Revista de Engenharia e Pesquisa Aplicada, Volume 2, Número 1, 2016

Aplicação de Algoritmos de Visão Computacional para UVAs em Missões de Busca

Alcoforado, M. S.
Escola Politécnica de Pernambuco
Universidade de Pernambuco
50.720-001 - Recife, Brasil
msa@ecomp.poli.br

Fernandes, B. J. T.
Escola Politécnica de Pernambuco
Universidade de Pernambuco
50.720-001 - Recife, Brasil

Resumo *Síntese e análise de textura tem uma variedade de aplicações em Visão Computacional, e Processamento de Imagens, como por exemplo recuperação de imagens por imagem (CBIR, Content-Based Image Retrieval). Este trabalho apresenta a aplicação de redes neurais autoassociativas motivadas por teorias a respeito do cérebro humano em tarefas de análise de textura. O objetivo é a implementação do modelo proposto em um Veículo Aéreo Autônomo (UAV, Unmanned Aerial Vehicles).*

Abstract *Texture analysis and synthesis has a variety of applications in Computer Vision and Image Processing, such as image retrieval image (CBIR, Content-Based Image Retrieval). This paper presents the application of autoassociative neural networks motivated by theories about the human brain in tasks of texture analysis. The main purpose is to implement the proposed model in Autonomous Aerial Vehicles (UAV, Unmanned Aerial Vehicles).*

1 Introdução

Veículos aéreos autônomos (UAV, "Unmanned Aerial Vehicles") [1] têm sido aplicados nos mais diferentes contextos objetivando problemas com necessidade de sensoriamento remoto e missões de busca automatizadas. A visão computacional tem sido utilizada para resolver problemas específicos da área como medição de posicionamento relativo [2], pouso [3] e visão cooperativa para construção de mapas únicos a partir de imagens capturadas por vários UAVs [4]. Os avanços das técnicas de processamento de imagens [5] e reconhecimento de padrões [6] formaram a base para o surgimento de novas técnicas e algoritmos de visão computacional. Os UAVs podem então ser beneficiados por tais estudos na construção de métodos de exploração inteligente, vôo com obstáculos e identificação de objetos são possibilitados na plataforma do UAV. Este é o foco de desenvolvimento deste trabalho, que visa à aplicação de técnicas de visão
computacional em missões de busca utilizando UAVs.

Redes Neurais Artificiais (RNAs) vêm sendo utilizadas para os diversos campos da computação [7], sendo um destes campos a visão computacional, com diversas finalidades, como reconhecimento de padrões, filtros, navegação de robôs, etc. Neste trabalho, a RNA foi utilizada para auxiliar no trabalho de Visão Computacional,na área de Reconhecimento de Texturas, mais especificamente direcionados a UAVs.

2 Análise de Textura

Definido o processo da metodologia utilizada para a curva de *recall* e os detalhamentos sobre utilizadas neste trabalho podem ser analisadas. A Descrição de Textura é tratada em Pedrini [8] e Gonzalez [5] relacionando vários cálculos com bases estatísticas a fim de obter valores das características.

Porém, a nível experimental, previamente utilizado à descrição, é preciso demonstrar como funciona o filtro de Gabor, o filtro que foi utilizado para obter uma melhor caracterização das imagens retirando os seus ruídos (Afshang [11]).

2.1 Filtro de Gabor

Explorado por Afshang [11], o filtro de Gabor é uma técnica que pode ser utilizada para melhorar o resultado obtido na curva de *recall* (Zhang [14]). Sua aplicação diminui consideravelmente o ruído na imagem dependendo dos parâmetros nele utilizados.

Matematicamente, para utilização de qualquer filtro em uma imagem, basta que, depois de calculado sua máscara, uma operação de convolução seja realizada entre o sinal da imagem com o sinal do filtro. Feito este processo, uma nova imagem filtrada é retornada. Na equação 1, a fórmula do Filtro de Gabor.

$$\square\square,\square;\square,\square = \exp 12 \quad \square^2\square\square 2+ \quad \square^2\square\square 2 \quad . \quad \exp(2\square.\square.\square.\square) \quad (1)$$

Para facilitar sua programação, a regra de Euler deve ser aplicada, como mostrado na equação 2.

$$\exp(2\square.\square.\square.\square) = \cos 2\square.\square.\square.\square + \square \quad . \quad \sin 2\square.\square.\square.\square \quad (2)$$

Este filtro é composto a partir do produto entre uma função gaussiana e uma função harmônica (sinusóide). Nela, temos quatro parâmetros: $\square\square 2$ $\square$ $\square\square 2$ são os desvios padrões que vêm da equação gaussiana, $\square$ é a freqüência da função harmônica definida sobre a função de gabor e $\square$ representa a indicação, e nele que se define se o filtro terá um bom resultado ou não, como demonstrado nas fig. 2.

