Sistemas & Gestão 9 (2014), pp 14-20

SISTEMAS & GESTÃO
Revista Eletrônica

UMA REDE NEURAL DINÂMICA NA OTIMIZAÇÃO DE UM CONJUNTO DE RESTRIÇÕES LINEARES

A DYNAMIC NEURAL NETWORK IN THE OPTIMIZATION OF A GROUP OF LINEAR RESTRICTIONS

Walter Roberto Hernández Vergara[a]; Fabio Alves Barbosa[a]

[a] **Universidade Federal da Grande Dourados** (UFGD) - Grande Dourados, MS, Brasil - Faculdade de Engenharia, Departamento de Engenharia de Produção

Resumo

Nesta pesquisa apresentamos uma aproximação de rede neural para resolver um problema de programação não linear. A rede neural é derivada de um problema de programação inteira zero-um em outro problema de programação não linear equivalente, substituindo restrições zero-um por restrições iguais quadráticas côncavas. O método da função de penalidade é implantado como uma rede neural análoga. Uma arquitetura de rede neural e o algoritmo associado, na forma de equações diferenciais ordinárias não lineares, foram desenvolvidos. Este método é aplicado para resolver um problema de alocação de recursos. Nossa abordagem é significativa na precisão da modelagem como também na redução dramática no tempo de treinamento. Os resultados das simulações são apresentados.

Palavras-chave: Redes neurais. Pesquisa operacional. Otimização. Método da função de penalidade. Programação inteira.

Abstract

This paper presents a neural network approach for solving nonlinear programming problems. The neural network is derived from a zero-one integer programming problem into an equivalent nonlinear programming problem, replacing zero-one constraints with quadratic concave equality constraints. The penalty function method is implemented as an analog neural network. Neural network architecture and its associated algorithm as nonlinear differential equations have been developed. This method is applied to solve a resource allocation problem. Our approach significantly improved the modeling accuracy and dramatically reduced the training time. Simulation results are also shown.

Key-words: *Neural networks. Operational research. Optimization. Penalty function method. Integer programming.*

1. INTRODUÇÃO

Em muitas aplicações científicas e da engenharia demandam-se soluções reais e rápidas a problemas de otimização. De qualquer forma, algoritmos tradicionais, processados em computadores digitais, não são muito precisos para fornecer soluções realísticas. Portanto, a pesquisa de soluções em tais casos torna-se importante e essencial.

Na década de 80, Hopfield *et* Tank (1985) introduziram uma nova abordagem, atrativa e promissória, baseadas em redes neurais recorrentes (Hopfield *et* Tank, 1985; Tank *et* Hopfield, 1986), que fornecem soluções ótimas a problemas de otimização. O trabalho produtivo deles foi inspirado na pesquisa de muitos pesquisadores no estudo das redes neurais na resolução de equações em problemas de programação linear e não linear (Abe, 1990; Chua *et* Lin, 1984; Liao *et* Qi, 1999). A característica mais significativa de uma rede neural na otimização é a formulação de um simples modelo (circuito) implementado em um computador. Em outras palavras, um circuito elétrico é construído para gerar soluções rápidas a problemas de otimização.

Muitos pesquisadores tentaram reproduzir o problema do caixeiro viajante (Traveling Sales Man Problem - TSP) a partir dessa abordagem (Chua *et* Lin, 1984). Esse problema (em geral, problema de atribuição quadrática) é caracterizado pela definição de uma função de energia quadrática e um conjunto de variáveis que devem satisfazer as restrições de uma matriz de permutações. Essas restrições são agrupadas em uma matriz quadrada, onde cada elemento de entrada tem um valor zero ou um. Os métodos tradicionais têm uma grande dificuldade em satisfazer as restrições da matriz de permutações, porque utilizam uma função de energia com um número de parâmetros fixos. Mas, recentemente, um

PROPPI / LATEC
DOI: 10.7177/sg.2014.v9.n1.a2

Revista Eletrônica Sistemas & Gestão
Volume 9, Número 1, 2014, pp. 14-20
DOI: 10.7177/sg.2014.v9.n1.a2

número de ferramentas, baseadas na inteligência artificial, foi criado com a finalidade de resolver eficientemente esses problemas. Eles incluem variáveis determinísticas que fixam a convexidade da função de energia com o propósito de atingir um ponto mínimo local ou global.

