ANÁLISE DE OTIMIZAÇÃO REALIZADA EM PARÂMETROS GEOMORFOLÓGICOS UTILIZANDO A PLATAFORMA DE DESENVOLVIMENTO *MATHEMATICA 8.0*

Jonathan Alves Rebouças [1] ;Soraia Arruda Cipriano[2]; Laldiane de Souza Pinheiro[3]& Cristian Epifânio de Toledo[4]

RESUMO – Os processos de otimização são parte fundamental da modelagem de sistemas físicos. As plataformas de desenvolvimento como *mathematica, matlab e maple* tem se mostrado de grande utilidade nesses processos, combinando em suas linguagens métodos gráficos e analíticos de estudo. Esse trabalho mostra os diferentes métodos de otimização da plataforma de desenvolvimento *mathematica 8.0* aplicados a um modelo de análise geomorfológica. Dentre os métodos utilizados com a plataforma de desenvolvimento foi possível escolher um que mostrou o melhor ajuste em detrimento dos demais. Dentre os métodos, aquele que mostrou a melhor a convergência foi o método *Nealder-Mead*.

ABSTRACT – The optimization processes are a fundamental part of the modeling of physical systems. The development plat forms such as *mathematica*, *matlab* and *maple* has proved very useful in these processes, combining in their languages analytical and graphical methods of study. This work shows the different methods of optimization of the development platform *mathematica 8.0* applied to a model of geomorphological analysis. Among the methods used for the development platform it was possible to choose one that showed the best fit to the detriment of others. Among the methods, that which showed the best convergence was the method *Nealder-Mead*.

Palavras-Chave – *mathematica 8.0*, otimização, modelagem.

INTRODUÇÃO

O processo de otimização ou calibração corresponde ao ajuste de parâmetros de um modelo matemático de tal forma que esse modelo venha a satisfazer um objetivo. O objetivo visado pelo modelo pode estar relacionado a coisas como sustentabilidade, produtividade ou verossimilhança a um fenômeno físico.

A acurácia do modelo é medida através de uma função denominada *função objetivo*. Essa função indica o quão bom foram os ajustes realizados nos parâmetros do modelo. A otimização pode ser realizada através de duas formas: manual ou automática. A forma manual corresponde ao

[1])Departamento de Engenharia Agrícola-UFC: Campus do Pici - Bloco 804 – Fortaleza-CE:, (85) 3366 9754, (85) 3366 9755, jonalvesreboucas@bol.com.br
[2])SaneController engenharia e meio ambiente, Rua Nunes Valente, 674 – Meireles/Fortaleza-CE, fone: (85) 3261.5664, e-mail: soraiacipriano@hotmail.com
[3]) Departamento de Engenharia Agrícola-UFC: Campus do Pici - Bloco 804 – Fortaleza-CE:, fone: (85) 3366.9757, fax: (85) 3366 9755, e-mail: laldiane.pinheiro@gmail.com
[4])Departamento de Engenharia Agrícola-UFC: Campus do Pici - Bloco 804 – Fortaleza-CE:, fone: (85) 3366.9757, fax: (85) 3366 9755, e-mail: cristianepifanio@yahoo.com.br

método básico de tentativa e erro, onde os ajustes ocorrem com base na experimentação e podem ser tão difíceis quanto menor for a sensibilidade do pesquisador ao fenômeno que está por trás do modelo. A forma automática de otimização se utiliza de algoritmos que buscam os parâmetros ótimos do modelo norteados por uma função objetivo.

Com a evolução das ciências computacionais a sofisticação dos métodos de otimização, bem como suas performances, vem sendo melhoradas a cada dia. Linguagens de programação como FORTRAN, PASCAL, COBOL e C ++ vem perdendo espaço para plataformas de desenvolvimento como o *mathematica, matlab e maple.* Essas plataformas diminuem consideravelmente o trabalho do programador devido as suas extensas bibliotecas que graças ao intenso uso nos dias atuais vem crescendo gradualmente. Essas plataformas possuem linguagens próprias, bem como interface com linguagens consagradas como FORTRAN e C++. Essas interfaces são dedicadas aos programadores antigos que podem se beneficiar da praticidade dessas plataformas sem abrir mão de suas linguagens preferidas. Outro atrativo dessas plataformas é a área de desenvolvimento visual dos cálculos. Note que, enquanto nas linguagens antigas o programador possuía apenas uma visão local do fenômeno devido aos resultados não serem seguidos de visualização gráfica, nas novas linguagens já é possível ter uma visão geral do objeto de estudo através de gráficos mostrados pelas próprias plataformas, bem como traçar todo o trajeto seguido pelo algoritmo de forma visual.

A plataforma de desenvolvimento *mathematica 8.0* foi concebida por Stephen Wolfram em meados dos anos 80. Uma de suas principais vantagens é sua linguagem própria que permite a combinação de programação imperativa, funcional, orientação ao objeto e simbólica. Seu uso em diversas áreas do conhecimento já vem de longa data seguindo uma trajetória de sucesso (Fateman (1992); Johnson e Moore (1993); Cavallini (1993); Zotos (2007)).

O uso de otimização em modelos hidrológicos já é algo a muito estudado. Esse processo, muitas vezes, se faz necessário devido a impossibilidade de uma parametrização física do modelo.

A geomorfologia de bacias hidrográficas e reservatórios fazem parte do processo de modelagem hidrológica. Conhecer como os parâmetros geomorfológicos influenciam os processos hidrológicos é algo que pode ser crucial no bom funcionamento do modelo. Os parâmetros geomorfológicos são os responsáveis por fazer uma descrição matemática da estrutura física de uma bacia hidrográfica ou reservatório. Esses parâmetros são encontrados em vários modelos que relacionam forma do reservatório, encaixe no vale, declividade da bacia hidrográfica e etc. Os modelos que descrevem a geomorfologia de reservatórios, geralmente têm como dados de saída o volume de água do reservatório ou o espelho da bacia hidráulica. Essas variáveis de saída tem efeito direto no balanço hidrológico da bacia hidrográfica e por sua vez na modelagem hidrológica.

Este trabalho tem como objetivo analisar os processos de otimização executados pelo software *mathematica 8.0* para parâmetros geomorfológicos com base em dados obtidos através de uma batimetria realizada no açude do Cambito, localizado na bacia do Banabuiú, município de Quixeramobim-CE.

