição 1 - Matriz de distâncias

Dado um conjunto de pontos $P = \{X_1, X_2, \dots, X_n\}$, $X_i \in \mathbf{R}^m$, $i = 1, \dots, n$, chama-se matriz de distâncias de P a matriz $A = (a_{ij})_{n \times n}$ tal que $a_{ij} = d(X_i, X_j)$ se $i \neq j$ e $a_{ij} = 0$ se $i = j$, sendo d uma métrica escolhida.

Definição 2 - Soma de ponto com vetor

Seja P um ponto do \mathbf{R}^n e \vec{v} um vetor do \mathbf{V}^n . Um ponto Q é definido como soma de P com \vec{v} , que se indica por $Q = P + \vec{v}$, se o segmento \overrightarrow{PQ} é um representante de \vec{v} (ver Figura 2).

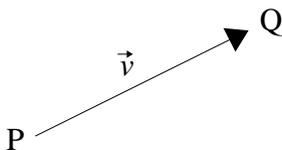


Figura 2 - Soma de ponto com vetor

2. GERAÇÃO DE PONTOS ALEATÓRIOS EM UMA BOLA DO \mathbf{R}^N

Seja $B(X_0, L)$ uma bola do \mathbf{R}^n , sendo $X_0 = (x_1, x_2, \dots, x_n)$ e $L > 0$. Consideremos um gerador de números aleatórios G que produz números em $[0, 1]$. Para gerar um ponto P em $B(X_0, L)$ pode-se proceder do seguinte modo:

- 1) Gere um ponto M utilizando o gerador G.
- 2) Para cada coordenada x_i de M aplique a função $f(x) = L(2x - 1)$, obtendo um ponto P. Esta função faz a dilatação do intervalo $[0, 1]$ para o intervalo $[-L, L]$.

- 3) Faça a soma $\overrightarrow{OX_0} + P$, sendo O a origem do \mathbf{R}^n , obtendo o ponto Q. Este ponto estará na bola $B(X_0, L)$.

Nota: Por conveniência computacional, neste trabalho será empregada a norma do máximo para determinar a bola B (Burden, 2005)

Por exemplo, considere no plano o ponto $X_0 = (4, 3)$ e $L = 1$ então $B(X_0, L)$ é o quadrado com centro em $(4, 3)$ e lados com 2 unidades de comprimento. Se o ponto gerado por G for $M = (0.2, 0.6)$ então, feita dilatação, tem-se $P = (-1.2, 0.4)$. Fazendo-se a soma de P com o vetor $\overrightarrow{OX_0}$ obtém-se $Q = (-1.2, 0.4) + (4, 3) = (2.8, 3.4)$.

3. DETALHAMENTO DO ALGORITMO

Suponhamos que se queira obter o ponto de mínimo de uma função f positiva em uma bola $B(X_0, L)$ do \mathbf{R}^n .

- 1) Gera-se inicialmente uma floresta contendo n árvores A_i em B, cada uma delas com os atributos centro (um ponto X_i de B onde a árvore A_i estará centrada) e altitude ($f(X_i)$).

2) Para cada par de árvores (A_i, A_j) calcula-se a distância $d(X_i, X_j)$ e constrói-se a matriz de distâncias $D = d[i,j] n \times n$. Na verdade o interesse é pela distância entre as copas, de onde os grãos de pólen irão sair ou chegar. Estas copas possuem aproximadamente a mesma altura, cerca de três metros, razão pela qual as distâncias entre elas podem ser avaliadas pelas distâncias entre seus centros. Evidentemente o trajeto seguido por pelos grãos de pólen serão os mais diversos possíveis, já que serão conduzidos pelo vento ou por insetos. Na impossibilidades de modelar estes percursos de um modo preciso é razoável que seja empregada a norma euclidiana para calcular tais distâncias.

- 3) A fim de gerar a próxima floresta, para cada uma das árvores existentes gera-se uma quantidade k de pólen. O valor de k deve servir como um parâmetro de controle, a ser ajustado experimentalmente de acordo com o problema que está sendo resolvido. Um grão de pólen de uma árvore A_i deve ser conduzido de forma probabilística a uma árvore A_j , de forma inversamente proporcional ao produto da altitude de A_j pela distância de A_i até A_j . Ele pode ainda ficar retido na própria árvore que o gerou, mas neste caso não se pode usar o processo anteriormente descrito já que haverá divisão por zero. Neste caso é preciso estipular uma certa probabilidade de que isto ocorra. Por exemplo, pode-se usar uma distribuição de probabilidade uniforme e estipular que a probabilidade de que um

grão de pólen fique retido seja de P%, $0 \leq P < 1$. Quando $P = 0$ nenhum grão de pólen ficará retido. No caso de $P = 1$ todos os grãos de pólen gerados ficarão retidos, o que não é interessante pois prejudicaria a convergência do algoritmo.