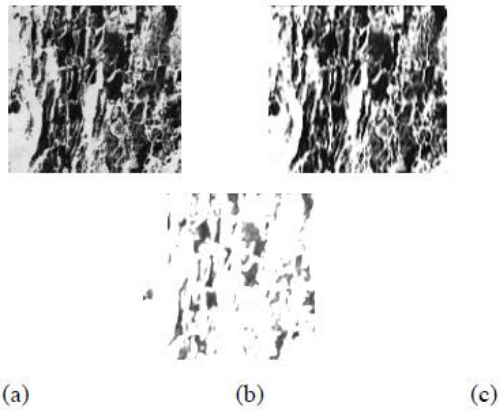

Fig 2. (a) Imagem Original. (b) e (c) Imagens Filtradas com o Filtro de Gabor.

Na fig. 2.b o filtro foi utilizado com os parâmetros corretos. Por outro lado a fig. 2.c teve uma utilização errônea do filtro. Como é possível notar, o primeiro resultado foi muito mais satisfatório que o segundo. A diferença entre eles deve-se a orientação escolhida para cada caso, $\square = \square / 10$ e $\square = \square / 2$ respectivamente. Tais números dados aos parâmetros foram puramente experimentais, já que para algumas imagens essa variação na indicação pode mudar de

imagem para imagem, assim como os desvios e freqüências. Para os resultados desta pesquisa, foram utilizados os parâmetros: desvios padrões foram fixados em 4 e 20 para x e y respectivamente, freqüência em 20 e indicação em □*/3* (Afshang, [11]).

2.2 Descritores de Textura Baseados na Matriz de Co-ocorrência

Após o estudo sobre o filtro de Gabor, o foco deste trabalho prossegue para seu problema central: a descrição de texturas. Iniciando pelos descritores baseados na matriz de co-ocorrência, propostos por Haralick [9]. Apesar de serem antigos, seus descritores são utilizados como referência em muitas pesquisas recentes que visam à descrição da textura (Abbadeni [10]). Foram propostas mais de 14 propriedades de características, porém Baraldi [13] mostrou que apenas seis eram realmente relevantes. Desses seis, cinco descritores foram programados: Máxima Probabilidade, Contraste, Homogeneidade, Uniformidade e Correlação.

Para poder calculá-los, é necessário obter a matriz de co-ocorrência □□, □. Ela é obtida através da freqüência entre tons vizinhos. O armazenamento dessas freqüências é feito como mostrado na fig. 3.

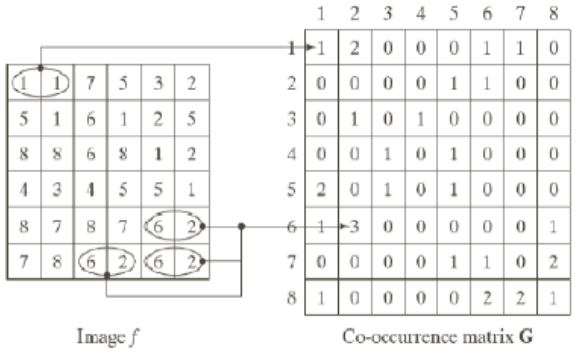

Fig 3. Imagem ilustrativa do cálculo da matriz de Co-ocorrência G, retirada de Gonzalez [5].