DESCRIÇÃO DA ÁREA DE ESTUDO

O açude cujos parâmetros geomorfológicos foram analisados está localizado na Bacia do Banabuiú, mais precisamente no Município de Quixeramobim-CE, (Figura **1**). O açude possui uma bacia hidráulica máxima de 19,42 ha. Essa área foi obtida através da vetorização de uma imagem LANDSAT 5 (resolução 30 m x 30m) datada do dia 24/07/2004, período em que o açude se encontrava próximo do seu ponto de sangria. Esse dado parece consistente uma vez que BURTE (2008) estimou a área máxima do espelho d'água em 18 ha. O açude possui uma profundidade máxima de 9,10 m e um volume máximo de 6,23. 10^5 m³. Sua bacia de contribuição possui uma área com 1.421 ha. No lado direito da Figura **1** é mostrado a malha de pontos obtida durante a batimetria do reservatório.

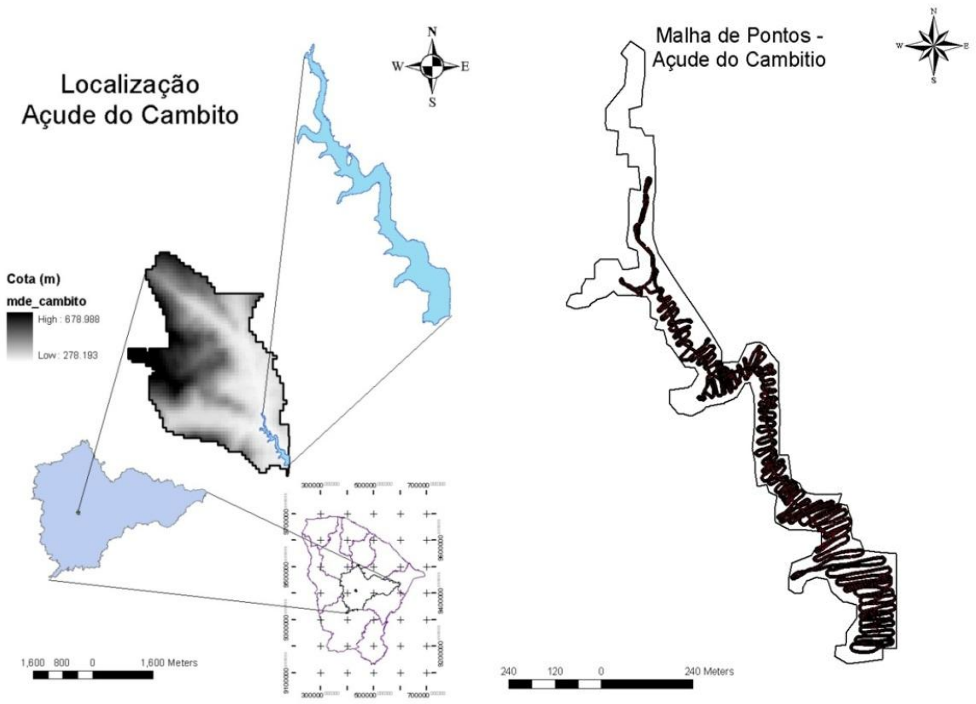

Figura 1- A esquerda é mostrado o açude do cambito, localizado na bacia do Banabuiú-CE. A direita é mostrado a malha de pontos obtida durante a batimetria no reservatório.

METODOLOGIA

A batimetria no Açude foi realizada entre os dias 22/05/2012 e 23/05/2012. Durante o trabalho de batimetria foram coletados 7354 pontos, o que constitui uma malha com uma densidade de 378,69 pontos/ha. A partir da batimetria realizada foi gerada uma curva cota x área x volume (CAV) para o reservatório, Figura **2**. Os parâmetros geomorfológicos utilizados no estudo foram o coeficiente de forma do açude 'a' e o coeficiente de abertura do vale 'K'. Esses parâmetros foram propostos por Molle e Cadier (1992). Os parâmetros se relacionam através da equação (1) para a relação profundidade-volume (V-H) e pela equação (2) para a relação profundidade-superfície (S-H):

$$V = K.H^{a} \quad (1)$$

$$A = a.K.H^{(a-1)} \quad (2)$$

na equação (1) V é o volume do reservatório e H é a profundidade obtida através da diferença entre a cota atual e a cota mais baixa do açude. Na equação (2), A corresponde a área da bacia hidráulica para uma dada cota.