Do ponto de vista da otimização, muitos modelos de redes neurais existem e poderiam ser divididos em duas classes. Uma classe representa os modelos de redes neurais baseados no gradiente, que são aplicados em problemas de otimização sem restrições. Esses problemas normalmente surgem:

Da transformação de um problema de minimização com restrições em outro problema sem restrições pelo método da função penalidade ou pelos multiplicadores de Lagrange (Martins *et al.*, 2011; Oliveira, 2005) e,

De problemas complementares com funções NCP (Liao *et* Qi, 1999), o que permite que vários protocolos da camada de uma rede operem no mesmo enlace de comunicação.

A outra classe está baseada em modelos de redes neurais no gradiente projetivo que derivam de problemas de minimização com restrições e problemas complementares com condições KKT (He, 1992), presentes em várias redes neurais na solução linear, quadrática e problemas de programação não linear convexo. Esses modelos correspondem a algumas variantes do método do gradiente projetivo para problemas complementares e problemas de desigualdade de variantes (Yanover *et al.*, 2006; He, 1992). Geralmente, essas abordagens consistem em três passos. Primeiro, as restrições de um problema de otimização são convertidas em outro problema equivalente de otimização sem restrições, isto é, as restrições zero-um são substituídas por restrições de igualdade quadráticas côncavas com o objetivo de forçar as variáveis de decisão a ser zero ou um. Segundo, a partir do problema de otimização sem restrições, um conjunto de equações diferenciais é derivado. Terceiro, segundo o sistema dinâmico é implementado um vetor neural de camada iterativa (Cichocki *et* Unbehauen, 1992).

Nesta pesquisa discutimos uma espécie de estrutura de rede neural, tal como a rede de Hopfield (1985). Esta estrutura tem boas propriedades de convergência e estabilidade. O método de *Branch and Bound* (Balas *et* Carrera, 1996) e o método da função de penalidade são relatados. A primeira metodologia fornece uma solução referencial ao problema e a segunda, representa a base da rede neural.

Normalmente, um termo de penalidade é incluído na função de energia com o objetivo de forçar o conjunto de restrições a encontrar um ponto mínimo local. Também este termo pode induzir a um falso mínimo local na função de energia e envolver parâmetros livres.

A nova abordagem baseada em redes neurais elimina a necessidade do termo de penalidade. Ela já foi aplicada a outros problemas de otimização mais complexos, incluindo problemas de atribuição paramétrica e problemas de atribuição quadrática (Farago *et* Morabito, 2000; Tosserams *et al.*, 2006; Bjorkund *et* al., 2005).

Assim, vamos considerar o seguinte problema de otimização: existem *m* centros de serviços e *n* consumidores. Cada consumidor tem r_j quantidades do requerimento de serviço, j = 1, 2, ..., n e este somente pode ser satisfeito por um centro de serviço. Cada centro de serviço tem b_i quantidades de capacidade de serviço, i = 1, 2, ..., m e estas quantidades não podem ser excedidas. Existe um custo c_{ij} entre cada centro de serviço e cada consumidor. Se o consumidor j é atendido por um centro de serviço, i = 1, 2, ..., m; j = 1, 2, ..., n. A questão é a seguinte: quantos consumidores devem ser alocados aos centros de serviços de forma que o custo total seja mínimo.