Calculada a matriz G, é necessário realizar sua normalização, dada como mostrado na equação 3.

$$□□,□ = □(□,□) \quad □ \qquad (3)$$

Onde □ é a soma das frequências entre os tons de cinza. Depois de calculada a Matriz de coocorrência da imagem, a descrição pode ser calculada. Na página seguinte, uma síntese dos descritores estudados, propostos por Haralick [9], com suas respectivas equações:

1. Máxima Probabilidade: Medida mais simples proposta por Haralick [9], consta na observação do maior valor contido na matriz de coocorrência e é dada pela equação 4.

$$□□□□□□□□=\max_{i,j}(□(□,□)) \qquad (4)$$

2. Contraste: Quando uma imagem apresenta alta freqüência entre tons de cinza altos e baixos possui um alto contraste. Em termos da matriz de co-ocorrência, quanto há uma alta concentração de elementos em torno da diagonal principal. Seu cálculo é dado pela equação 5.

$$□□□□□□= \quad □=0□□ \quad □=0□□ \quad □-□ \quad 2 \quad . \quad □(□,□) \qquad (5)$$

3. Homogeneidade: Mede a descrição dos pixels. Quanto mais homogênea a textura, maior seu valor de homogeneidade. Essa medida é apresenta uma correlação inversa com o contraste, pois possui maior concentração na diagonal principal da matriz de co-ocorrência e é dada pela equação 6.

$$□/ℎ□□= \quad □=0□□ \quad □=0□□11+□-□ \quad 2 \quad .□ \quad □,□ \qquad (6)$$

4. Segundo Momento Angular ou Uniformidade: Como já dito em seu nome, este descritor mede a uniformidade de uma imagem. Em termos de matriz de co-ocorrência, ela possui seus valores bem distribuídos ao longo da mesma. Sua fórmula é definida pela equação 7.

$$□□□□= \quad □=0□□ \quad □=0□□ \quad □^2(□,□) \qquad (7)$$

5. Correlação: Mede a dependência linear entre dois tons de cinza. Valores altos desta característica indicam uma existência linear entre os pares de tons de cinza. É dada pela equação 8.

$$□□□□□= 1□□□□□=0□□ \quad □=0□□□-□□□-□□□□,□ \qquad (8)$$

6. Onde □□□ □□ é dado pelas equações 9.

$$□□= \quad □=1□□□.□=1□□□(□, \quad □)□□= \quad □=1□□□.□=1□□□(□,□) \qquad (9)$$

2.3 Descritores de Textura Baseados Função de Autocorrelação

Em janeiro de 2011, Abbadeni [10] teve o seu artigo publicado, propondo novos descritores de texturas baseados na função de autocorrelação. Dois de seus descritores foram estudados. Porém, antes de começar a tratar sobre os seus descritores, um estudo em cima da função de autocorrelação foi necessário. Sua fórmula é dada pela equação 10.

$$\square\square\square, \square\square = 1\square - \square\square\square - \square\square$$

$$\square\ \square = 0\square - \square\square - 1\ \square = 0\square - \square\square - 1\square\square, \square \quad . \quad \square(\square + \square\square, \square + \square\square) \qquad (10)$$

Onde I é a matriz representando os valores de pixels da imagem, $0 \leq \square\square < \square$ e $0 \leq \square\square < \square$. $\square\square$ e $\square\square$ representam o mapeamento das linhas e colunas da função, respectivamente. Diferentemente da Matriz de Co-ocorrência, que realiza suas operações com relação a frequência entre pixels vizinhos, a Função de Autocorrelação verifica a interação entre regiões. Abbadeni [10] a escolheu por causa de sua periodicidade equivalente e que tem como característica uma lenta diminuição apresentando poucas variações.

1. Rugosidade: De acordo com o Abbadeni [10], a própria função de autocorrelação por si só, já traz informações de rugosidade. A dedução e explicação, disponíveis em seu trabalho, mostram que esta característica depende da quantidade de máximos presentes na função de autocorrelação. É dada pela equação 11.

$$\square\square = 112\ \square\ \ \square = 0\square - 1\square = 0\square - 1\square\square\square\square(\square, \square) \quad \square + \square = 0\square - 1\square = 0\square - 1\square\square\square\square(\square, \square) \quad \square \ (11)$$

Onde $\square\square\square\square(\square, \square)$ e $\square\square\square\square(\square, \square)$ foram obtidos pelo cálculo dos máximos utilizando derivadas parciais em x e y da função de autocorrelação, respectivamente.

2. Ocupação: Possui uma relação inversa com a rugosidade. É calculada a partir do resultado obtido na Rugosidade de uma dada imagem. Seu cálculo é dado pela equação 12.

$$\square\square\square\square\square = 1 - \square\square 1\square \qquad (12)$$

Onde $\square = 4$, segundo Abbadeni [1], dentre os vários valores testados, este foi o que obteve melhores resultados.

