Proceeding Series of the Brazilian Society of Applied and Computational Mathematics, Vol. 5, N. 1, 2017.

Trabalho apresentado no CNMAC, Gramado - RS, 2016.

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

Algoritmo Genético de Dois Níveis para Problemas de Otimização

Aparecida de Fátima Castello Rosa[1]
Fabio Henrique Pereira[2]
Programa de Pós-graduação em Informática e Gestão do Conhecimento da Universidade Nove de Julho, PPGI/UNINOVE

Resumo. Algoritmos de otimização com dois níveis usam, em geral, uma combinação de dois modelos: um definido no espaço de busca original (*fine model*) e outro em um subespaço aproximado (*coarse model*). Em alguns casos, o *coarse model* precisa ser refinado a cada iteração elevando o custo computacional. Este trabalho apresenta um Algoritmo Genético de dois níveis que utiliza um *coarse model* global definido com base em um modelo de rede neural artificial, o qual é criado a partir da análise de componentes principais de um conjunto preliminar de soluções obtidas no espaço de busca original. Para ilustrar uma aplicação da abordagem proposta são apresentados os resultados promissores para um problema exemplo.

Palavras-chave. Algoritmo Genético, Otimização, Análise de Componentes Principais, Métodos dois-níveis, Rede Neural Artificial.

1 Introdução

A aplicação de algoritmos de otimização metaheurísticos, como o Algoritmo Genético (AG), à problemas das engenharias é muito difundida [1–3]. De modo geral, as abordagens baseiam-se na construção de um modelo substituto (*surrogate model*) construído a partir de um conjunto limitado de simulações computacionais, por exemplo, do método dos elementos finitos. A fim de encontrar uma solução ótima, ou próxima do ótimo, esses algoritmos normalmente exigem muitas avaliações do modelo, o que demanda um custo computacional razoavelmente alto.

Muitos algoritmos de dois níveis tem sido proposto na tentativa de acelerar a resolução desses problemas de otimização [4, 5]. Esses algoritmos utilizam uma combinação de dois modelos substitutos no procedimento de otimização: um definido no espaço de busca original (*fine model*) e outro em um subespaço (*coarse model*). O *coarse model* é criado com base em algumas avaliações do *fine model* e a sua precisão influencia fortemente a convergência. Em muitos casos, porém, os algoritmos de dois níveis possuem características locais, o que significa que o *coarse model* deve ser construído a cada iteração o que representa uma importante desvantagem dessas abordagens.

[1] afc.rosa@uninove.edu.br
[2] fabiohp@uninove.br

Portanto, apresenta-se neste trabalho um Algoritmo Genético de dois níveis (2LGA) que utiliza um *coarse model* global definido com base em Rede Neural Artificial (RNA) e Análise de Componentes Principais (PCA). Em problemas nos quais uma função matemática não é explicitamente disponível, o *fine model* é definido como uma regressão não-linear de um conjunto preliminar de soluções aproximadas (pontos no espaço de busca), os quais são determinados de acordo com um delineamento experimental adequado a fim de reduzir o número de simulações [6]. O *coarse model* é construído como uma aproximação do *fine model* em um subespaço de menor dimensão gerado pela Análise de Componentes Principais.

2 Modelos de Rede Neural Artificial

Uma Rede Neural Artificial (RNA) consiste em uma técnica de inteligência artificial inspirada na estrutura e funcionamento do cérebro humano. Uma RNA é composta por elementos interconectados, denominados neurônios artificiais, que são responsáveis pelo processamento das informações [7]. O modelo de RNA *Multilayer Perceptron* (MLP), usado em tarefas de reconhecimento de padrões, é caracterizado por ter, além das camadas de entrada e saída, uma ou mais camadas ocultas. As camadas ocultas atuam como detectores de características, enquanto que a camada de saída recebe o estímulo da última camada oculta e constroi o padrão que será a resposta. Normalmente, o algoritmo supervisionado *backpropagation* é utilizado para treinar um modelo MLP [7].

3 Subespaço dos Componentes Principais

A Análise de Componentes Principais é uma forma de identificar padrões em dados e expressá-los em um subespaço de dimensão reduzida [8]. O método funciona como definido nas etapas a seguir:
Etapa 1: obter um conjunto de dados - um conjunto de N variáveis independentes e as respectivas respostas do *fine surrogate model*: $fData$.
Etapa 2: subtrair a média - subtrair o valor médio de cada dimensão, a fim de produzir um conjunto de dados com média zero.
Etapa 3: calcular a matriz de covariância: $C = \mathbf{cov}(fData)$.
Etapa 4: calcular os autovetores e os autovalores da matriz de covariância: $A_{N\times N} = \mathbf{eig}(C)$
Etapa 5: escolher os n componentes principais - os autovetores com os maiores autovalores têm maior significado: $R = A_{N\times N}$. A noção de redução de dimensão surge ao ignorar os componentes menos significantes ($N - n$ colunas de $A, n<N$).
Etapa 6: determinar o novo conjunto de dados Y - a projeção dos dados originais no subespaço dos Componentes Principais: $Y = fData * R$.