Esse problema pode ser formulado como um problema de programação inteira zero-um.

$$\text{Min f(x) ou } Min\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}\, x_j \qquad (1)$$

Onde x é valor requerido para satisfazer as restrições de igualdade geral:

$$\sum_{i=1}^{m} x_{ij} = 1,\ j = 1, 2, ..., n \qquad (2)$$

E as restrições de desigualdade linear:

$$\sum_{j=m+1}^{m+p} a_{ij}\, x_j \le b_i\ ,\ i = m+1, m+2, ..., m+p \qquad (3)$$

Com x_{ij} = 0 ou 1, i = 1, 2, ..., m; j = 1, 2, ..., n (4)

Onde se observa que:

- x_{ij} é variável de decisão zero-um, que assume o valor de 1 se o consumidor j é servido pelo i-ésimo centro de serviço e zero em outro caso, i = 1, 2, ..., m; j = 1, 2, ..., n;
- O total de carga de cada centro de serviço não deve exceder sua capacidade (restrição 3);
- Cada consumidor somente pode ser servido por um centro de serviço (restrição 4).

A função f(x) pertence a $\Re \rightarrow \Re$ e é definida inicialmente como a função de custo, sendo que, nesse caso, a mesma toma a forma de $a^T x$, a_{ij} é uma matriz (m+p)xn e b é um

Revista Eletrônica Sistemas & Gestão
Volume 9, Número 1, 2014, pp. 14-20
DOI: 10.7177/sg.2014.v9.n1.a2

vetor que pertencem a $\Re^{m+p}$ – esses parâmetros definem as características do problema. Na prática, eles são conhecidos aproximadamente, de modo que a satisfação das restrições do problema envolve um conjunto de igualdades e desigualdades e que tal problema pode ser observado como prático e realístico. De qualquer forma, a solução para o problema pode não ser única quando se deseja encontrar um ponto-solução – portanto, um critério de otimização deve ser introduzido.

Nesta pesquisa propomos uma abordagem na reformulação do problema (1), (2), (3) e (4) como um problema de otimização por meio de modelos de redes neurais (Abe, 1990). Utilizando uma função de penalidade, podemos forçar *x* a encontrar uma região factível determinada pela função $f_k(x) = 0$, enquanto minimiza a função de energia (E(x)).

Min E(x, β) (função de energia da função de penalidade) (5)

A expressão E(x): $\Re^n \to \Re$ representa a função objetivo denominada de função de energia do sistema e o fator $\beta_k > 0$ determina a quantidade de penalidade que deve ter esta função. Nesse contexto, a representação de um problema de otimização (com restrições) em uma apropriada função de custo é uma abordagem de um projeto de rede neural artificial (Anthony *et* Bartlett, 1999). Consequentemente, a construção de uma apropriada função de energia E(x) para um estado mínimo que corresponde a uma solução ótima (x*) é um pré-requisito para a formulação de um problema de otimização em termos de redes neurais.

Para o problema de minimização (1), (2), (3) e (4), uma função de energia baseada na função de penalidade pode ser construída (Cichocki *et* Unbehauen, 1992). Na pesquisa, a função de penalidade toma a forma de uma função quadrática $f(x) = x^2/2$, representada por:

$$E(x,\beta) = f(x) + \sum_k \beta_k \, v(r_k(x))$$
(5)

Onde $r_i(x)= a^T x - b_i$ representam os resíduos e $v(r_i(x))$ = Max{0, sgn($r_i(x)$ P($r_i(x)$))} são as funções de penalidade P($r_i(x)$) ≥ 0.

A minimização de x para um valor de βk representa um problema de otimização sem restrições. O mínimo no limite βk → $\infty(\forall k)$, descrito como:

$$x^* = \arg \lim_{\beta_k \to \infty(\forall k)} \min_x E(x,\beta) \qquad (6)$$

Busca satisfazer as restrições. Nesse sentido, um algoritmo é implantado para resolver a seguinte sequência de igualdades e desigualdades no problema de minimização:

1: Escolher uma sequência fixa β(0), β(1), β(2),, ..., β(k) com β(t) < β(t+1) e β(t) → ∞ .

2: Para cada β(t) encontrar um mínimo local: $x^*(\beta^{(t)}) = \arg \lim_{\beta_k \to \infty(\forall k)} \min_x E(x,\beta^{(t)})$.

3: Terminar quando $f_K\left(x^*\left(\beta^{(t)}\right)\right)$ é muito pequeno.

