

UMA ABORDAGEM NEURO-IMUNE PARA A SOLUÇÃO DO PROBLEMA DE MÚLTIPLOS CAIXEIROS VIAJANTES

THIAGO A. S. MASUTTI, LEANDRO N. DE CASTRO

*Laboratório de Sistemas Inteligentes, Programa de Mestrado em Informática
Universidade Católica de Santos*

R. Dr. Carvalho de Mendonça, 144, Vila Mathias, 11070-906

E-mails: {tmasutti, lnunes}@lsin.unisantos.br

Abstract— Two types of neural networks are usually applied to solve combinatorial optimization problems: networks based on the optimization of energy surfaces, such as Hopfield networks; and self-organized networks, like the self-organizing maps. This paper presents a hybrid algorithm between a self-organizing map and artificial immune systems to solve the multiple traveling salesmen problem (MTSP). The network investigated, named RABNET-MTSP, has a single layer of neurons that subdivide into several sub-networks, being each one of them responsible for representing one of the m salesmen that will solve the problem. It is also important to remark that this type of network is unsupervised and, thus, do not use information about the quality of the solutions proposed. Even with such characteristic, these networks present good quality (competitive) results when compared with other techniques from the literature.

Keywords— Artificial Neural Networks, Self-Organizing Maps, Artificial Immune Systems, Multiple Traveling Salesmen Problem.

Resumo— Dois tipos de redes neurais são comumente usadas para resolver problemas combinatórios: redes baseadas em otimização de superfícies de energia, como as redes de Hopfield; e redes auto-organizadas, como os mapas auto-organizáveis. Este trabalho apresenta um algoritmo híbrido entre uma rede neural auto-organizada e sistemas imunológicos artificiais para a solução do problema de múltiplos caixeiros viajantes (MTSP). A rede investigada, nomeada de RABNET-MTSP, possui uma única camada de neurônios que se subdividem em diversas sub-redes, sendo que cada uma delas é responsável por representar um dos m caixeiros que resolverão o problema. Outro aspecto relevante é o fato de que a rede opera de forma totalmente não-supervisionada, ou seja, não há informação sobre a qualidade das soluções sendo propostas. Mesmo assim, os resultados apresentados são de boa qualidade e competitivos em relação a outras ferramentas da literatura.

Palavras-chave— Redes Neurais Artificiais, Mapas Auto-Organizáveis, Sistemas Imunológicos Artificiais, Problema de Múltiplos Caixeiros Viajantes.

1 Introdução

O Problema de Múltiplos Caixeiros Viajantes (MTSP, do inglês *Multiple Traveling Salesmen Problem*) é uma generalização do já bem conhecido TSP, no qual é possível a utilização de mais de um caixeiro para se obter uma solução. O MTSP pode ser descrito da seguinte maneira (Bektas, 2006): dado um conjunto de cidades, m representa o número de caixeiros que se encontram na cidade depósito. As cidades restantes podem ser chamadas de cidades intermediárias. Sendo assim, o MTSP consiste em determinar uma rota para todos os m caixeiros, que devem partir e voltar ao depósito, de tal maneira que todas as cidades intermediárias sejam visitadas apenas uma vez e por apenas um caixeiro e que o custo total destas rotas seja minimizado. O custo pode ser, por exemplo, a distância percorrida ou o tempo de viagem.

Enquanto o objetivo geral do MTSP é minimizar o custo total, podendo ser chamado de critério *min-sum*, o critério *minmax* objetiva minimizar o custo da maior rota, que é mais apropriado aos problemas reais nos quais a distribuição das tarefas deva ser equitativa (França *et al.*, 1995).

O MTSP é um problema que pode ser facilmente relacionado a aplicações de mundo real como, por exemplo, planejar a ordem de execução de determinadas tarefas que um grupo de robôs autônomos tem

que executar (Zhong *et al.*, 2002); roteamento de veículos como, por exemplo, ônibus escolares (Angel *et al.*, 1972); e determinação da ordem de visitaç o de centrais telefônicas por um grupo de técnicos (Svestka e Rinnooy Kan, 1975).