3 Redes Neurais

Redes Neurais Artificiais (RNAs) são técnicas computacionais que apresentam um modelo matemático inspirado na estrutura neural de organismos inteligentes e que adquirem conhecimento através da experiência. As redes neurais, ou redes artificiais de neurônios, foram desenvolvidas a partir de uma tentativa de criar em computador, a partir de estudos neurobiológicos, um modelo computacional que simule a estrutura e funcionamento do cérebro humano. Assim como o sistema nervoso é composto por bilhões de células nervosas, a RNA também seria formada por unidades que nada mais são que pequenos módulos que simulam o funcionamento de um neurônio. Estes módulos devem funcionar de acordo com os elementos em que foram inspirados, recebendo e retransmitindo informações.

Um neurônio artificial é capaz de um único processamento. Cada entrada recebe somente um tipo de sinal ou informação. Como um neurônio pode possuir várias entradas, então ele pode perceber diferentes sinais. Porém, ligar vários neurônios similares em rede, faz com que (10) o sistema consiga processar mais informações eoferecer mais resultados.

Com esse objetivo, cria-se uma rede em camadas, também conhecida como Perceptron Multicamadas (Multilayer Perceptron – MLP). Usualmente as camadas são classificadas em três grupos:

- **Camada de Entrada**: onde os padrões são apresentados à rede;

- **Camadas Intermediárias ou Ocultas**: onde é feita a maior parte do processamento, através das conexões ponderadas; podem ser consideradas como extratoras de características;

- **Camada de Saída**: onde o resultado final é concluído e apresentado.

A arquitetura de uma rede MLP, também denominada rede Feed Forward de múltiplas camadas, contém pelo menos uma camada intermediária entre as camadas de entrada e de saída. A figura 4 apresenta a arquitetura de uma rede MLP.

A propriedade mais importante das redes neurais é a habilidade de aprender em seu ambiente, e com isso melhorar seu desempenho. Isso é feito através de um processo iterativo de ajustes aplicado a seus pesos, o treinamento. O aprendizado ocorre quando a rede neural atinge uma solução generalizada para uma classe de problemas. Como citado anteriormente, o modelo de aprendizado utilizado na MLP, durante a iniciação científica, foi o autoassociativo.

A escolha do mesmo foi motivada pelo modelo do livro On Intelligence do autor Jeff Hawkins.

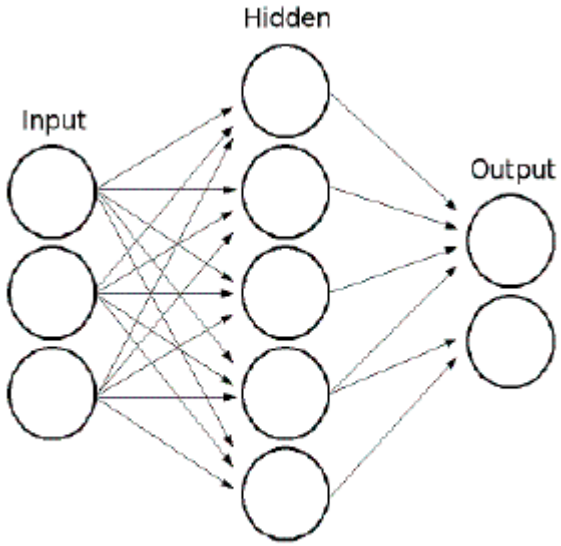

Figura 4. Modelo de MLP

No livro, Hawkins faz uma argumentação convincente e perspicaz, apresentando um modelo da inteligência humana baseado na análise do funcionamento do neocórtex humano. Segundo o autor, esta região do cérebro responsável pela inteligência tem seu funcionamento baseado na noção de autoassociatividade. Ainda segundo Jeff, o funcionamento da inteligência humana ocorre com base na nossa capacidade de predição de memória, onde armazenamos as informações recebidas pelos nossos sentidos durante o decorrer da nossa vida. Com base em estímulos do meio, e feed-backs interiores, podemos acionar trechos de memória no neocórtex já vividos e com isso criar o que seria a inteligência de saber o que fazer ou pensar para cada situação.

Sendo assim, o modelo autoassociativo seria o modelo mais próximo do funcionamento do cérebro humano.