DOI: 10.5540/03.2017.005.01.0482

4 Proposta do AG Dois Níveis

Nesta abordagem, a população de soluções candidatas do AG é evoluída em algumas gerações com o *fine model*. Espera-se que depois de algumas gerações iniciais as soluções candidatas estejam distribuídas em regiões próximas ao ponto ótimo e retenham uma parte da informação sobre a variação da função. Após essas gerações iniciais, as soluções candidatas são projetadas no subespaço dos componentes principais, conforme definido na seção 3. O PCA identifica os padrões nas soluções candidatas e as expressa em um subespaço de menor dimensão, no qual a avaliação das soluções pode ser computacionalmente mais barata.

Então, os dados projetados, associados aos valores originais de aptidão, são utilizados para treinar um modelo de rede neural, *coarse model*, que é aplicado para avaliar soluções em um novo processo de evolução nesse subespaço. A este processo dá-se o nome de correção da solução no subespaço.

Ao final, a última população obtida no processo de correção é transferida de volta ao espaço original, no qual algumas gerações do AG são realizadas para refinar as melhores aproximações da solução. O procedimento computacional é descrito em Algoritmo 1. Todos os parâmetros do algoritmo são definidos na Tabela 1.

Algoritmo 1: AG Dois Níveis (2LGA)

Entrada: $fModel, \nu_1, \nu_2, \nu_3$
Saída: $lp^f, fbest, ffitness$
1. *Escolha da população inicial* ip^f
2. $[lp^f, fbest, ffitness] =$ *evolui* ip^f para ν_1 gerações com $fModel$
3. $fData = [lp^f; fbest]$
4. $Y =$ PCA $(fData, n)$
5. $cModel =$ newff $(Y, ffitness)$
6. $[lp^c, cbest, cfitness] =$ *evolui* Y *para* ν_2 *gerações com* $cModel$
7. $cData = [lp^c; cbest]$
8. $X =$ iPCA$(cData)$
9. $[lp^f, fbest, ffitness] =$ *evolui* X *para* ν_3 gerações com $fModel$
10. Retorna

Para operar este processo, os seguintes componentes básicos devem ser definidos:

Fine model: é definido como um modelo de regressão não-linear baseado em rede neural (modelo substituto ou *surrogate model*), para um conjunto preliminar de simulações (ou a função matemática do problema de otimização, se disponível).

Global coarse model: um conjunto de parâmetros de otimização, associado às respectivas respostas do *fine model*, é projetado no subespaço dos componentes principais pelo operador de restrição R. O novo conjunto de dados resultante é utilizado para treinar um modelo de rede neural.

Operadores de prolongamento e restrição: uma matriz R, formada pelos componentes principais (colunas da matriz A no PCA), é utilizada para projetar no subespaço as

DOI: 10.5540/03.2017.005.01.0482

Tabela 1: Definição dos Parâmetros do Algoritmo.

$fModel$	Modelo rede neural no *fine level*
ν_1	Número de gerações na primeira evolução no *fine level*
lp^f	Última população após ν_1 gerações
$fbest$	Melhor solução no *fine level*
$ffitness$	Valores de aptidão dos indivíduos em lp^f
$fData$	Dados para serem projetados no subespaço PCA
Y	Aproximação dos dados no *coarse level*
n	Dimensão do *coarse level*
$cModel$	Modelo da rede neural no *coarse level*
ν_2	Número de gerações no *coarse level*
lp^c	Última população após ν_2 gerações
$cbest$	Melhor solução no *coarse level*
$cfitness$	Valores de aptidão dos indivíduos em lp^c
$cData$	Dados a serem transferidos para o *fine level* com PCA inverso
ν_3	Número de gerações na segunda evolução no *fine level*
X	População resultante no *fine level*

soluções aproximadas calculadas pelo AG. O prolongamento é realizado apenas tomando a transposta, R^T, e multiplicando-a pelos dados a serem transferidos.