Os Mapas Auto-Organizáveis (SOM, do inglês *Self-Organizing Maps*) de Kohonen (de Castro, 2006; Haykin, 1999; Kohonen, 2000) foram temas de diversos trabalhos para a solução do TSP (Angeniol *et al.*, 1988; Pasti e de Castro, 2006; Somhom *et al.*, 1997) e os resultados destes trabalhos são muito satisfatórios. Porém, pouca atenção foi dada à aplicação de redes neurais auto-organizáveis para a solução do MTSP. Neste artigo é apresentada uma ferramenta baseada em mapas auto-organizáveis e sistemas imunológicos artificiais (de Castro e Timmis, 2002) para a solução do MTSP, a RABNET-MTSP, que é uma variação do algoritmo proposto por Pasti e de Castro (2006).

Este artigo está organizado da seguinte maneira. A Seção 2 apresenta alguns trabalhos baseados em redes competitivas para a solução do MTSP e a Seção 3 descreve a RABNET-MTSP. Na Seção 4.1 são propostos valores padrões para os parâmetros da RABNET-MTSP e na Seção 4.2 são apresentados os resultados obtidos com o algoritmo proposto. O trabalho é concluído na Seção 5 com uma discussão e propostas de trabalhos futuros.

2 Revisão da Literatura

Esta seção faz uma breve descrição de alguns algoritmos baseados em redes auto-organizadas para a solução do MTSP presentes na literatura. Em geral, tais algoritmos são modificações daqueles aplicados à solução do TSP. Dentre os trabalhos descritos, Somhom *et al.* (1999) é o único a utilizar instâncias publicamente acessíveis para avaliar o desempenho de seu algoritmo e, por isso, seus resultados serão usados como benchmark para a ferramenta proposta aqui.

Baseando-se no trabalho de Angeniol *et al.* (1988), Hsu *et al.* (1991) propuseram um algoritmo para a solução do MTSP. Neste algoritmo, m ‘bolhas’ crescem a partir do depósito em direção às cidades intermediárias definindo as rotas que cada um dos m caixeiros deverá seguir. Assim como o algoritmo de Angeniol, o algoritmo proposto possui funções de criação e exclusão de nós.

Vakhutinsky e Golden (Vakhutinsky e Golden, 1994) propuseram uma extensão da *rede elástica* (Durbin e Willshaw, 1987) para a solução do MTSP. Nesse algoritmo m ‘tiras elásticas’ de neurônios se alongam a partir do depósito em direção às cidades intermediárias de acordo com duas forças: a primeira induz todos os nós da rede a se moverem na direção da cidade mais próxima, a segunda força faz com que cada nó se mova na direção de seus vizinhos.

Partindo de um trabalho posterior para o TSP (Somhom *et al.*, 1997), Somhom *et al.* (1999) propuseram uma rede baseada em SOM para a solução do *minmax* MTSP. No algoritmo é utilizado um número fixo M de neurônios, sendo $M = 2*N$, onde N é o número de cidades a serem visitadas. Além disso, há uma restrição na regra de competição deste algoritmo para propiciar custos equitativos das rotas de cada caixeiro. Para avaliar o desempenho do algoritmo, os autores realizaram testes com instâncias para o C-VRP da TSPLIB (Reinelt, 1991) e compararam os resultados com a *Elastic Net* e com o algoritmo 2-opt (Potvin *et al.*, 1989). De acordo com os resultados, o algoritmo proposto pelos autores apresentou soluções de melhor qualidade que os outros. Em relação ao esforço computacional, o tempo de processamento do algoritmo proposto foi inferior ao da *Elastic Net* e superior ao do 2-opt.

3 A RABNET-MTSP

A RABNET-MTSP (*real-valued antibody network to the multiple traveling salesmen problem*) é uma rede neural que combina características de um SOM e sistemas imunológicos artificiais (de Castro e Timmis, 2002) para a solução do MTSP. Esta rede é uma modificação da RABNET-TSP (Pasti e de Castro, 2006). Dentre as principais características deste algoritmo é possível destacar:

- Arquitetura do tipo *feedforward* com uma única camada de pesos a serem ajustados.
- Rede competitiva com aprendizado não-supervisionado.
- Arquitetura construtiva através de um mecanismo de clonagem de neurônios.
- Fase de poda de neurônios ao final do treinamento.
- Vizinhança pré-definida entre os neurônios.

O objetivo da RABNET-MTSP é, durante a fase de treinamento, posicionar uma célula presente nas sub-redes (cada sub-rede corresponde à rota de um caixeiro distinto) em cada cidade da instância do MTSP a ser resolvida. Sendo assim, a vizinhança pré-definida entre os neurônios das sub-redes determinará a ordem de visitação das cidades.