3.1 Rede Neural Autoassociativa

No modelo autoassociativo, deseja-se que a entrada passada à rede seja retornada como saída. Logo, a rede neural possui a mesma quantidade de neurônios na camada de entrada e na camada de saída.

As redes autoassociativas buscam aprender a constituição interna de uma dada classe de padrões com base apenas nos elementos que a compõem. Nos padrões autoassociativos, poucas camadas intermediárias são usadas, filtrando, assim, as informações mais relevantes.

A desvantagem desse modelo é a alta complexidade computacional que ele possui, devido a quantidade de neurônios e conexões. Por outro lado, as redes neurais autoassociativas são capazes de aprender de modo implícito as características internas dos dados de entrada. Ou seja, elas conseguem aprender certas características pertencentes ao padrão apresentado na entrada sem a necessidade de qualquer conhecimento anterior ou instrução específica.

4 Resultados Obtidos

No caso dessa Iniciação Científica, usando as características extraídas das 60 texturas obtidas préviamente , as quais foram divididas cada uma em 16 imagens(figura 5),cada imagem formada por 8 características, resultando em um banco de dados com 960 imagens, foi treinada a Figura 4. Modelo de MLP MLP autoassociativa. Para cada imagem passada, a rede neural foi treinada com a mesma, e todas as outras imagens restantes foram passadas uma por uma como entrada, e obtendo-se a saída de cada uma, foi calculada a distância *city-block* (diferença entre a entrada dada e a saída obtida).

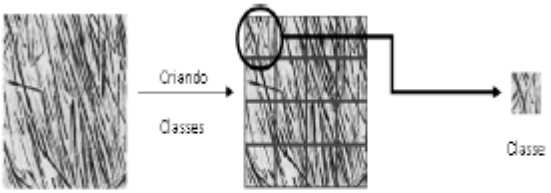

Fig. 5. Imagem ilustrativa de como as classes foram obtidas

Logo após, os resultados foram ordenados pelo de menor distância entre entrada e saída e o *recall*, número de imagens corretas retornadas dividido pelo número total de imagens corretas, de cada imagem foi calculado e dividido por 960 com o propósito de encontrar a tava média de *recall* entre todas as imagens. Com a taxa média de *recall* foi analisado o quanto a MLP autoassociativa conseguiu distinguir as texturas.

Os treinos foram feitos passando-se as características de três formas: todas as 8 de uma vez, de 7 em 7 e de 1 em 1. Os valores do *recall* foram armazenados a cada 4 imagens testadas. Os resultados obtidos mais relevantes, utilizando um fator de aprendizado de 0.2, podem ser vistos nas Tabelas 1,2 e 3.

Como se pôde observar, quando se testou a rede neural introduzindo como entrada todas as características de uma vez, o melhor resultado ocorreu com um *recall* médio de 18,04%. Notase também que os melhores resultados foram

obtidos à medida que se aumentou a quantidade de camadas ocultas. Os resultados notados na Tabela 2 demonstram que ao passar apenas uma característica de cada vez, a rede neural autoassociativa apresentou um melhor desempenho em comparação com o teste que recebeu todas as 8 características, consequência que será analisada em estudos futuros. Não se observa diferenças significantes entre a os valores da Tabela 1 e a Tabela 3, mas, ao passar todas as 8 características como entrada, ainda obteve-se resultados superiores.

Tabela 1

Características	Camadas Ocultas	Neurônios na Camada Oculta	Número de Iterações	Recall (Após 104 imagens testadas)
8	1	5	1000	17,54 %
8	1	2	1000	17,62%
8	1	3	1000	17,88%
8	1	10	1000	17,83%
8	2	5	1000	17,79%
8	2	2	1000	18,04%
8	2	3	1000	17,70%
8	3	5	1000	17,84%
8	3	2	1000	17,98%
8	3	3	1000	17,95%

Tabela 2

Características	Camadas Ocultas	Neurônios na Camada Oculta	Número de Iterações	Recall (Após 104 imagens testadas)
8	1	5	1000	17,54 %
8	1	2	1000	17,62%
8	1	3	1000	17,88%
8	1	10	1000	17,83%
8	2	5	1000	17,79%
8	2	2	1000	18,04%
8	2	3	1000	17,70%
8	3	5	1000	17,84%
8	3	2	1000	17,98%
8	3	3	1000	17,95%