4) Uma vez gerados nk grãos de pólen (número de árvores pelo número de grãos gerados em cada uma delas), e feita a distribuição deles, supõe-se a produção de sementes. Não são necessárias variáveis para representar frutos e nem sementes. Pode-se supor que cada grão de pólen se transformará em uma semente e que esta irá germinar. Assim, se uma árvore A_i recebeu M pólenes durante a distribuição, geram-se M novas plantas em torno desta árvore, ou seja, em uma bola $B(X_i, r)$, $r > 0$. Estas novas plantas irão atingir a fase adulta após três anos, quando terão cerca de 3 metros de altura. Portanto, é necessário simular a passagem destes três anos para que seja feita a comparação entre as árvores geradoras com as árvores geradas. Neste intervalo de tempo novas plantas serão geradas, mas não serão investigadas. Aqui deve-se supor que o crescimento de todas as plantas é uniforme. Para que o processo convirja, e também por limitações computacionais, entre as nk árvores de interesse geradas e as árvores que as geraram deve-se selecionar apenas as n melhores para compor a nova floresta de observação. Esta nova floresta será portanto possivelmente composta por plantas da floresta mais recente e por plantas da floresta que as gerou (matrizes).

5) Nesta fase faz-se um teste de parada. Este teste pode ser o alcance de um número máximo de gerações pré-estabelecido. Pode-se também optar por parar quando todas as árvores da floresta tiverem se concentrado próximo de um único ponto, onde se supõe ser ponto de mínimo de f . Isto deve ocorrer quando a distância entre todos os pares de árvores forem menor que um valor $\varepsilon > 0$ estipulado, o que pode ser observado calculando-se a norma da matriz de distâncias.

A descrição anterior pode ser resumida por meio do pseudocódigo a seguir. Nesta abordagem será fixada em 0 a probabilidade de um grão de pólen ficar retido na própria árvore que o gerou.

Dados de entrada

- função real f positiva definida no R^n ;
- NA: número de árvores;
- NP: número máximo de grãos de pólen gerados por árvore;
- $B(X_0, R)$: bola contida no R^n (região de busca);
- NG: número de gerações;

Variáveis

- RA: raio de busca em torno de uma árvore;
- V_a : vetor de dimensão NA, em que V_{a_i} é a altitude da árvore A_i ;
- V_p : vetor de dimensão NA, em que V_{p_i} é a quantidade de grãos de pólen gerada pela árvore A_i ;
- X_i : Vetor de dimensão n , $i = 1, \dots, NA$, que representa o centro da árvore A_i ;
- D: matriz $NA \times NA$ (matriz de distâncias);
- z: fator de redução do raio de busca em torno de uma árvore ($0 < z < 1$);

- 1) Gerar NA árvores A_i distintas em B, com centro em X_i ;
- 2) Faça $V_{a_i} = f(X_i)$ e $V_{p_i} = 0$;
- 3) Obtenha a matriz de distâncias D;
- 4) Obtenha a matriz V_s tal que $V_{s_{ij}} = 1/(V_{a_i} * D_{ij})$, $i = 1, \dots, NA$, $j = 1, \dots, NA$, $i \neq j$;
- 5) Faça $V_{p_i} = \text{random}(NP)$, $i = 1, \dots, NA$ (um número inteiro aleatório de 0 a NP);
- 6) Para cada grão de pólen da árvore A_i , $i = 1, \dots, NA$, sorteie uma árvore $j \neq i$ pela regra da roleta,

com base na soma $S = \sum_{k=1}^{NA} S_{i,k}$, ($k \neq i$);

- 7) Contabilize em V_{p_i} o total de grãos de pólen recebidos pela árvore A_i , $i = 1, \dots, NA$;
- 8) Para cada árvore A_i , $i = 1, \dots, NA$, gere V_{p_i} árvores na bola $B(X_i, RA)$;
- 9) Dentre as novas árvores recém produzidas e as anteriores, escolha as NA melhores para formar a nova floresta e descarte as demais.
- 10) Se a árvore de menor altitude desta nova floresta for idêntica à anterior, reduza o raio de busca de RA para zRA .
- 11) Repita os passos 2 até 10 até que o número de gerações tiver sido atingido. Tome como solução o centro da árvore de menor altitude da última floresta.

4. ILUSTRAÇÃO DO PROCESSO

Para ilustrar o funcionamento do algoritmo considere-se a função teste dada por $f(x, y) = (x - y^2)^2 + (x - 1)^2 + 2$, cujo valor mínimo é 2 e ocorre no ponto $X = (1, 1)$. Para região de busca tomou-se a bola $B[(0, 0), 10]$ (norma euclidiana) e como configuração para os parâmetros tomou-se 5 árvores, 10 grãos de pólen gerados por planta e um total de 40 gerações. O quadro 1 a seguir mostra os pontos onde a população inicial de árvores foi produzida, as 5 primeiras populações geradas a partir destas bem como as 7 últimas. Para geração das árvores (pontos aleatórios), utilizou-se o gerador do Pascal com $randseed = 5$.