Na prática, o passo (2) é usualmente realizado utilizando um método iterativo. De qualquer forma, para obter o seguinte ponto mínimo x*(t+1) utilizamos x*(t) como ponto inicial com o seguinte fator de penalidade: β_k(t+1).

Aplicando o método da função de penalidade para o modelo (5) obtemos,

$$E(x,K) = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_j + \frac{K}{2}\sum_{j=1}^{n}\left(\sum_{i=1}^{m} x_j - 1\right)^2 + \frac{K}{2}\sum_{i=1}^{m}\min^2\left\{0, b_i - \sum_{j=1}^{n} r_j x_j\right\} + \frac{K}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} x_j^2\left(1 - x_j\right)^2 \qquad (7)$$

Onde o parâmetro de desequilíbrio K é o fator de penalidade e $\min^2${} é o quadrado de min{}.

Na prática é difícil encontrar uma função de penalidade que seja efetiva e eficiente para substituir as restrições que faltam. Na pesquisa de uma solução, o esforço requerido da função de penalidade para um problema dado, pode obviar no processamento de dados, um ganho em uma eventual solução de qualidade como foi observado por (Siedlecki *et* Sklansky, 1989). Muitas dificuldades aparecem porque a solução ótima, frequentemente, encontra-se na fronteira da região factível. Muitas soluções similares do genótipo da solução ótima não são confiáveis. Portanto, essa restrição às soluções prováveis torna difícil encontrar um esquema que leve dos resultados parciais a um ponto ótimo.

2. MODELO DE REDE NEURAL

Para eliminar as restrições do problema de programação inteira em outro problema equivalente de programação não linear, devemos substituir as restrições zero-um por restrições de igualdade quadráticas côncavas não lineares. Isso é realizado utilizando o método da função de penalidade, assim, a restrição do problema poderia ser simplesmente convertida em outro, da seguinte forma:

$$Min \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_j \qquad (8)$$

Revista Eletrônica Sistemas & Gestão
Volume 9, Número 1, 2014, pp. 14-20
DOI: 10.7177/sg.2014.v9.n1.a2

E está sujeito às seguintes restrições:

$$\sum_{i=1}^{m} x_{ij} = 1 \text{, j = 1, 2, ..., n} \qquad (9)$$

$$\sum_{j=1}^{n} r_j x_j \le b_i \text{, i = 1, 2, ..., m} \qquad (10)$$

x_{ij} (1- x_{ij})= 0, i = 1, 2, ..., m; j = 1, 2, ..., n (11)

O problema descrito nas equações (8-11) é equivalente ao problema original (1-4) de programação inteira zero-um. Nesse sentido, como previamente mencionado, esse problema de programação não linear pode ser reformulado como um problema de minimização sem restrições (problema quadrático côncavo com funções de igualdade nas restrições), utilizando-se o método do gradiente descendente (1, 2). A função objetivo E(x) é continuamente diferenciável para todo x ε $\Re$ n. Assim, é natural utilizar uma declive descendente baseada em um modelo de redes neurais para o problema em menção.

$$\frac{dx}{dt}(t) = -\mu \nabla E(x(t)) \text{, } x(0) = x_0 \qquad (12)$$

Onde μ > 0 é a taxa da constante de aprendizagem.

Para Cichocki *et* Unbehauen (1992), a rede neural transformada pode ser representada como um sistema de equações ordinárias diferenciáveis, como se segue:

$$\frac{du}{dt} = -\mu \left[\nabla f(x) + A^T \psi(r(x)) \right] \qquad (13)$$

$$x = g(u) \qquad (14)$$

Onde:

$$\varphi_j(x_j) = \frac{\partial}{\partial x_j} f(x) \qquad (15)$$

É a função de ativação dos neurônios de saída.

$$\psi_i(r_i) = \begin{cases} P'(r_i) & se \quad r_i \ge o \\ 0 & para\ outro\ valor \end{cases} \qquad (16)$$

É a função de ativação dos neurônios de entrada.

A rede consiste de duas camadas processando unidades – a primeira camada calcula os residuais r_i(x) e os erros $\psi(r_i(x)$, enquanto as variáveis de interesse x_j são calculadas na segunda camada, que combina e integra os erros de $\psi(r_i)$ no tempo de processamento.