Tabela 3

Características	Camadas Ocultas	Neurônios na Camada Oculta	Número de Iterações	Recall (Após 104 imagens testadas)
2,3,4,5,6,7,8	1	5	1000	17,84%
1,3,4,5,6,7,8	1	5	1000	17,85%
1,2,4,5,6,7,8	1	5	1000	17,65%
1,2,3,5,6,7,8	1	5	1000	17,82%
1,2,3,4,6,7,8	1	5	1000	17,65%
1,2,3,4,5,7,8	1	5	1000	17,76%
1,2,3,4,5,6,8	1	5	1000	17,73%
1,2,3,4,5,6,7	1	5	1000	17,90%

5 Conclusão

Com os dados e as análises feitas neste trabalho, compreende-se que a MLP autoassociativa não rendeu resultados com um nível bom o suficiente para ser implementado em UAVs quando treinadas para reconhecer texturas. Como trabalhos futuros, serão realizados estudos mais aprofundados sobre técnicas para escolha dos padrões utilizados no treinamento das redes neurais, como taxa de aprendizado, quantidade de camadas ocultas, pesos, entre outros, a fim de obter um melhor resultado.A forma sugerida para melhorar a busca pela configuração ótima da rede é através do algoritmo genético[15] ou ,até mesmo, o uso do PSO(Particle Swarm Optimization)[16].

Referências

[1] A. Ollero e L. Merino, "Control and perception techniques for aerial robotics," Annu. Revs. Contr., vol. 28, no. 2, pp. 167-178, 2004.

[2] O. Amidi, T. Kanade, e K. Fujita, "A visual odometer for autonomous helicopter flight," Robot. Autonom. Syst., vol. 28, no. 2, pp. 185-193, 1999.

[3] S. Saripalli, J.F. Montgomery, e G.S. Sukhatme, "Visually guided landing of na unmanned aerial vehicle," IEEE Trans. Robot. Automat., vol. 19, no. 3, pp. 371-380, 2003.

[4] L. Merino, J. Wilklund, F. Caballero, A. Moe, J. Dios, P-E Forssén, K. Nordberg, e A. Ollero, "Vision-Based Multi-UAV Position Estimation," IEEE Robotics & Automation Magazine, pp. 53-62, 2006.

[5] R. C. Gonzalez e R. E. Woods, "Processamento de Imagens Digitais," 3.Ed. São Paulo: Pearson Prentice Hall, 2010.

[6] R. O. Duda e P. E. Hart, "Pattern Classification and Scene Analysis," New York: Wiley, 1973.

[7] http://www.inf.ufsc.br/~visao/2001/saulo/ .Último acesso em 14 de setembro de 2011.

[8] Pedrini, H. e Schwartz, W. R. (2008) "Análise de Imagens Digitais – Princípios, Algoritmos e Aplicações".

[9] Haralick, R. e Shapiro, L. G. (1993) "Computer and Robot Vision", Vol. 2, Addison- Wesley, Reading, MA, Estados Unidos.

[10] Abbadeni, N. (2011) "Computacional Perceptual Features for Texture Representation and Retrieval", In: IEEE Transactions on Image Processing.

[11] Afshang, M., Helfroush, M.S. e Zahernia, A., (2009) "Gabor Filter Parameters Optimization for Texture Classification Based on Genetic Algorithm" IEEE, In: Machine Vision, 2009. ICMV.

[12] Aigrain, P., Zhang, H., e Petkovic, D. (1996) "Content-based representation and retrieval of visual media: A review of the stateof- the-art" Multimed. Tools Appl. 3, 3,179– 202.

[13] Baraldi, A. e F. Parmiggiani, 1995. "An investigation of the textural characteristics

associated with grey level co-occurrence matrix statistical parameters", *IEEE Trans. Geosci. Remote Sensing*, Vol. 33, 2, pp. 293-304.

[14] Zhang, J., e Ye, L. (2009) "Ranking Method for Optimizing Precision/Recall of Content-Based Image Retrieval" In: Ubiquitous, Autonomic and Trusted Computing, 2009. UICATC '09. Symposia and Workshops on. pp. 356 – 361

[15] http://pt.wikipedia.org/wiki/Algoritmo_gen%C3%A9tico . Último acesso em 14 de setembro de 2011.

[16] http://en.wikipedia.org/wiki/Particle_swarm_optimization . Último acesso em 14 de setembro de 2012.