A RABNET-MTSP é um algoritmo não-supervisionado, ou seja, não utiliza uma função de custo ou avaliação, durante a fase de treinamento da rede, para determinar a qualidade da solução.

A seguir são descritas as principais etapas da RABNET-MTSP.

3.1 Inicialização das sub-redes

Para a RABNET-MTSP, cada sub-rede representará um caixeiro, que deverá percorrer uma rota independente dos outros caixeiros. Sendo assim, são inicializadas m sub-redes. Inicialmente são inseridos dois neurônios em cada sub-rede. O primeiro contém um vetor de coordenadas idêntico ao do depósito e o segundo é inicializado randomicamente. Para facilitar a implementação e a identificação do neurônio vinculado ao depósito, o neurônio de índice um é aquele que sempre estará vinculado ao depósito.

3.2 Apresentação dos padrões de entrada

Cada cidade representa um padrão de entrada, que nos sistemas imunológicos é correspondente a um antígeno. Sendo assim, todos os antígenos devem ser apresentados à rede e isto é feito iterativamente, ou seja, um antígeno de cada vez. A apresentação de todos os antígenos é caracterizada como uma época. Para diminuir a dependência do algoritmo de treinamento da rede em relação à ordem inicial destes antígenos, a cada época esta ordem é alterada aleatoriamente.

3.3 Competição

O objetivo dessa etapa é calcular o neurônio contido em uma das m sub-redes que apresente a maior afinidade (menor distância) ao antígeno apresentado, sendo este neurônio caracterizado como o vencedor para a competição.

A regra que determina o neurônio de maior afinidade ao antígeno apresentado pode ser definida pela equação a seguir:

$$I, J = \arg \min (\| \mathbf{w}_i^j - \mathbf{x} \|), \forall i \in j, j = 1, \dots, m, \quad (1)$$

na qual \mathbf{w} é o vetor de coordenadas do neurônio de índice i pertencente à sub-rede de índice j , \mathbf{x} é o vetor de coordenadas do antígeno apresentado e I é o índice do neurônio vencedor pertencente à sub-rede de índice J .

O algoritmo de treinamento não apresenta restrição alguma quanto ao número de vezes que um neurônio pode vencer uma competição. Sendo assim, um neurônio pode estar relacionado a zero, um ou mais antígenos. Um vetor de concentração, \mathbf{p}^m , armazena, para cada um dos m caixeiros, a quantidade de vezes que cada neurônio venceu uma competição. Esta informação é reinicializada a cada época e é importante para outras etapas como, por exemplo, o crescimento das sub-redes. Quando o antígeno apresentado se refere à cidade base, não há competição. Porém, os neurônios de índice um de cada sub-rede são utilizados nas próximas etapas.

3.4 Cooperação

A fase de cooperação da RABNET-MTSP é igual à de um SOM. Nesta fase, o estímulo causado ao neurônio vencedor é propagado aos neurônios vizinhos, mas de uma forma mais branda conforme a distância lateral entre eles aumenta.

Aqui, o estímulo a um neurônio se refere à sua adaptação, ou seja, mover o neurônio na direção do antígeno para o qual ele venceu. Desta maneira, os vizinhos do neurônio vencedor também serão movidos na direção do antígeno, mas com um passo menor do que o dado por ele. Na RABNET-MTSP os neurônios de cada sub-rede estão dispostos em uma vizinhança circular e a distância lateral entre dois neurônios pode ser definida pela seguinte equação:

$$d = \min(|i - I|, M - |i - I|), \quad i \in J, \quad (2)$$

na qual i é o índice do neurônio analisado, I é o índice do neurônio vencedor pertencente à sub-rede J e M é o número total de neurônios desta sub-rede.

Para caracterizar essa diminuição do passo de um neurônio conforme sua distância lateral com o neurônio vencedor aumenta, utiliza-se uma função de vizinhança que, na RABNET-MTSP, é a mesma utilizada na RABNET-TSP e proposta por Haykin (1999):

$$h = \begin{cases} f(\sigma, d) = \exp(-d^2/2\sigma^2) & i \neq 1 \\ 0 & \text{caso contrário} \end{cases}, \quad (3)$$

na qual h representa o tamanho do passo que o neurônio i dará em relação ao passo do neurônio I , σ é o parâmetro que controla a influência da vizinhança e d é a distância lateral entre os neurônios i e I .