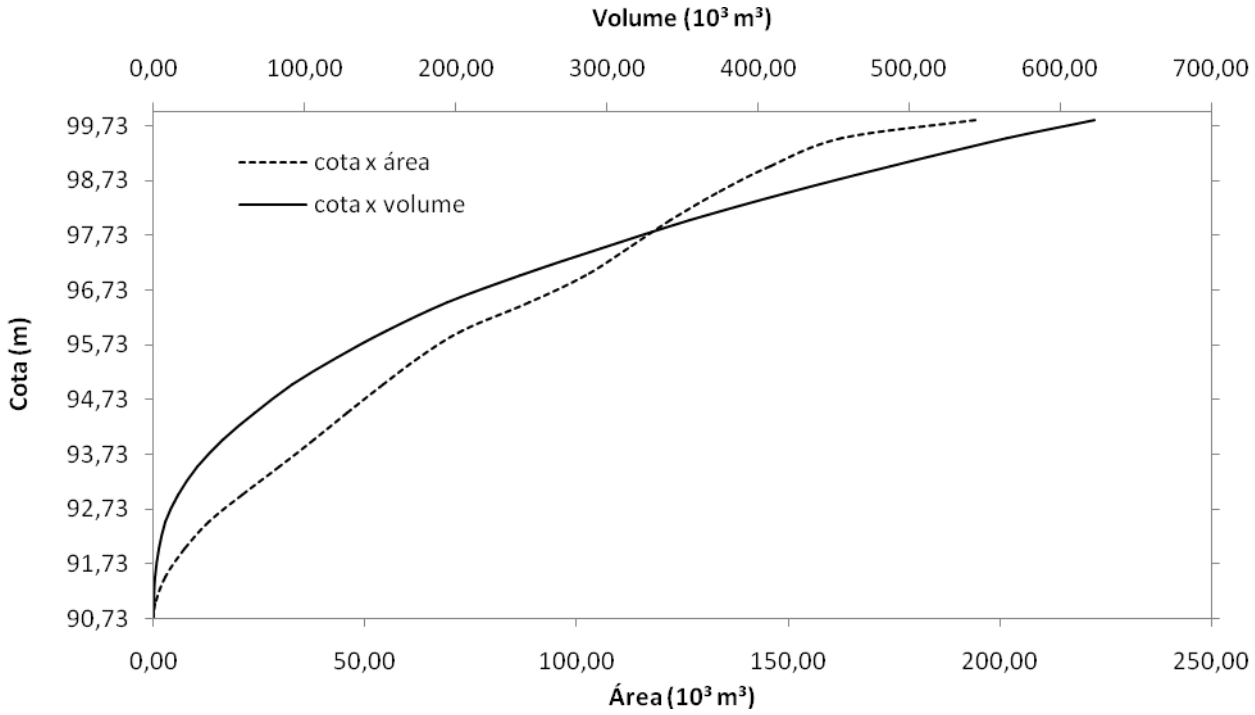

Figura 2 - Curva cota-área-volume montada para o reservatório do Cambito.

A função objetivo utilizada para o processo de otimização foi a função proposta por Nash e Sutcliffe (1970), equação (3). Essa função tem seu valor máximo igual a 1, significando ajuste perfeito. Portanto, o processo de otimização será de maximização.

$$NSE = 1 - \frac{\sum_{i=1}^{n}\left(O_i - P_i\right)^2}{\sum_{i=1}^{n}\left(O_i - O_{avg}\right)^2} \quad (3)$$

Na equação (3) NSE é o valor da função objetivo, Oi é o valor medido, Pi é o valor estimado e Oavg é a média dos valores medidos.

Para o processo de otimização foi utilizada a plataforma de desenvolvimento *mathematica 8.0*. Essa ferramenta conta com quatro métodos de busca heurísticos de máximos e mínimos globais: *Differential Evolution* (Storn e Price (1997)), *Nelder-Mead* (Luersen e Riche Le (2004)), *Random Search* (Jeżowski *et al*. (2005)) *e Simulated Annealing* (Kirkpatrick *et al*. (1983)). Foi realizado um ajuste dos parâmetros para cada um dos métodos. Também foi realizada uma análise gráfica dos passos seguidos por cada algoritmo verificando qual possuía uma forma de busca mais adensada e objetiva. Foi observado durante o processo de otimização que ao se ajustar os parâmetros utilizando a equação (1) os valores obtidos de 'a' e 'K' eram diferentes dos que se obtinham quando se ajustavam os parâmetros para a equação (2). Foi realizada uma análise gráfica dessa situação especificando as duas funções objetivos NSEv, função objetivo para a relação profundidade-volume e NSEa, função objetivo para a relação profundidade-superfície. Apesar de aparecerem duas funções não se trata de um problema multiobjectivo já que a função objetivo utilizada é a mesma, só que para critérios de análise diferentes.

Método de Nelder & Mead (Nelder-Mead)

O método proposto por Nelder & Mead é um método de busca direta. Toda sua teoria está associada ao conceito de uma figura geométrica chamada *simplex.*

Um *simplex* é uma figura geométrica que possui um número de vértices igual a o número de dimensões no espaço em que está contido mais 1, (Walters et al., 1991).

O método Nelder-Mead usa seu poliedro de busca (*simplex)* para tentar resolver o problema de adequação do passo na busca do ótimo global.

O algoritmo começa formando um poliedro de dimensões (D + 1), ou seja, um *simplex* de D + 1 pontos , x_i, onde i = 0, 1, 2,...,D. Esses pontos são aleatoriamente distribuídos ao longo da superfície de resposta . O índice dos pontos são ordenados de acordo com o critério de ascendência do valor da função objetivo, de tal forma que x_0 é o melhor ponto e x_D é o pior ponto. Para se obter um "ponto tentativa", x_r, o pior ponto, x_D, é refletido através da face oposta do poliedro usando um fator de ponderação F1. Geralmente, F1 é maior que zero. Se a reflexão através do centroide

melhorar o melhor ponto, x_0, ou seja, se esse novo vetor aplicado a função objetivo conseguir um resultado melhor do que o obtido quando se aplica x_0, o algoritmo Nelder-Mead toma outro passo na mesma direção baseado na suposição que o resultado pode ser melhor ainda. Esse novo passo, x_e, é obtido gerando um novo fator de escala F2. Se esta expansão do passo também melhorar o valor de $f(x_0)$, então se substitui x_D por x_e. Esta nova configuração de D + 1 pontos torna-se o próximo *simplex*, dando início ao recomeço do processo. Se, porém, x_e não melhorou x_D, então x_r substitui x_D. Se, porém, x_r não melhora x_0 na primeira instância, então x_r é comparado ao próximo pior ponto, x_{D-1}. Se x_r for melhor que x_{D-1}, então x_r substitui x_D. Se, porém, x_r é pior que x_{D-1}, utiliza-se uma nova escala F3, contraindo todo o *simplex*.

Método da Simulação por Emparelhamento (Simulated Annealing)

Esse método é um minimizador por natureza.

O termo “annealing” refere-se ao processo de lento resfriamento de uma substância fundida tal que seus átomos possam se emparelhar em uma configuração de energia mínima. Considerando que a substância se mantem próximo do equilíbrio, em uma temperatura T, temos que as energias atômicas E são distribuídas de acordo com a equação de Boltzman.

O algoritmo começa ao se estabelecer de forma aleatória um ponto xo e uma temperatura To. O ponto xo constituirá, inicialmente, o ponto ótimo da superfície de resposta. Através de uma distribuição gaussiana é gerado um desvio no valor do vetor. Se a função objetivo obtida aplicando-se o vetor xo for maior do que a função objetivo associada ao vetor xo acrescido do desvio então se toma essa nova função como a ótima e o vetor xo acrescido do desvio como sendo o novo xo. Caso contrário calcula-se a diferença d entre as duas funções objetivo comparadas e gera-se um número aleatório no intervalo [0,1] através de uma distribuição uniforme. Se esse valor aleatório for menor que o valor obtido pelo exponencial do valor de d multiplicado pela constante de Boltzman dividido pela temperatura multiplicada por -1, o novo valor de xo será o acrescido do desvio, (Metropolis et al., 1953). Caso contrário o algoritmo reduz a temperatura por um fator de redução recomeçando o processo.