Procedimentos de otimização: o AG é utilizado aqui. Após algumas gerações do AG no espaço original do problema, a projeção das soluções e a criação do *coarse model*, ocorre uma nova busca do AG no subespaço para encontrar uma melhor aproximação da solução. Posteriormente, a aproximação obtida no subespaço é refinada no espaço original.

5 Materiais e Métodos

A abordagem proposta foi aplicada na função exemplo denominada Hartmann 6, a qual representa um problema de seis dimensões. Trata-se de um problema multimodal, com 6 mínimos locais que dificultam a convergência do AG. Algumas informações importantes sobre esse problema estão resumidas na Tabela 2.

Os parâmetros tanto do *fine model* quanto do *coarse model* foram experimentalmente definidos. Foi utilizado o algoritmo de treinamento *Levenberg-Marquardt backpropagation*, 2 camadas ocultas, 23 neurônios por camada e a função de transferência Tangente sigmoide. O coeficiente de correlação entre os valores previstos e os valores conhecidos foi utilizado como uma medida eficiente do treinamento da rede neural.

Uma vez que o melhor modelo foi definido, ele foi utilizado para avaliar as soluções candidatas tanto para o AG padrão quanto para a abordagem proposta. Para o problema Hartmann 6, dois conjuntos de testes foram realizados utilizando 500 e 800 amostras (60% para o treinamento do modelo e 40% para validação e testes).

DOI: 10.5540/03.2017.005.01.0482

Tabela 2: Informações sobre a função Hartmann 6.

Número de variáveis	6
Domínio de busca	$0 < x_i < 1, i = 1, 2, ..., 6$
Número de mínimos locais	6 mínimos locais no intervalo especificado
Mínimo global	$x^* = (0,20169,\ 0,150011,\ 0,476874,\ 0,275332,$ $0,311652,\ 0,6573)$
Valor ótimo	$f(x^*) = -3,32237$

Essas amostras foram definidas de acordo com o método de amostragem do hipercubo latino, que é frequentemente utilizado para construir experimentos computacionais [6]. Ademais, a função original foi também utilizada a fim de avaliar os efeitos da precisão do *fine model* nos resultados. Foram realizados testes com outras funções que não são abordados neste trabalho por limitação de espaço.

Em todos os testes, um AG simples no MATLAB R2011 foi utilizado com os seguintes parâmetros: 50 gerações, tamanho da população 100, *crossover* 0,9 e taxa de mutação 0,05. Para o 2LGA, ν_1, ν_2 e ν_3 igual a 10, 10, e 5, respectivamente, foram utilizados. Outros valores desses parâmetros foram testados sem melhoria. Em especial, o aumento do número de gerações do AG no espaço original (ν_1 e ν_3) amplia a tendência de convergência para um mínimo local, própria do AG neste caso. A dimensão do subespaço para o *coarse model* é sempre definida como a metade em relação ao *fine model* ($n = N/2$ no PCA). Os resultados são apresentados na seção a seguir.

6 Resultados Numéricos - função teste Hartmann 6

Os resultados para a função Hartmann 6 são apresentados nas Tabelas 3 e 4. Neste caso, sete abordagens diferentes foram testadas conforme Tabela 3.

Uma vez que a função Hartmann 6 tem seis mínimos locais e apenas um mínimo global, os resultados da Tabela 3 foram definidos com base na distância Euclidiana entre a melhor solução encontrada e a solução ótima global. O objetivo, nesse caso, é avaliar o comportamento de convergência do método em relação aos mínimos locais da função. Vale destacar, no entanto, que no procedimento de otimização sempre foi considerado o valor da função objetivo (FO) na avaliação da qualidade das soluções geradas pelo AG.

É claro que a precisão do *surrogate model* tem uma forte influência na convergência do AG, e isso também é verdade para a abordagem proposta. No entanto, o 2LGA apresentou melhores resultados em relação ao AG mesmo quando o *surrogate model* não é tão preciso (quando ele é gerado com 500 amostras).

Foi observado que o AG tende a convergir para mínimos locais quando o *surrogate model* não é tão preciso (500 amostras). Por outro lado, se o número de gerações for reduzido o AG não encontra uma solução satisfatória. A evolução da população no *coarse level* pode evitar esse comportamento em alguns casos, principalmente quando o número

DOI: 10.5540/03.2017.005.01.0482

Tabela 3: Melhores e piores resultados em relação à distância euclidiana da solução ótima global.