Conforme a Eq. (3), o neurônio vinculado ao depósito permanece sempre fixo, pois seu valor de h será sempre zero.

O parâmetro σ é responsável por determinar a influência da vizinhança. É comum que este parâmetro,

a partir de um valor inicial, seja reduzido ao decorrer das épocas. Na RABNET-MTSP utiliza-se a função de redução proposta por Haykin (1999):

$$\sigma_t = \sigma_0 * \exp(-t/\tau_1), \quad (4)$$

na qual σ_0 é o valor inicial para este parâmetro, t é a época atual e $\tau_1 = 1000/\log(\sigma_0)$.

Esta etapa da RABNET-MTSP é descartada assim que é definida a estabilização dos vencedores, que será descrita adiante.

3.5 Adaptação

A fase de adaptação se refere a mover os neurônios na direção do antígeno apresentado. Na RABNET-MTSP a adaptação de um neurônio pode ser definida pela seguinte equação:

$$\mathbf{w}_i^j(t+1) = \begin{cases} \mathbf{w}_i^j(t) + \alpha * h_{ii} * (\mathbf{x} - \mathbf{w}_i^j(t)) & h_{ii} > \kappa \\ \mathbf{w}_i^j(t) & \text{caso contrário} \end{cases}, \quad (5)$$

na qual \mathbf{w} é o vetor de coordenadas do neurônio de índice i pertencente à sub-rede de índice J , \mathbf{x} é o vetor de coordenadas do antígeno apresentado, α é a taxa de aprendizagem, h é o parâmetro definido na Eq. (3), t é a época atual e κ é o limiar para h .

O parâmetro α representa o passo que será dado pelo neurônio. Este parâmetro, a partir de seu valor inicial, é reduzido ao decorrer das épocas. Para isto foi adotada a função proposta por Haykin (1999):

$$\alpha_t = \alpha_0 * \exp(-t/\tau_2), \quad (6)$$

na qual α_0 é o valor inicial para este parâmetro, t é a época atual e $\tau_2 = 1000$.

Diferentemente do algoritmo proposto por Pasti e de Castro (2006), na RABNET-MTSP apenas neurônios que obtiveram um valor mínimo para h na etapa de cooperação são adaptados. Esta restrição é imposta através da inclusão do limiar κ na Eq. (5) e faz com que apenas mudanças realmente significativas sejam realizadas na rede.

Conforme alguns resultados de testes preliminares, esta restrição não produz efeito significativo na qualidade da solução, porém diminui o esforço computacional do algoritmo.

3.6 Crescimento das sub-redes

A etapa de crescimento é baseada em sistemas imunológicos artificiais (de Castro e Timmis, 2002), na qual os neurônios mais estimulados durante uma época são selecionados para serem clonados. Para cada sub-rede é selecionado o neurônio mais estimulado, que é aquele que reconhece a maior concentração de antígenos. A indicação do neurônio de maior concentração de cada sub-rede pode ser definida pela equação abaixo:

$$k^j = \arg \max(\mathbf{p}^j) \quad (7)$$

na qual k é o índice do neurônio de maior concentração relativo à sub-rede j e \mathbf{p} é o vetor de concentra-

ção dessa sub-rede. Caso dois ou mais neurônios apresentem a maior concentração, um deles é escolhido aleatoriamente.

O neurônio de índice k selecionado através da Eq. (7) será utilizado para clonagem somente se $\rho_k^j > 1$. Caso contrário, a sub-rede à qual pertence este neurônio não sofrerá alterações.

Dentre os neurônios selecionados como candidatos à clonagem, de todos os antígenos relacionados a ele, aquele de maior distância Euclidiana (menor afinidade) é escolhido. Se esta distância for maior que o parâmetro ε , então este neurônio é clonado, caso contrário, nenhuma mudança nessa sub-rede será realizada. O novo neurônio criado terá um vetor de coordenadas idêntico ao de seu 'neurônio pai' e será introduzido na rede como vizinho de grau um dele.

3.7 Estabilização dos vencedores

Esta é uma etapa que não existe no algoritmo original proposto por Pasti e de Castro (2006) e foi introduzida para propiciar uma diminuição no tempo de processamento do algoritmo. Para que essa diminuição ocorra, a etapa de cooperação é descartada assim que é caracterizada a estabilização dos vencedores, fazendo com que apenas os neurônios vencedores sejam movidos na direção do antígeno.

