

# 改进的连续 Hopfield 网络求解组合优化问题

## ——以 TSP 求解为例

邱树伟

(汕头职业技术学院 计算机系, 广东 汕头 515078)

**摘要:** 利用神经网络求解组合优化问题, 是一种有效的途径。对连续 Hopfield 网络的数学模型及稳定性进行了分析; 探讨了组合优化问题的神经网络求解方法, 针对传统方法参数配置复杂、收敛速度慢等不足, 提出了改进算法; 最后, 通过系统仿真与性能测试验证了该算法的可行性。

**关键词:** 组合优化; 神经网络; 连续 Hopfield 网络; 旅行商问题

**中图分类号:** TP183

**文献标志码:** A

**文章编号:** 1673-9833(2011)03-0042-05

## Improved Continuous Hopfield Neural Network for Solving Combinatorial Optimization Problems: An Example to Solve the TSP

Qiu Shuwei

(Department of Computer Science, Shantou Polytechnic, Shantou Guangdong 515078, China)

**Abstract:** Using neural networks to solve combinatorial optimization problems is an effective approach. Analyzes the mathematical model and stability of continuous Hopfield neural network, discusses the use of CHNN to solve combinatorial optimization problem and puts forward the improved algorithm aiming at the insufficiency of the traditional method of parameters configuration complex and convergence rate slower. Finally, through the system simulation and performance testing, demonstrates the algorithm feasible.

**Keywords:** combinatorial optimization; neural network; continuous hopfield neural network; traveling salesman problem

人工神经网络是目前国际上迅速发展的前沿研究方向之一, Hopfield 网络作为一种全连接型的神经网络, 利用与阶层型神经网络不同的结构特征和学习方法, 模拟生物神经网络的记忆机理, 在解决组合优化问题方面取得了很好的效果<sup>[1]</sup>。组合优化问题是一类在社会生产中普遍存在的问题, 求解较为困难。由于神经网络具有内在的并行性和容错性, 因此, 在求解组合优化问题领域得到了广泛应用<sup>[2]</sup>。本文以 TSP (traveling salesman problem, 旅行商问题) 求解为例, 讨论了利用连续 Hopfield 网络解决组合优化问题的一般方法。针对传统方法参数配置复杂、

收敛速度较慢<sup>[3]</sup>、易出现等价解等不足, 提出了改进算法。同时, 从参数配置、问题规模及拓扑结构等方面对算法性能进行测试。

## 1 连续 Hopfield 网络

1984 年, 美国科学家 Hopfield 设计了 Hopfield 网络模型的电路, 指出神经元可以用运算放大器来实现, 所有神经元的连接可用电子线路来模拟, 称为连续 Hopfield 网络<sup>[4]</sup>。与离散 Hopfield 网络不同, 连续 Hopfield 网络采用 Sigmoid 函数代替阈值函数作为

收稿日期: 2011-03-10

作者简介: 邱树伟 (1979-), 男, 广东潮州人, 汕头职业技术学院讲师, 硕士, 主要研究方向为网络与分布计算, 人工智能,

E-mail: city-qsw@163.com

神经元的激活函数，在时间上的连续性决定了其工作方式的并行性，提高了解决问题的效率。

### 1.1 数学模型构建

假设神经元 $j(j=1, 2, \dots, N)$ 的内部膜电位状态用 $U_j$ 表示，细胞膜输入电容为 $C_j$ ，细胞膜的传递电阻为 $R_j$ ，输出电压为 $V_j$ ，外部输入电流用 $I_j$ 表示。其中， $R_j$ 和 $C_j$ 并联模拟生物神经元的时延常数， $w_{ij}$ 模拟神经元间的突触特性，运算放大器模拟神经元的非线性特性， $I_j$ 相当于阈值。由基尔霍夫电流定律可得网络动态方程为

$$\begin{cases} C_j \frac{dU_j(t)}{dt} = \sum_{i=1}^N w_{ij} V_i(t) - \frac{U_j(t)}{R_j} + I_j, \\ V_j(t) = g_j(U_j(t)), \end{cases} \quad (1)$$

式中： $N$ 为神经网络中神经元的个数；

$g_j$ 为神经元的传递函数；

$W = (w_{ij}) (i, j=1, 2, \dots, N)$ 为网络权系数矩阵<sup>[5]</sup>。

### 1.2 稳定性分析

网络的能量函数定义为

$$E(t) = -\frac{1}{2} \sum_{j=1}^N \sum_{i=1}^N w_{ij} V_i(t) V_j(t) - \sum_{j=1}^N V_j(t) I_j + \sum_{j=1}^N \frac{1}{R_j} \int_0^{V_j(t)} g^{-1}(V) dV, \quad (2)$$