Assim, a função obtida para o problema de atribuição está composta de um conjunto de equações diferenciais com a seguinte estrutura:

$$\frac{d}{dt} x_{pq} = -\mu \left(c_{pq} + K \sum_{i=1}^{m} (x_{iq} - 1) - K\, r_q \min\left\{0, b_p - \sum_{j=1}^{n} r_j x_{jp}\right\} + K\, x_{pq} (1 - x_{pq})(1 - 2x_{pq}) \right) \qquad (17)$$

Onde p = 1, 2, ..,.m e q = 1, 2, ..., n.

Em cada nó da rede, um neurônio binário é atribuído. O primeiro termo da esquerda da expressão (17) minimiza as conexões entre os nós, enquanto que os outros termos penalizam as configurações não balanceadas.

As equações diferenciais não são apresentadas porque as expressões são muito extensas. O modelo de rede neural da função de penalidade detalhado pode ser observado na Figura 1.

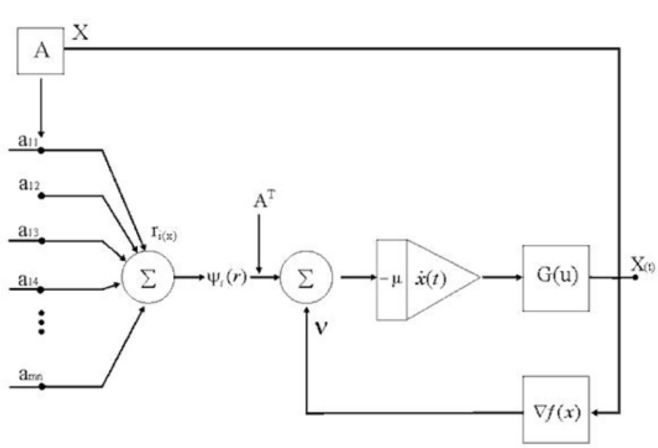

Figura 1. Modelo de rede neural detalhado.

Fonte: Modelo de Cichocki *et* Unbehauen (1992) adaptado ao problema.

Revista Eletrônica Sistemas & Gestão
Volume 9, Número 1, 2014, pp. 14-20
DOI: 10.7177/sg.2014.v9.n1.a2

A minimização da função de energia consiste na alteração de v(t) em direção a uma solução ótima, isto é, a alteração de v(t) na direção oposta ao gradiente de E(x, β). Então, as iterações sucessivas na pesquisa de v(t) levam a saída da rede para um ponto de equilíbrio que corresponde a uma solução localmente ótima do problema.

3. SIMULAÇÃO NUMÉRICA

A Tabela 1 mostra os custos unitários (reais por tonelada) de uma empresa de pneumáticos que está pensando locar seis novos armazéns capazes de absorver uma nova produção.

Tabela 1. Informações do problema de localização.

		1	2	3	4	5	6	← Almoxarifado (i)
Produção $[b_i]$ ↓		15	10	8	9	12	7	← Demanda $[r_j]$
17	A	0,95	0,6	0,9	1,65	0,35	0,25	
25	B	1,15	0,85	1,1	1,45	0,48	0,48	
13	C	1,2	0,78	0,85	1,3	0,57	0,57	
11	D	0,85	1,1	1,2	1,28	0,52	0,52	

Fonte: Problema extraído e adaptado de Moreira (2008).

A primeira aproximação é realizada aplicando o método de *Branch and Bound* no objetivo de obter a melhor localização (Tabela 2).

Tabela 2. Resultados da simulação minimizando a função objetivo.

	x_1	x_2	X_3	x_4	x_5	x_6	$\sum x_i . r_j$
A	0	1	0	0	0	1	17
B	1	0	1	0	0	0	23
C	0	0	0	0	1	0	12
D	0	0	0	1	0	0	9
$\sum x_i$	1	1	1	1	1	1	4,95

Fonte: Simulação realizada em Matlab 10.0®.