A estabilização dos vencedores ocorre quando há apenas uma pequena variação dos neurônios vencedores para cada antígeno. Esta variação é dada pela seguinte equação:

$$\Delta V = \sum_{n=1}^N |I_n(t-1) - I_n(t)|, \quad (8)$$

na qual ΔV é o grau de variação dos vencedores, N é o número total de antígenos, I é o índice do neurônio vinculado ao antígeno de índice n e t é a época atual.

Para se caracterizar a estabilização dos vencedores, é necessário que ΔV assuma um valor menor ou igual a um determinado limiar β .

3.8 Critério de parada

Para garantir que o algoritmo obtenha uma solução factível, cada antígeno deve ter um neurônio a uma distância mínima λ . Quando esta condição é satisfeita, o processo de aprendizado da rede é finalizado.

3.9 Poda

Ao término do processo de treinamento da rede, todo neurônio com nível de concentração igual a zero é excluído. Isso faz com que o número final de neurônios seja igual a $N + m - 1$, e a vizinhança definida pelos neurônios de cada sub-rede determinará a ordem da rota de cada caixeiro.

4 Avaliação da Performance do Algoritmo

4.1 Definição dos Parâmetros de Treinamento

Para que a RABNET-MTSP possa ser aplicada a uma dada instância do MTSP, alguns parâmetros devem ser definidos. Tais parâmetros podem influenciar a maneira com que as sub-redes se desenvolvem e, conseqüentemente, o esforço computacional e/ou a qualidade da solução. A Figura 1 apresenta o comportamento do algoritmo durante o processo de aprendizagem, com $m = 4$, para uma das instâncias analisadas.

Assim como proposto em Pasti e de Castro (2006), neste trabalho foi utilizado $\varepsilon = md * 0,2$ e $\lambda = md * 0,01$, nas quais md representa a menor distância entre as cidades da instância analisada.

Os parâmetros α_0 e σ_0 podem influenciar a qualidade da solução que será obtida e seus valores podem variar de acordo com as características de cada instância. Portanto, os valores ideais destes parâmetros só podem ser definidos a partir de testes preliminares. Quando não for possível a realização destes testes, propomos utilizar $\alpha_0 = 0,75$ e $\sigma_0 = 19$, valores obtidos empiricamente para algumas instâncias da literatura. Sugerimos também $\kappa = 0,01$, caracterizando como significativo um passo maior que 1% do passo dado pelo neurônio vencedor.

O valor de β depende do valor adotado para α_0 , seguindo a seguinte equação:

$$\beta = \begin{cases} 0,05 * N & \alpha_0 \geq 0,7 \\ -1 & \text{caso contrário} \end{cases} \quad (9)$$

na qual N é o número de cidades da instância analisada. O valor negativo para β faz com que não seja caracterizada a estabilização dos vencedores.

4.2 Resultados Computacionais

Nos testes apresentados aqui foram utilizadas instâncias do Problema de Roteamento de Veículos Capacitados (CVRP, do inglês *Capacitated Vehicle Routing Problem*) da TSPLIB (Reinelt, 1991) sem considerar a restrição de capacidade do veículo. Também foram consideradas algumas instâncias para o problema do caixeiro viajante da TSPLIB. As instâncias utilizadas para este trabalho são as mesmas que foram utilizadas no trabalho de Somhom *et al.* (1999) e são publicamente acessíveis através do site <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>.

O algoritmo foi implementado em MATLAB e executado em um Pentium IV de 3,0GHz com 1Gb de RAM. Para cada instância analisada o algoritmo foi executado 30 vezes para dois, três e quatro caixeiros respectivamente. A Tabela 1 apresenta o esforço computacional da RABNET-MTSP e compara os resultados, para o critério *minmax*, com aqueles apresentados em Somhom *et al.* (1999).