式中： $g^{-1}(V)$ 是 $V_j(t) = g_j(U_j(t))$ 的反函数。

对 $E(t)$ 求时间导数 $\frac{dE(t)}{dt}$ ，如果 $w_{ij} = w_{ji}$ ，则

$$\frac{dE(t)}{dt} = -\sum_{j=1}^N \frac{dV_j(t)}{dt} \left[ \sum_{i=1}^N w_{ij} V_i(t) + I_j - \frac{U_j(t)}{R_j} \right]. \quad (3)$$

将动态方程式(1)代入式(3)，同时，由于 $V_j(t) = g_j(U_j(t))$ ，则有

$$\frac{dE(t)}{dt} = -\sum_{j=1}^N \left( \frac{dV_j(t)}{dt} \right)^2 \times C_j \times \frac{dg_j^{-1}(V_j(t))}{dt}. \quad (4)$$

若传递函数 $g(u)$ 为单调递增的连续有界函数，则其反函数也为单调递增函数，故其导数必定大于0，

即 $\frac{dg_j^{-1}(V_j(t))}{dt} > 0$ 。同时， $C_j > 0$ ， $\frac{dV_j(t)}{dt} \geq 0$ ，因此

$\frac{dE(t)}{dt} \leq 0$ ，且仅当 $\frac{dV_j(t)}{dt} = 0$ 时，有 $\frac{dE(t)}{dt} = 0$ 。

从以上分析可知：当网络神经元的传递函数单调递增且网络权系数矩阵对称时，网络能量会随着时间变化下降或保持不变；当且仅当神经元的输出不再随时间变化时，网络的能量才会不变<sup>[5]</sup>。

## 2 组合优化问题的求解方法及改进

### 2.1 组合优化问题的求解原理

组合优化问题的目标是从问题的可行解集中求出最优解<sup>[6]</sup>，可描述为：令 $\Omega = \{s_1, s_2, \dots, s_N\}$ 为所有状态构成的解空间， $C(s_i)$ 为状态 $s_i$ 对应的目标函数值，要求寻找最优解 $s^*$ ，使得对于所有的 $s_i \in \Omega$ ，有 $C(s^*) = \min(C(s_i))$ 。比较典型的组合优化问题有旅行商问题、加工调度问题、邮路选择问题、空中航线规划问题等，这些问题有很强的工程代表性，在实际社会生产活动中有很广泛的应用。但最优化求解却较为困难，主要原因是求解此类问题需要较大的时空开销，以致某些大规模问题不可能在可接受的时间里获得求解。因此，研究组合优化问题的求解方法具有重要的理论价值和实际意义。

利用连续 Hopfield 网络可较为快速地求解组合优化问题<sup>[7]</sup>，其原理是将问题的目标函数转换成网络的能量函数，把问题变量映射为神经元的状态，当神经元状态趋于平衡点时，能量函数也趋于极小值。网络由初态向稳态收敛的过程就是优化计算的过程，当能量函数收敛于极小值时，问题的最优解（或近似最优解）也就产生了。由于神经网络是并行计算的，其计算量不会随问题维数的增加而发生指数性“爆炸”，因而对于组合优化问题的快速求解非常有效。

### 2.2 连续 Hopfield 网络求解 TSP 的传统算法

TSP 问题是组合优化问题的典型代表，可描述为：已知 $N$ 个城市间的距离，某一旅行商从某个城市出发访问每个城市1次且仅1次，最后回到出发城市，怎样安排才使其所走路线最短<sup>[8]</sup>。解决问题的方法是先将 TSP 的合法解映射为置换矩阵，并给出相应的能量函数，同时将满足置换矩阵的能量函数最小值与 TSP 问题的最优解相对应。

使用 $N \times N$ 神经网络，用神经元的状态来表示某一条有效路径中的位置。神经元 $x_i$ 的状态用 $v_{xi}$ 表示，其中 $x, i \in \{1, 2, \dots, N\}$ ， $v_{xi}$ 表示第 $x$ 个城市 $C_x$ 在路径中的第 $i$ 个位置出现， $v_{xi} = 0$ 表示 $C_x$ 在路径中第 $i$ 个位置不出现，此时第 $i$ 个位置上是其它城市。 $N \times N$ 矩阵 $V$ 可表示一次有效的路径，即 $V$ 矩阵可唯一确定对所有城市的访问次序。