Método da Evolução Diferencial (Differential Evolution)

O método Differential Evolution é um método de busca global. Sua base conceitual repousa sob o conceito de mutação.

Essa simples função estocástica é um otimizador que utiliza uma população para atacar o problema do ponto inicial, através de uma amostragem da função objetivo em vários pontos iniciais escolhidos aleatoriamente. Parâmetros de controle predefinidos definem o domínio a partir do qual os Np vetores desta população inicial são escolhidos. Price (2005) descreve esse algoritmo da seguinte forma: cada vetor é indexado com uma numeração que vai de 0 até Np – 1. Assim como outros métodos baseados em população, esse ponto gera novos pontos que são baseados na população de pontos já existentes. A geração desses novos números ocorre na forma de reflexões. Esses novos pontos obtidos passam por um estágio de seleção, onde são "defrontados" com a primeira população gerada obedecendo a ordem de criação, ou seja, o primeiro vetor gerado pela reflexão confronta-se com o vetor de índice zero da primeira população, o segundo gerado por reflexão confronta-se com o vetor de índice um e assim sucessivamente. Quando um membro da primeira população possui um valor de função objetivo melhor do que o valor da função objetivo do vetor de mesmo índice da nova população, dizemos que esse vetor sobreviveu ao processo de seleção. Após todos os vetores da primeira geração serem confrontados inicia-se um novo processo de seleção com os "melhores" membros da nova e antiga população.

Método da Busca Aleatória (Random Search)

O método de busca Random Search emprega, da mesma forma que o método Nelder-Mead, uma figura *simplex* de D + 1 pontos distribuídos aleatoriamente a partir de uma população de $N_p >$ D + 1 vetores. Uma reflexão através do centroide gera um novo ponto x_r. Se esse ponto é melhor que o atual pior ponto x_w, x_r substitui x_w.

Segundo Price (2005) a desvantagem desse algoritmo é que recolocar continuamente o atual pior ponto exerce uma alta pressão seletiva, o que pode forçar a população a convergir prematuramente.

RESULTADOS E DISCUSSÕES

Dentre os métodos utilizados o que obteve melhor convergência foi o método *Nelder-Mead*. A diferença entre os valores de NSE encontrados variam muito pouco, algo da ordem de 10^{-2}. Essa busca refinada do método de otimização só se justifica, com diferenças dessa ordem, para casos em que os resultados são muito sensíveis a erros no valor do parâmetro. O parâmetro 'a' se encaixa nesse caso devido ao seu caráter exponencial. A Tabela **1** mostra de forma resumida os dados encontrados durante os processos de otimização.

Tabela 1- Métodos usados para o processo de otimização utilizados seguidos dos valores ótimos de a e K para as funções objetivo NSEv e NSEa.

Método	a(V-H)	K(V-H)	NSEv	a(S-H)	K(S-H)	NSEa
Differential Evolution	2.523053	2380.73	0.9999686085381	2.582556	2107.60	0.994095329600
Nelder-Mead	2.524455	2373.77	0.9999687013460	2.582044	2110.20	0.994095362972
Random Search	2.524453	2373.78	0.9999687013457	2.582042	2110.21	0.994095362971
Simulated Annealing	3.319389	411.71	0.9662694490552	2.857844	1168.12	0.974077941262
Média	2.7228375	1885.00	0.9915438650713	2.651121	1874.03	0.98909099920
Desvio Padrão	0.3444199	850.61	0.0145921908905	0.119352	407.56	0.00866779304

Durante a análise da relação entre os valores encontrados de NSEv e NSEa foi gerado um gráfico com todos os valores possíveis dos NSE's para um intervalo definido de 'a' e 'K'. Foi possível perceber que na maioria dos casos o ajuste para a relação V-H foi melhor do que o ajuste realizado para a relação S-H. Isso pode ser visualizado através da tendência amostral dos pontos na Figura **3**. Note que é possível perceber algo semelhante às frentes de Pareto nos gráficos. Nesse trabalho o termo "frentes de Pareto" será utilizado com aspas, uma vez que não se trata de um problema multiobjectivo, já que a função objetivo usada é a mesma. Nas "frentes de Pareto" é possível observar que o valor máximo encontrado para NSEv não tem como par o valor máximo encontrado para NSEa e vice-versa. Isso confirma que um ajuste utilizando a equação (1) irá privilegiar a relação V-H, enquanto que um ajuste utilizando a equação (2) irá privilegiar a relação S-H. Isso implica que os parâmetros que otimizam a relação V-H não são os mesmos que otimizam a relação S-H.