A função de penalidade (7) e o conjunto de restrições que compõem o sistema são utilizados com a finalidade de obter o mínimo da função objetivo. Na avaliação, o valor da constante de penalidade (K) é 10 (ver Tabela 3).

Tabela 3. Resultados da simulação da função de penalidade em 300 iterações.

	x_1	x_2	x_3	x_4	x_5	x_6	$\sum x_i . r_j$
A	0,45	0,07	0,56	0,04	0,4	0,15	18,14
B	0,66	0,24	0,22	0,58	0,32	0,4	25,92
C	0,2	0,79	0,04	0,05	0,37	0,05	16,46
D	0,14	0,09	0,51	0,39	0,35	0,13	15,7
$\sum x_i$	1,45	1,19	1,33	1,06	1,44	0,73	5,8156

Fonte: Simulação realizada em Matlab 10.0®.

A função que representa a rede neural de penalidade (13) é composta por um conjunto de equações diferenciais ordinárias e é utilizada para obter o mínimo da função objetivo. As equações diferenciais do sistema são resolvidas numericamente pelo método de Runge-Kutta de ordem quarta. Os resultados ajustados a valores limites da variável são apresentados na Tabela 4.

Revista Eletrônica Sistemas & Gestão
Volume 9, Número 1, 2014, pp. 14-20
DOI: 10.7177/sg.2014.v9.n1.a2

Tabela 4. Resultados da simulação da função de penalidade neural.

	x_1	x_2	x_3	x_4	x_5	X_6	$\sum x_i.r_j$
A	0	1	0	0	1	1	29
B	0	0	0	0	0	0	0
C	0	0	1	0	0	0	8
D	0	0	1	0	0	0	24
$\sum x_i$	1	0	0	1	0	0	4,18

(μ = 2, K = 10, tamanho de passo = 0,001, iterações = 300)

Fonte: Simulação realizada em Matlab 10.0®.

Em treino e teste, no caso do método da função de penalidade, a quantidade de vetores de entrada foi simulada por uma matriz de números aleatórios. Todos os processos numéricos foram interrompidos entre 300 e 1000 iterações segundo a metodologia utilizada. O sistema de equações diferenciais converge numa solução global. Em todos os casos dos experimentos foi utilizada uma taxa de aprendizado de 2. O valor do fator de penalidade considerado é 10.

Embora o processo de convergência em direção aos pontos de equilíbrio seja garantido, devemos observar que esta característica não necessariamente implica a obtenção de um ponto mínimo global em relação à função de energia ($E_K(x, \beta)$) associada à rede de Hopfield (1985). Nas simulações realizadas, a rede neural converge sempre em um ponto de energia mínima que corresponde a uma solução razoável (próxima da solução globalmente ótima) do problema de atribuição.

Na avaliação da rede neural, o valor da função objetivo compara a precisão do modelo com os resultados do modelo de referência (*Branch and Bound*), no qual a predição representa, simplesmente, os valores de todos os padrões de entrada da amostra. Um ajuste é perfeito ou o experimento é representativo quando o valor obtido para a função objetivo é menor que o valor do modelo de referência, e um ajuste pobre ocorre quando ele é maior. As amostras geradas aleatoriamente não são representativas do modelo quando as predições da rede neural não são boas e, em consequência, os valores da função objetivos são não significativos. Todas as simulações foram realizadas em Matlab 10.0®.

4. CONCLUSÕES

Nesta pesquisa temos estudado uma abordagem de rede neural com o propósito de resolver problemas de otimização não linear. A rede neural é obtida pela reformulação de um problema de minimização sem restrições por meio de uma função de penalidade.

Experimentos numéricos foram realizados a fim de observar a convergência da trajetória da rede neural. Os resultados demonstram que o modelo é eficiente na captura de padrões na alocação de recursos que têm uma grande influência em um problema de distribuição. Por outro lado, encontramos que o modelo da função de penalidade, baseado em uma estrutura de rede neural, é um bom modelo na solução de um problema de otimização combinatória pela rápida convergência que implica na estabilidade exponencial das funções assintóticas no sistema formulado. Em consequência, a metodologia utilizada é uma alternativa na solução desses problemas de otimização de forma eficiente.