Gera- ção	Centros das árvores					Maior altitude
	A ₁	A ₂	A ₃	A ₄	A ₅	
1	(0.69, 0.54)	(1.39, 2.07)	(0.34, -2.86)	(-3.43, -3.20)	(-2.33, 4.07)	368.0923
2	(0.69, 0.54)	(0.72, 0.32)	(1.34, 0.49)	(0.91, 1.45)	(0.54, -0.41)	4.1299
3	(0.69, 0.54)	(1.25, 0.67)	(0.43, -0.02)	(0.91, 1.45)	(0.64, -0.18)	3.7623
4	(0.29, 0.50)	(0.85, 0.60)	(0.45, 0.07)	(0.25, 1.16)	(0.64, -0.18)	3.7623
5	(0.40, 0.85)	(0.85, 0.60)	(-0.01, 0.04)	(0.25, 1.16)	(0.89, 0.02)	3.7460
6	(0.52, 0.90)	(0.39, 0.71)	(0.80, 1.17)	(0.25, 1.16)	(0.89, 0.02)	3.7460
34	(1.01, 1.00)	(1.00, 1.00)	(1.02, 1.01)	(1.00, .01)	(1.00, 1.01)	2.0003
35	(1.01, 1.00)	(1.01, 1.01)	(1.00, 1.00)	(1.00, 1.01)	(1.01, 1.00)	2.0001
36	(1.01, 1.00)	(1.01, 1.01)	(1.00, 1.00)	(1.001, .01)	(1.01, 1.00)	2.0001
37	(1.01, 1.00)	(1.01, 1.01)	(1.00, 1.00)	(1.01, 1.01)	(1.00, 1.01)	2.0001
38	(1.01, 1.00)	(1.01, 1.00)	(1.00, 1.00)	(1.01, 1.01)	(1.00, 1.01)	2.0001
39	(1.01, 1.00)	(1.01, 1.00)	(1.01, 1.01)	(1.01, 1.01)	(1.01, 1.00)	2.0000
40	(1.01, 1.00)	(1.01, 1.00)	(1.01, 1.00)	(1.01, 1.00)	(1.01, 1.00)	2.0000

O gráfico a seguir mostra a evolução das altitudes das árvores mais altas, da segunda até à vigésima primeira geração.

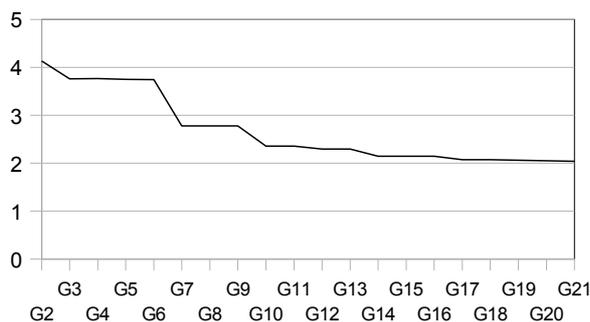


Figura 3 - Altitudes das árvores mais altas

5. ANÁLISE DOS RESULTADOS

Como se pode observar no Quadro 1, a primeira população está distante do ponto de mínimo. À medida em que as novas gerações vão sendo produzidas, as árvores vão nascendo cada vez mais próximo deste ponto. Na 39ª iteração este ponto é alcançado com uma precisão de 4 casas decimais, apesar da reduzida quantidade de árvores consideradas. No gráfico da Figura 3 pode-se observar como os piores casos vão se aproximando do valor ótimo com tendência à estabilização. Nas últimas gerações todas as árvores tendem a ficar muito próximas umas das outras, o que caracteriza a convergência. Evidentemente mesmo esta solução poderia ser obtida com menos iterações, bastando fazer uma escolha mais apropriada dos parâmetros, como por exemplo, maior número de árvores.

6. CONCLUSÃO

Neste trabalho procurou-se fazer uma abordagem do algoritmo Árvore da Montanha, que ainda se encontra em fase inicial de pesquisa e desenvolvimentismo. Porém, testes realizados com funções mais complexas mostraram que ele possui um bom funcionamento. Seu ponto fraco é a lentidão de convergência quando a população se aproxima do ponto ótimo, conforme pode ser observado no gráfico apresentado. Todavia, a expectativa é que ele possa ser melhorado, tornando-se mais eficiente neste aspecto.

REFERÊNCIAS

- BURDEN, R. L. et al. **Análise Numérica**, Ed. Thonson Learning, 2003.
- DORIGO, M. et al. **The Ant System: Optimization by a Colony of Cooperating Agents**. Institute of Electrical and Electronics Engineers. IEEE Transactions on Systems, Man and Cybernetics Part B, vol. 26, n. 1, p.1-13, 1996. Disponível em <http://iridia.ulb.ac.be/~mdorigo/ACO/publications.html>. Acesso em 04 de junho de 2009.

VICENTE, A. **Um Modelo Matemático para a Estruturação de um Sistema de Produção Agrícola Integrado**. Tese (Doutorado) - Universidade Federal de Santa Catarina, Florianópolis, 1999.