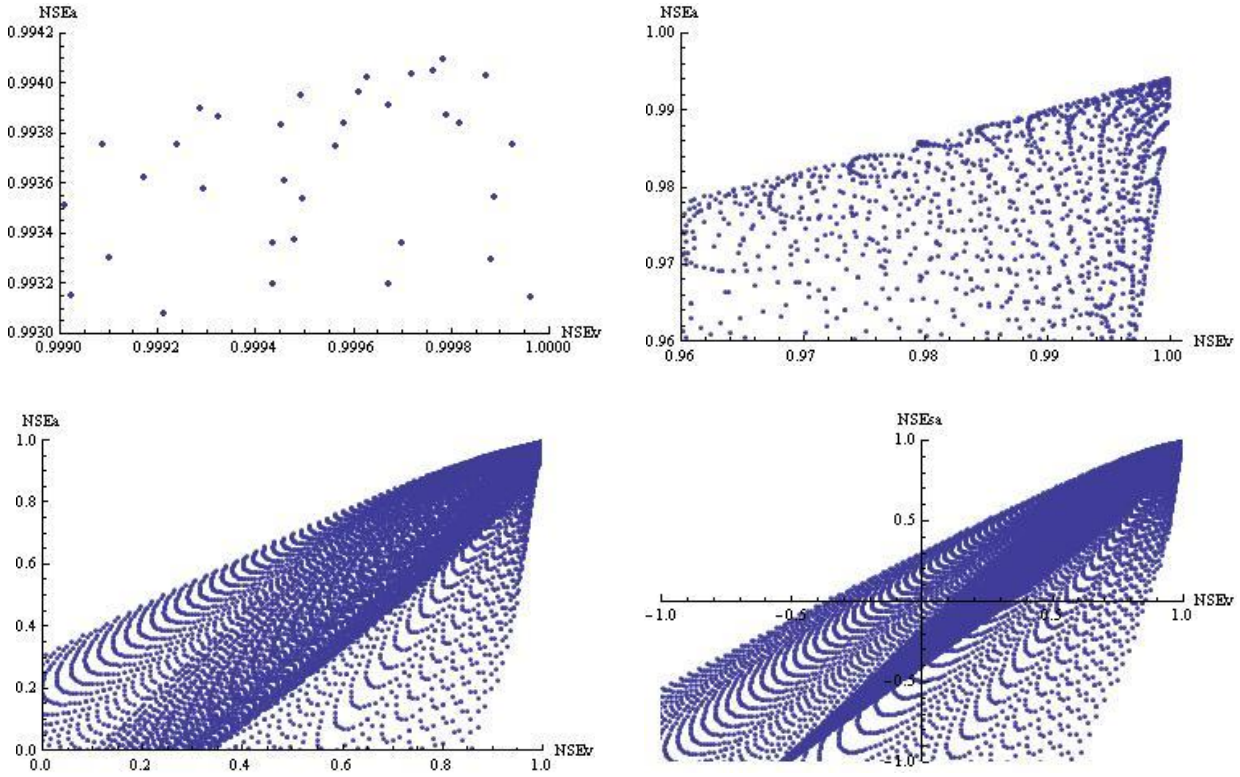

Figura 3- Gráficos de NSEv vs NSEa em diferentes escalas.

A plataforma *mathematica 8.0* permite que o usuário visualize a busca realizada pelo algoritmo de otimização através de setas que traçam o caminho percorrido. Esse tipo de instrumento pode ser muito útil na confecção de algoritmos de busca, pois além de avaliar o quão próximo o resultado do seu algoritmo chegou do máximo também é possível verificar a eficiência do mesmo em se aproximar do resultado desejado. Esse fluxo de setas permite, sobretudo, ao pesquisador ter uma visão global do seu algoritmo. As figuras Figura **4**, Figura **5**, Figura **6** e Figura **7** mostram as superfícies de resposta para função objetivo NSEv percorridas pelos algoritmos de otimização. Os pontos pretos correspondem aos valores da função objetivo durante a busca e as setas o caminho percorrido até o ótimo.

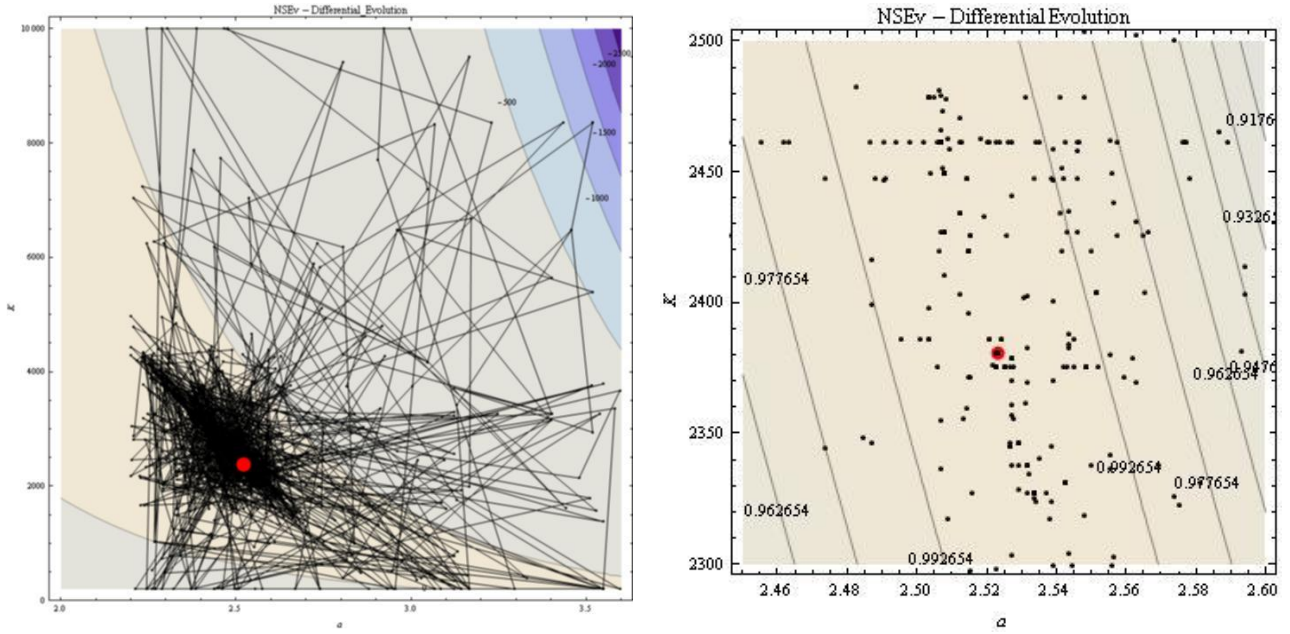

Figura 4- A imagem a esquerda mostra os caminhos percorridospelo algorítmo de otimização em uma escala maior. A imagem a direita mostra os pontos de busca próximos do ótimo.

O método *Differential Evolution* é um algoritmo que faz uma busca pormenorizada do espaço amostral. O ponto em vermelho mostra o resultado ótimo encontrado para os parâmetros. Note que apesar do bom "esquadrinhamento" feito pelo algoritmo ele não foi o responsável pela melhor resposta. Isso mostra que nem sempre algoritmos com malhas densas de busca convergem melhor para os resultados ótimos.