定义连续 Hopfield 网络的能量函数为

$$E = \frac{A}{2} \sum_x \sum_i \sum_{j \neq i} v_{xi} v_{xj} + \frac{B}{2} \sum_i \sum_x \sum_{x \neq y} v_{yi} v_{xj} + \frac{C}{2} \left( \sum_x \sum_i v_{xi} - N \right)^2 + \frac{D}{2} \sum_x \sum_{x \neq y} \sum_i d_{xy} v_{xi} (v_{y,i+1} + v_{y,i-1}), \quad (5)$$

式中:  $A, B, C, D$  均为大于 0 的常数;

$d_{xy}$  为城市  $x$  与城市  $y$  之间的距离。

式 (5) 的前 3 项保证矩阵  $\mathbf{v}$  的每 1 行不多于 1 个 1 时所对应的项的值最小, 第四项包含有效路径长度信息, 为问题的目标项。

根据式 (1) 求得网络的动态方程为

$$\begin{cases} C_{xi} \frac{du_{xi}}{dt} = -\frac{u_{xi}}{R_{xi}} - A \sum_{j \neq i} v_{xj} - B \sum_{y \neq x} v_{yi} - \\ C \left( \sum_x \sum_j v_{xj} - N \right) - D \sum_{y \neq x} d_{xy} (v_{y,i+1} + v_{y,i-1}), \quad (6) \\ v_{xi} = f(u_{xi}) \end{cases}$$

### 2.3 连续 Hopfield 网络求解 TSP 的改进算法

应用一般的连续 Hopfield 网络方法求解 TSP, 虽然在速度上有一定的优势, 但不能保证  $dE/dt=0$  的点就是真正的极小点。利用该方法求解时, 参数配置方法较为复杂, 若参数选择不恰当, 可能得不到最优解, 有时甚至出现“非法”或“冻结”现象。因此, 有必要对上述算法进行改进。通过分析发现, 式 (5) 的第三项仅在网络输出全为 0 时才起到约束作用, 否则前 2 项已经保证了第三项成立, 故只需对前 2 项稍做修改, 第三项即可省去。由于行列的对称性, 可取  $B=A$ ,  $D$  项由式 (5) 的  $(v_{y,i+1} + v_{y,i-1})$  简化为  $(v_{y,i+1})$ , 方程式仍满足优化目标约束的要求。因此, 能量函数可简化为

$$E = \frac{A}{2} \sum_{x=1}^N \left( \sum_{i=1}^N v_{xi} - 1 \right)^2 + \frac{A}{2} \sum_{i=1}^N \left( \sum_{x=1}^N v_{xi} - 1 \right)^2 + \frac{D}{2} \sum_{x=1}^N \sum_{y=1}^N \sum_{i=1}^N v_{xi} d_{xy} v_{y,i+1}, \quad (7)$$

相应的动态方程为

$$\begin{aligned} \frac{du_{xi}}{dt} &= -\frac{\partial E}{\partial v_{xi}} = -A \left( \sum_{i=1}^N v_{xi} - 1 \right) - \\ &A \left( \sum_{y=1}^N v_{yi} - 1 \right) - D \sum_{y=1}^N d_{xy} v_{y,i+1} \circ \end{aligned} \quad (8)$$

经过归纳及总结后, 给出改进后的算法<sup>[5]</sup>。

Step1 输入 TSP 问题的城市位置坐标并计算各城市间的距离;

Step2 计算动态方程  $\frac{du_{xi}}{dt}$ , 利用一阶欧拉法计算

$$u_{xi}(t+1) = u_{xi}(t) + \frac{du_{xi}}{dt} \Delta T;$$

Step3 计算连续 Hopfield 网络神经元  $x_i$  的状态值  $v_{si}(t)$ ;

Step4 计算能量函数  $E$ ;