Abordagem	Descrição	Distância Euclidiana em 10 execuções	
		Max(pior)	Min(melhor)
AG500	AG Simples (500 amostras)	1,2895	0,5135
2LGA500	AG dois níveis (500 amostras)	0,2946	0,1158
AG800	AG Simples (800 amostras)	1,1196	0,1519
2LGA800	AG dois níveis (800 amostras)	0,1983	0,1364
AG original	Evoluções com a função original	0,0888	0,0508
2LGA original	Evoluções com a função original	0,1349	0,0601
2LGA original $(\nu_1, \nu_2, \nu_3) = (20, 30, 5)$	Evoluções com a função original	0,0978	0,0506

Tabela 4: Resultados dos valores do parâmetros $x_i, i = 1, ..., 6$ e da função objetivo (FO).

Parâ-metro	GA 500	2LGA 500	GA 800	2LGA 800 original	GA original	2LGA original	2LGA $\nu_1 = 20$ $\nu_2 = 30$ $\nu_3 = 5$	Ótimo
x_1	0,0143	0,1325	0,1136	0,1424	0,2095	0,2191	0,2485	0,2017
x_2	0,1629	0,0872	0,2026	0,1989	0,1505	0,1964	0,1494	0,1500
x_3	0,9362	0,5385	0,5071	0,5650	0,4525	0,5021	0,4579	0,4769
x_4	0,3562	0,2495	0,3428	0,2960	0,2429	0,2565	0,2742	0,2753
x_5	0,2074	0,3000	0,2884	0,3358	0,3062	0,2989	0,3138	0,3117
x_6	0,6588	0,6658	0,7382	0,7200	0,6283	0,6581	0,6545	0,6573
FO	-1,2991	-3,1568	-2,8909	-3,0541	-3,2603	-3,2702	-3,2936	-3,3224

de gerações é mantido pequeno. Isso motivou a escolha de um número menor de gerações no 2LGA em relação ao AG. Isso significa que a influência da convergência do *fine model* sobre a evolução no subespaço foi reduzida.

Em contraste, quando a função original é utilizada para avaliar as soluções, um maior número de gerações pode produzir melhores resultados em ambos os casos, o que é esperado. Em relação ao custo computacional, houve uma redução de 2,49 segundos para 2,04 segundos, em média, para cada geração do algoritmo genético, o que representou uma redução cerca de 20% no custo computacional total.

DOI: 10.5540/03.2017.005.01.0482

7 Conclusão

A abordagem de dois níveis parece ser muito promissora para problemas de otimização. Apesar da influência da precisão do *fine model*, o 2LGA pode encontrar soluções mesmo quando o modelo é criado a partir de um pequeno número de amostras. Além disso, o 2LGA promove uma redução no custo computacional total, quando se considera a criação dos modelos aproximados e a transferência das soluções entre os subespaços. Uma extensão natural desta abordagem é aplicar algumas etapas do procedimento de dois níveis de forma iterativa, refinando o *coarse model* a cada etapa. Isso pode proporcionar um melhor modelo aproximado e aperfeiçoar os resultados, mas com um custo computacional adicional.

Agradecimentos

Os autores agradecem à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), processo (2014/08688-4), e à Universidade Nove de Julho.

Referências

[1] G. Crevecoeur, P. Sergeant, L. Dupre, and R. Van de Walle. A Two-Level Genetic Algorithm for Electromagnetic Optimization. *IEEE Transactions on Magnetics*, pages 2585–2595, 2010. DOI: 10.1109/TMAG.2010.2044186.

[2] O. E. Bukharov and D. P. Bogolyubov. Development of a decision support system based on neural networks and a genetic algorithm . *Expert Systems with Applications*, 42(15-16):6177–6183, 2015. DOI: 10.1016/j.eswa.2015.03.018.

[3] C. Xu, S. Lin, and Y. Yang. Optimal design of viscoelastic damping structures using layerwise finite element analysis and multi-objective genetic algorithm . *Computers and Structures*, 157:1–8, 2015.

[4] J. W. Bandler, R. M. Biernacki, S. H. Chen, P. A. Grobelny, and R. H. Hemmers. Space mapping technique for electromagnetic optimization. *IEEE Transactions on Microwave Theory and Techniques*, pages 2536–2544, 1994. DOI: 10.1109/22.339794.

[5] D. Echeverria and P. W. Hemker. Space Mapping and Defect Correction. *Comput. Meth. Appl. Math.*, pages 107–136, 2005. DOI: 10.2478/cmam-2005-0006.

[6] H. Xu. *A Catalogue of Three-Level Regular Fractional Factorial Designs*. Dissertation, Univ. California, 2004.

[7] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 1998.

[8] I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 2nd. edition, 2002.