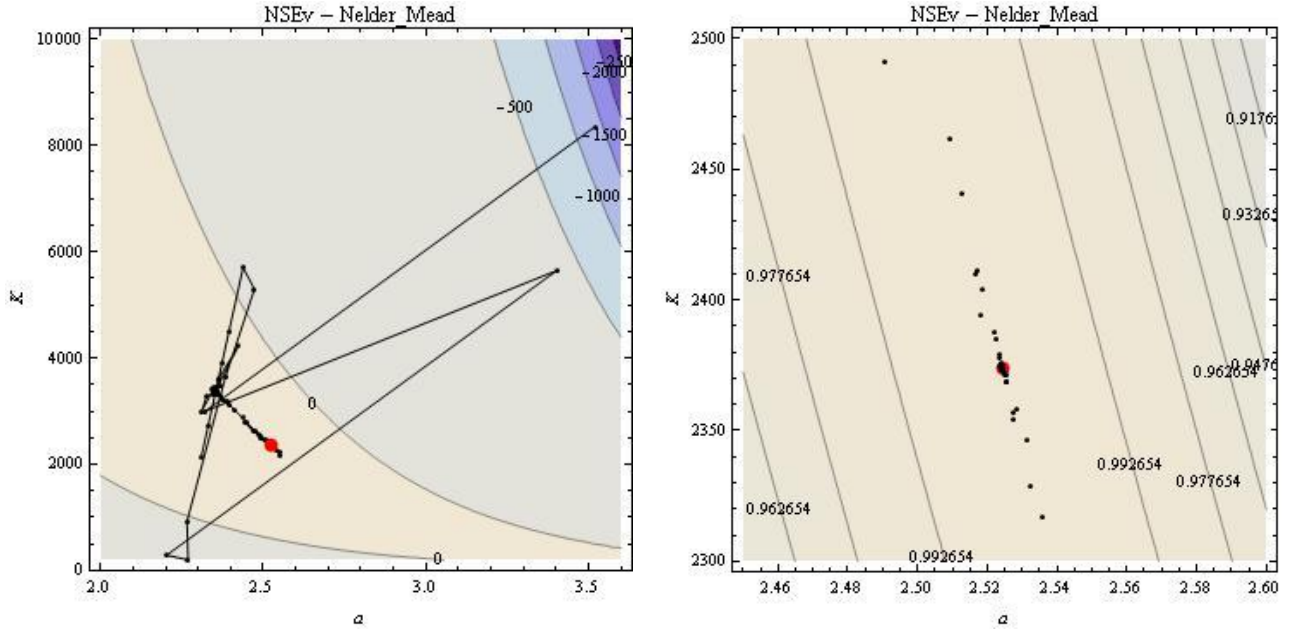

Figura 5- A imagem a esquerda mostra os caminhos percorridospelo algorítmo de otimização em uma escala maior. A imagem a direita mostra os pontos de busca próximos do ótimo.

O método *Nelder-Mead* se mostrou um método de otimização muito mais objetivo em detrimento dos demais. Apesar de sua malha de busca pouco densa esse foi o método que mostrou

melhor convergência. Note que o estreitamento da busca acontece após pouquíssimos passos convergindo no final para um paralelismo com as curvas de nível mais centrais.

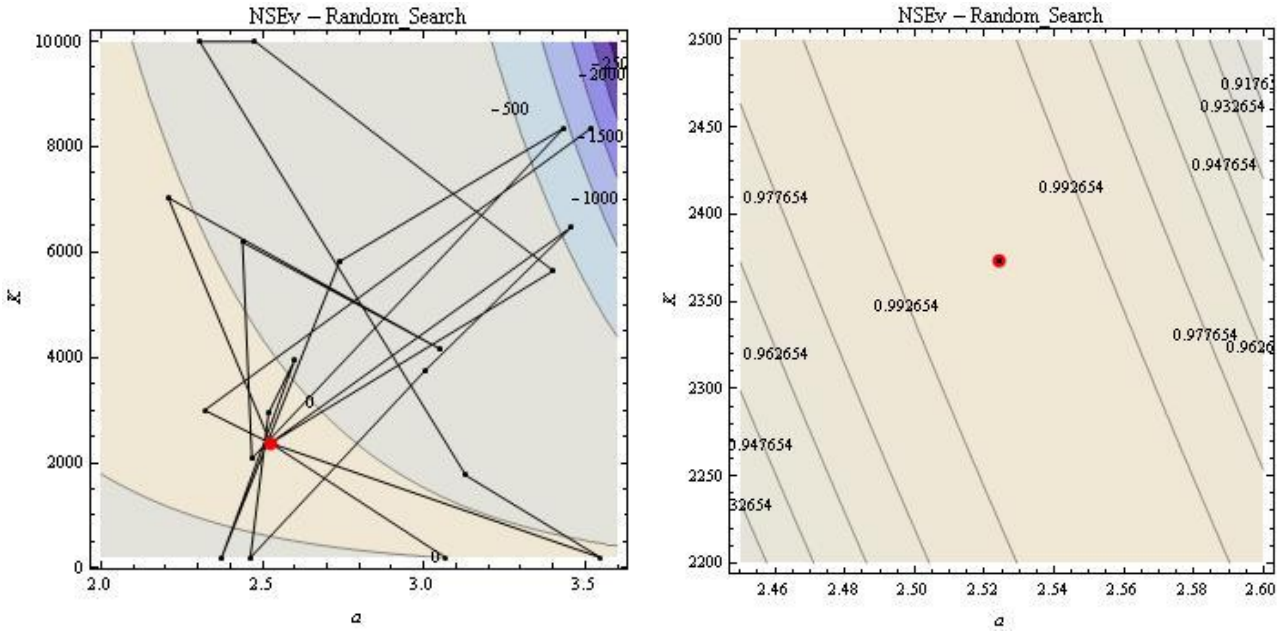

Figura 6- A imagem a esquerda mostra os caminhos percorridospelo algorítmo de otimização em uma escala maior. A imagem a direita mostra os pontos de busca próximos do ótimo.