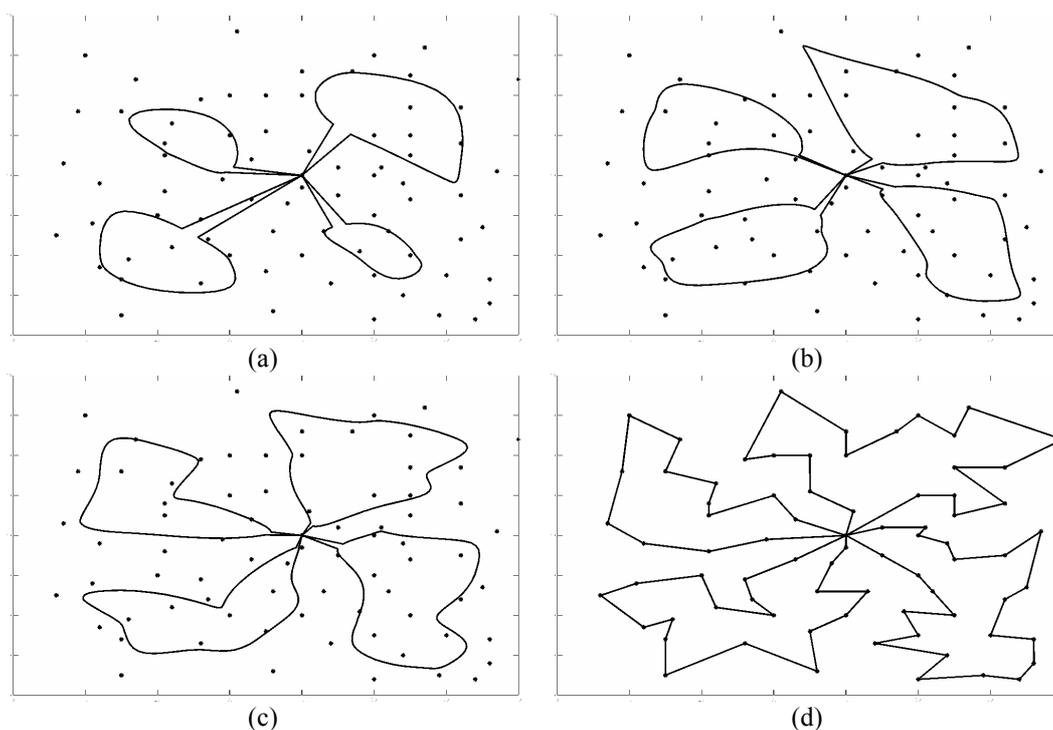


Figura 1 – Evolução das sub-redes que representam os 4 caixeiros para uma das instâncias analisadas. (a) 50 épocas; (b) 100 épocas; (c) 150 épocas; (d) resultado final com 225 épocas.

Tabela 1 – Esforço computacional da RABNET-MTSP e comparação das soluções obtidas com as apresentadas por Somhom *et al.* (1999). Tempo é o tempo médio, em segundos, para uma execução; Menor é o custo obtido para a melhor solução e Média é a média do custo das soluções obtidas. Os valores grifados destacam as melhores soluções entre os dois algoritmos.

Instância	N	m	RABNET-MTSP			Somhom <i>et al.</i>	
			Tempo	Menor	Média	Menor	Média
eil22	22	2	8,41	159	163,13	157	158,50
		3	9,39	120	161,33	117	119,42
		4	10,95	113	127,93	111	112,50
eil30	30	2	15,05	233	233,00	230	232,00
		3	17,83	228	228,00	174	180,00
		4	20,52	228	228,00	171	174,75
eil51	51	2	25,31	230	234,70	247	248,67
		3	28,70	167	184,50	170	172,00
		4	24,04	135	151,67	136	137,33
eil76	76	2	32,05	280	293,87	289	292,00
		3	44,61	207	222,70	205	210,50
		4	42,45	158	178,73	159	162,75
eil101	101	2	43,08	329	350,80	340	344,67
		3	49,41	237	257,53	232	236,00
		4	59,07	187	205,27	187	189,67
kroA100	100	2	39,12	11484	12457,37	11484	11532,80
		3	88,38	9350	9668,70	9062	9276,25
		4	55,96	7512	8466,37	7497	7516,75
kroA150	150	2	83,68	13984	15324,03	14885	15076,00
		3	145,59	11078	11557,70	10527	10756,00
		4	100,22	8511	8890,03	8571	8625,60
kroA200	200	2	114,01	15728	16346,27	17353	17547,50
		3	133,94	11378	12440,83	11502	11722,00
		4	151,01	9521	11001,50	10433	10776,33
fl417	417	2	301,85	7488	8365,50	7207	7266,75
		3	361,70	5490	6755,87	5618	5902,50
		4	400,97	5063	5367,97	5032	5109,50

5 Discussões e Perspectivas Futuras

Redes neurais artificiais têm sido amplamente estudadas, em especial para a solução de problemas de otimização combinatória como, por exemplo, TSP e MTSP. Este artigo apresentou uma abordagem baseada em Mapas Auto-Organizáveis e sistemas imunes artificiais para a solução do MTSP. A ferramenta proposta, denominada RABNET-MTSP, foi capaz de apresentar boas soluções para o *minmax* MTSP com um tempo de processamento razoável.