A metodologia utilizada é consistente e precisa na medida em que integra o conhecimento adquirido pela inter-relação de um número de neurônios dinâmicos. Esse processo iterativo conduz a uma rápida convergência na pesquisa de um ponto mínimo local. A melhoria na otimização dos resultados é evidente em nosso exemplo. De qualquer forma, a redução no tempo de processamento requerido para obter tal precisão não foi dramática.

5. REFERÊNCIAS

Abe, S. (1990), "Theories on the Hopfield neural networks with uncertainty constrains", *Proc. Of IJCNN-90*. Washington DC, Vol.1, pp. 349-352.

Anthony, M. e Bartlett, P. L. (1999), Neural network learning: Theoretical foundations. Cambridge Univ Pr.

Balas, E. e Carrera, M.C. (1996), "A dynamic subgradient-based branch-and-bound, procedure for set covering", *Operations Research*, Vol.44, pp. 875-890.

Bjorkund, P., Varbrand, P. e Yuan, D. (2005), "Optimized planning of frequency hopping in cellular networks", *Computers and Operations Research*, Vol.32, pp. 169-186.

Chua, L. O. e Lin, G. N. (1984), "Nonlinear programming without computation, IEEE Trans", *Circuits Syst*., Vol.31, pp. 182-188.

Cichocki, A. e Unbehauen, R. (1992), "Neural networks for computing eigenvalues and eigenvectors", *Biological*

Revista Eletrônica Sistemas & Gestão
Volume 9, Número 1, 2014, pp. 14-20
DOI: 10.7177/sg.2014.v9.n1.a2

Cybernetics, Vol.68, pp. 155-164.

Farago, R. e Morabito, R. (2000), “Um método heurístico baseado em relaxação lagrangiana para o problema de carregamento de paletes do produtor”, *Pesquisa Operacional*, Vol.20, No.2, pp. 197-212.

He, B. S. (1992), “A projection and contraction method for a class of linear complementary problem and its application in convex quadratic programming”, *Applied Mathematics and Optimization,* Vol.25, pp. 247-262.

Hopfield, J. J. e Tank, D. W. (1985), “Neural computation of decisions in optimization problems”, *Biol. Cybern*., Vol.52, pp. 141-152.

Liao, L. Z. e Qi, H. (1999), “A neural network for the linear complementarity problem, Math”, *Comp Ut. Modelling*, Vol.29, No.3, pp. 9-18

Martins, A. F. T., Figueiredo, M. A. T., Aguiar, P. M. Q., Smith, N. A. e Xing, E. P. (2011), “An augmented lagrangian approach to constrained map inference”, in *Proceedings of the International Conference on Machine Learning (ICML), Bellevue,* WA, June/July, pp. 169-176.

Moreira, D. A. (2008), Administração da Produção e Operações. São Paulo: Pioneira Thomson Learning.

Oliveira, L. K. (2005), Métodos exatos baseados em relaxações lagrangiana e surrogate para o problema de carregamento de paletes do produtor, Tese de Doutorado em Engenharia de Produção, Universidade de São Paulo, São Carlos, SP.

Siedlecki, W. e Sklansky, J. (1989), “Constrained genetic optimization via dynamic reward-penalty balancing and its use in pattern recognition”, in *Proceeding of the third International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, pp. 141-150.

Tank, D. W. e Hopfield, J. J. (1986), “Simple neural optimization networks: An A/D convert, signal decision circuit, and a linear programming circuit”, *IEEE Trans.Circuits Syst*., Vol.33, pp. 533-541.

Tosserams, S., Etman, L., Papalambros, P. e Rooda, J. (2006), “An augmented lagrangian relaxation for analytical target cascading using the alternating direction method of multipliers”, *Structural and Multidisciplinary Optimization*, Vol.31, No.3, pp. 176-189.

Yanover, C., Meltzer, T. e Weiss, Y. (2006), “Linear programming relaxations and belief propagation - an empirical study”. *Journal of Machine Learning Research*, Vol.7, pp.1887-1907.