O método *Random Search* obteve resultados muito próximos do método *Nelder Mead*, com a diferença aparecendo somente a partir da décima casa decimal. Talvez essa sutil diferença seja devido a malha pouco densa de pontos do método próximo a região do valor ótimo. Note que os passos desse método não sofrem o mesmo refinamento que o *Nelder Mead.*

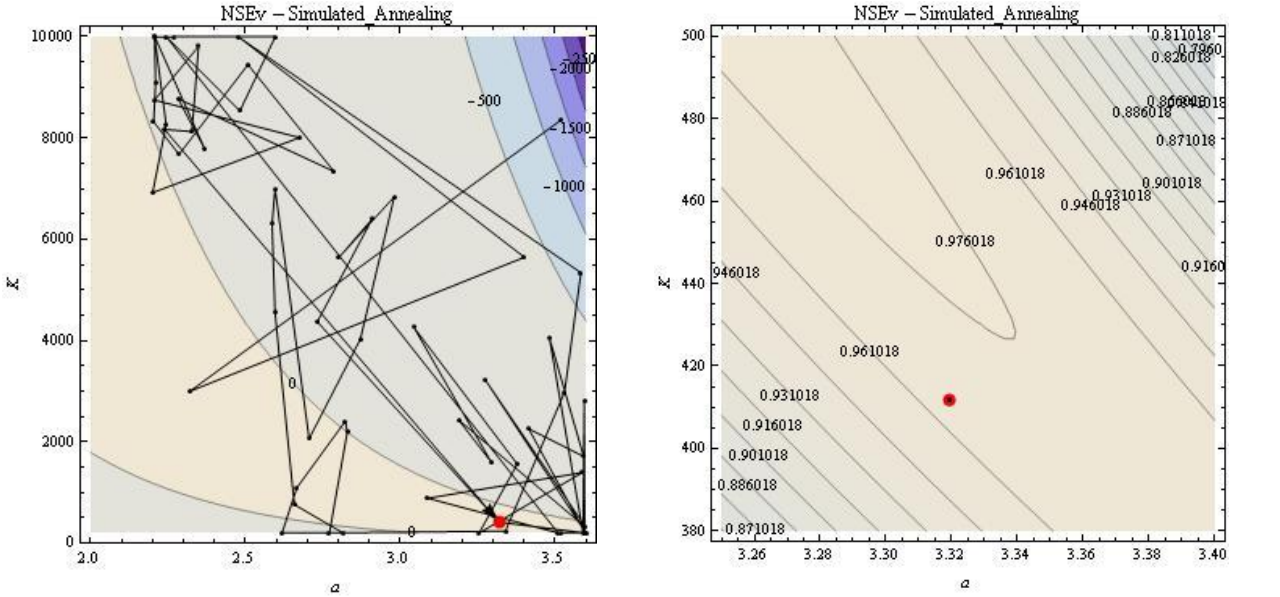

Figura 7- A imagem a esquerda mostra os caminhos percorridospelo algorítmo de otimização em uma escala maior. A imagem a direita mostra os pontos de busca próximos do ótimo.

O método *Simulated Annealing* dentre os quatro métodos testados foi o que mostrou pior convergência. A diferença para os demais métodos aparece a partir da segunda casa decimal. Dentre todos os métodos testados esse parece necessitar de um pós-processamento. Analisando seu diagrama de flechas é possível perceber que houve buscas próximas dos ótimos obtidos pelos demais métodos, só que devido a baixa acurácia dos passos não foi possível a convergência.

A diferença encontrada nos parâmetros parece não afetar muito os resultados da curva CAV, especialmente para a curva cota-volume. Na Figura **8** é plotado o gráfico da análise de resíduos para a relação entre a profundidade e o volume. Perceba que quando se utiliza a equação (2) para a realização do ajuste não houve uma piora significativa nos resultados, sendo a maior diferença cerca 6 mil m³, ou seja, menos de 1% do volume total. Outro fato curioso é que mesmo no ajuste em que se privilegia a relação V-H é possível perceber que para pequenas profundidades o ajuste feito com a equação (2), S-H, dá resultados melhores. Na Figura **9** é plotado o gráfico da análise de resíduos para a relação entre a profundidade e superfície da bacia hidráulica. Note que diferente da relação entre a profundidade e o volume é possível perceber um equilíbrio entre as diferenças, ora o melhor ajuste aparece para equação (1), ora para a equação (2). Porém, em termos de NSE, o resultado utilizando a equação (2) apresenta melhor ajuste.

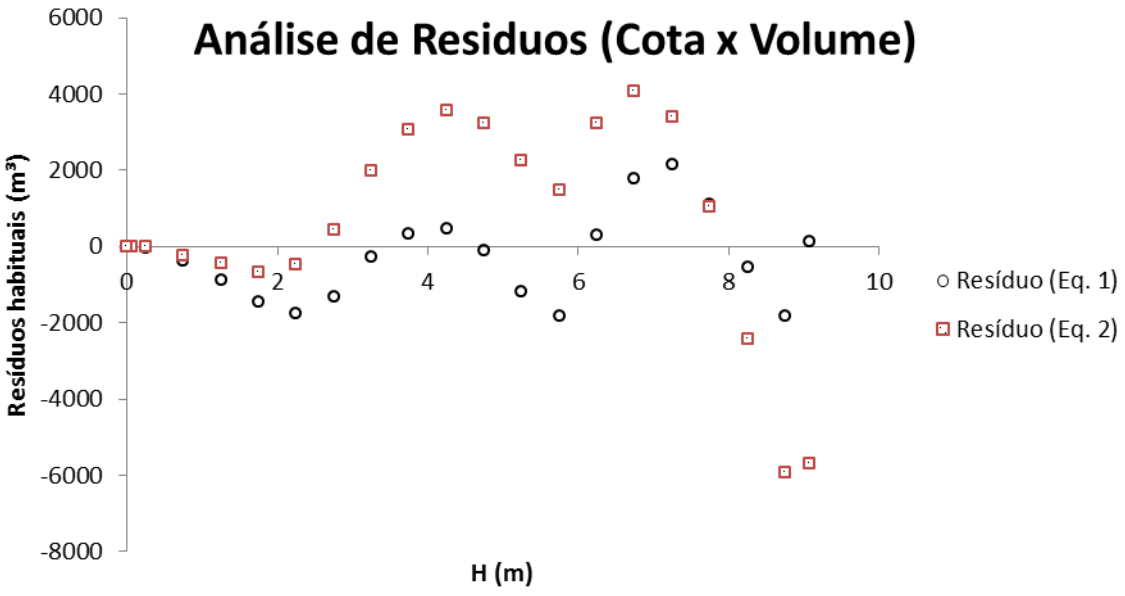

Figura 8 – Esse gráfico mostra a análise de resíduos feita para relação V-H quando se utilizam as duas funções objetivos NSEv e NSEa.