Uma comparação entre as soluções obtidas com a RABNET-MTSP e com resultados apresentados na literatura demonstrou um bom desempenho do algoritmo proposto.

Os resultados encorajadores apresentados neste artigo confirmam a efetividade da aplicação de algoritmos auto-organizados para a solução de problemas combinatórios e nos motivam a realizar adaptações no algoritmo para a solução de outro problema, o CVRP. Além disso, um estudo mais profundo com mais instâncias para o MTSP será alvo de investigação futura.

Agradecimentos

Os autores agradecem à UNISANTOS, à FAPESP e ao CNPq pelo apoio financeiro.

Referências Bibliográficas

Angel, R.D., Caudle, W.L., Noonan, R. e Whinston, A. (1972). Computer-Assisted School Bus Scheduling, *Management Science*, **18**(6): B279-B288.

Angeniol, B., Croix Vaubois, G. e Le Texier, J-Y. (1988). Self-Organizing Feature Maps and the Traveling Salesman Problem, *Neural Networks*, **1**(4): 289-293.

Bektas, T. (2006). The Multiple Traveling Salesman Problem: An Overview of Formulations and Solution Procedures, *Omega*, **34**(3): 209-219.

de Castro, L.N. e Timmis, J. (2002). Artificial Immune Systems: A New Computational Intelligence Approach, Springer-Verlag.

de Castro, L.N., (2006). Fundamentals of Natural Computing: Basic Concepts, Algorithms, and Applications, CRC Press LLC.

Durbin, R. e Willshaw, D. (1987). An Analogue Approach to the Traveling Salesman Problem Using an Elastic Net Method, *Nature*, **326**: 689-691.

França, P.M., Gendreau M., Laporte, G. e Müller, F.M. (1995). The m-Traveling Salesman Problem with Minmax Objective, *Transportation Science*, **29**(3): 267-275.

Haykin, S. (1999). Neural Networks: A Comprehensive Foundation, 2nd Ed., Prentice Hall.

Hsu, C., Tsai, M. e Chen, W. (1991). A Study of Feature-Mapped Algorithm for the Multiple Traveling Salesmen Problem, *IEEE International Symposium on Circuits and Systems*, vol. 3, pp. 1589-1592.

Kohonen, T. (2000). Self-Organizing Maps, 3rd Ed., Springer-Verlag.

Pasti, R. e de Castro, L.N. (2006). A Neuro-Immune Network for Solving the Traveling Salesman Problem, *International Joint Conference on Neural Networks*, pp. 3760-3766.

Potvin, J., Lapalme, G. e Rousseau, J. (1989). A Generalized k-opt Exchange Procedure for the MTSP, *Information Systems and Operations Research*, **27**(4): 474-481.

Reinelt, G. (1991). TSPLIB - A Traveling Salesman Problem Library, *ORSA Journal on Computing*, **3**(4): 376-384.

Somhom, S., Modares, A. e Enkawa, T. (1997). A Self-Organising Model for the Traveling Salesman Problem, *Journal of the Operational Research Society*, **48**(9): 919-928.

Somhom, S., Modares, A. e Enkawa, T. (1999). Competition-Based Neural Network for the Multiple Travelling Salesmen Problem with Minmax Objective, *Computers and Operations Research*, **26**(4): 395-407.

Svestka, J.K. e Rinnooy Kan, A.H.G. (1975). Some Simple Applications of the Traveling Salesman Problem, *Operational Research Quarterly*, vol. 26, pp. 717-733.

Vakhutinsky, A.I. e Golden, B.L. (1994). Solving Vehicle Routing Problems Using Elastic Nets, *IEEE International Conference on Neural Network*, pp. 4535-4540.

Zhong, Y., Liang, J., Guochang, G., Zhang, R. e Yang, H. (2002). An Implementation of Evolutionary Computation for Path Planning of Cooperative Mobile Robots, *Proceedings of the 4th World Congress on Intelligent Control Automation*, vol. 3, pp. 1798-1802.