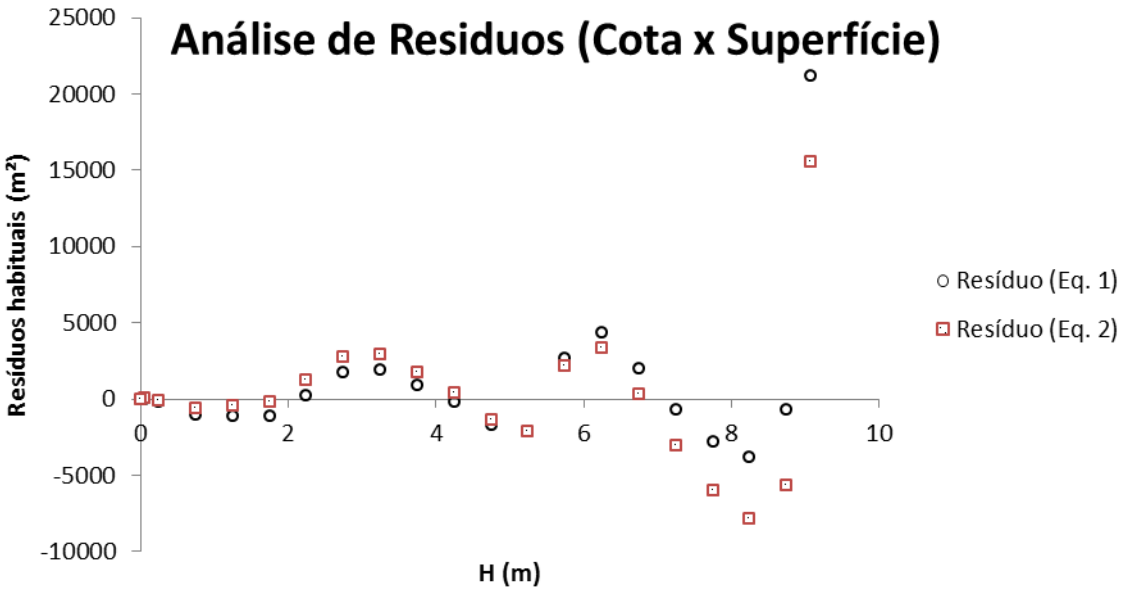

Figura 9 - Esse gráfico mostra a análise de resíduos feita para relação S-H quando se utilizam as duas funções objetivos NSEv e NSEa.

CONCLUSÕES

Dentre os métodos utilizados o que mostrou melhor convergência foi o método *Nelder Mead*. A diferença desse método para o *Random Search* só aparece a partir da décima casa decimal, mostrando que ambos os métodos convergem bem para a situação proposta nesse trabalho. Como a precisão utilizada nesse trabalho para cada método foi a *default* do sistema, acredita-se que com o aumento da precisão do sistema é possível obter um maior refinamento dos resultados.

O aparecimento de dois valores de NSE ótimos para a situação desse trabalho indica que a calibração desse modelo geomorfológico, especificamente, deve ser seguido de ponderação dos resultados. Ou seja, o usuário do modelo deve levar em consideração qual a relação mais importante no seu estudo, se no caso for V-H deve-se optar por NSEv, se for S-H deve-se optar por NSEa, se for S-V-H deve-se buscar uma nova função objetivo que pondere a relação entre os valores de NSE estudados nesse trabalho.

BIBLIOGRAFIA

BURTE, J. **LES PETITS AQUIFERES ALLUVIAUX DANS LES ZONES CRISTALLINES SEMI-ARIDES: FONCTIONNEMENT ET STRATEGIES DE GESTION.Etude de CAS DANS LE NORDESTE BRÉSILIEN.**, 17. Jun. 2008. Universidade federal do Ceará/Université Montepellier 2. Disponível em: <http://tel.archives-ouvertes.fr/tel-00399714/en/>. .

CAVALLINI, F. Computing the unit hydrograph via linear programming. **Computers & Geosciences**, v. 19, n. 9, p. 1285–1294, 1993. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/0098300493900309>. Acesso em: 30/6/2012.

FATEMAN, R. J. A review of mathematica. **Journal of Symbolic Computation**, v. 13, n. 5, p. 545–579, 1992. Academic Press Limited. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0747717110800112>. Acesso em: 30/6/2012.

JEŻOWSKI, J.; BOCHENEK, R.; ZIOMEK, G. Random search optimization approach for highly multi-modal nonlinear problems. **Advances in Engineering Software**, v. 36, n. 8, p. 504–517, 2005. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0965997805000475>. Acesso em: 23/6/2012.

JOHNSON, S. E.; MOORE, R. R. Surface reconstruction from parallel serial sections using the program mathematica: Example and source code. **Computers & Geosciences**, v. 19, n. 7, p. 1023–1032, 1993. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/009830049390007R>. Acesso em: 30/6/2012.

KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by simulated annealing. **Science (New York, N.Y.)**, v. 220, n. 4598, p. 671–80, 1983. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/17813860>. Acesso em: 3/3/2012.

LUERSEN, M. A.; RICHE, R. LE. Globalized Nelder–Mead method for engineering optimization. **Computers & Structures**, v. 82, n. 23-26, p. 2251–2260, 2004. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0045794904002378>. Acesso em: 18/6/2012.

METROPOLIS, N.; ROSENBLUTH, A. W.; ROSENBLUTH, M. N.; TELLER, A. H. Equation of State Calculations by Fast Computing Machines. **Journal of Chemical Physics**, v. 21, p. 1087 – 1092, 1953.

MOLLE, F.; CADIER, E. **Manual do Pequeno Açude**. III ed. Recife: SUDENE/ORSTON/TAPI, 1992.

NASH, J. E.; SUTCLIFFE, J. V. River flow forecasting through conceptual models part I — A discussion of principles. **Journal of Hydrology**, v. 10, n. 3, p. 282–290, 1970. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/0022169470902556>. Acesso em: 20/3/2012.

PRICE, K. V.; STORN, R. M.; LAMPINEN, J. A. **Differential Evolution**. 2005.

STORN, R.; PRICE, K. Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. **Journal of Global Optimization**, v. 11, n. 4, p. 341–359, 1997. Springer Netherlands. Disponível em: <http://dx.doi.org/10.1023/A:1008202821328>. .

WALTERS, F. H.; PARKER, L. R.; MORGAN, S. L.; DEMING, S. N. **Sequential Simplex Optimization**. 1991.

ZOTOS, K. Performance comparison of Maple and Mathematica. **Applied Mathematics and Computation**, v. 188, n. 2, p. 1426–1429, 2007. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0096300306014949>. Acesso em: 22/5/2012.