Step5 判断是否结束, 如果结束则输出最优解并退出, 否则迭代累加器加 1 并返回 Step2。

流程图如图 1 所示。

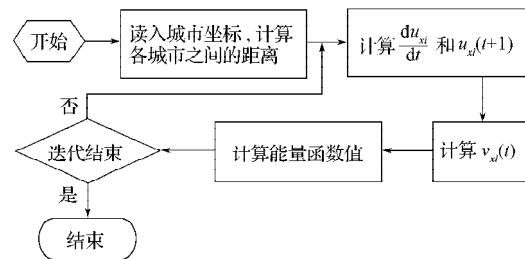


图 1 算法流程图

Fig. 1 Algorithm flow chart

## 3 系统仿真与性能测试

系统仿真与测试的环境为: CPU1.73 GHz, 内存 512 M, 操作系统 WindowsXP, 编程语言 MATLAB R2010a。本实验首先对 16 个结点的 TSP 问题利用改进后的算法进行仿真, 然后针对算法的参数选择、问题规模及城市拓扑等方面对算法进行性能仿真测试。

### 3.1 系统仿真

实验对象是规模为 16 个结点的 TSP 问题, 各城市坐标如表 1 所示。网络初始化条件为

$$u_{xi}(t) = u_0 \ln(N-1) + \delta_{xi} \quad (x, i = 1, 2, \dots, N; t = 0),$$

式中:  $u_0=0.1$ ,  $N=16$ ,  $\delta_{xi}$  为  $(-1, +1)$  的随机值, 参数  $A=200$ ,  $D=100$ , 迭代次数为 10 000。

表 1 城市坐标

Table 1 City coordinates

序号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
横坐标	0.12	0.16	0.23	0.37	0.33	0.34	0.55	0.56	0.47	0.43	0.59	0.67	0.75	0.65	0.66	0.87
纵坐标	0.34	0.76	0.56	0.11	0.38	0.67	0.25	0.45	0.57	0.89	0.69	0.22	0.34	0.55	0.88	0.64

系统运行后求得路径长度为 3.385, 最优能量值为 131.638, 见图 2 和 3。从图 3 可看出, 网络能量随迭代过程不断减少, 当网络能量变化很小时, 网络的神经元状态也趋于平衡点, 此时对应的城市顺序

即为待求的优化路径。实验表明, 利用连续 Hopfield 神经网络可快速准确地解决 TSP 问题。同理, 对于其他利用枚举法会产生“组合爆炸”的组合优化问题, 利用连续 Hopfield 神经网络也可进行优化计算<sup>[9]</sup>。

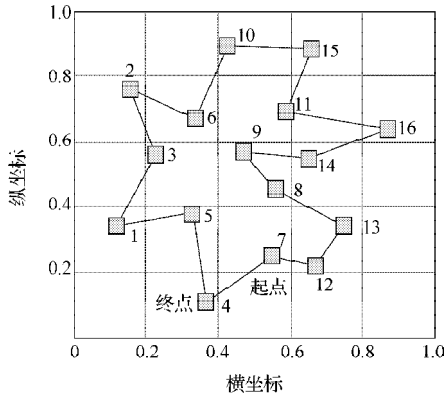


图2 最优路径

Fig. 2 Optimal path

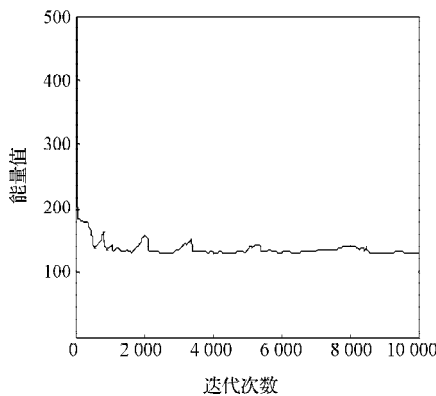


图3 最优能量值

Fig. 3 Optimal energy value

### 3.2 算法性能测试

为了验证本文算法的实际性能, 需要从多个角度对算法进行考察, 包括参数选取方案测试、问题规模测试及拓扑结构测试。

#### 3.2.1 参数测试

实验选取 5 套方案, 如表 2 所示。迭代次数为 12 000, 试验次数为 24×5 次。

表2 参数配置与测试数据

Table 2 Parameter configuration test data

方案 序号	参数选取		平均运行 时间 /s	平均路径 长度	最优路径 长度	有效 收敛率 /%
	A	D				
1	100	50	14.802	3.808	3.593	58.33
2	200	100	15.251	3.860	3.385	58.33
3	300	150	14.751	3.807	3.601	62.50
4	400	200	14.189	3.587	3.464	79.17
5	500	250	14.478	3.760	3.437	79.17

从表 2 可看出, 参数配置方案对算法的运行时间和路径寻优有一定影响, 但影响不大, 从平均路径长度来看, 5 种方案都较接近。但参数配置对有效收敛率的影响较大, 其中, 方案 1 和方案 2 的有效收敛率只有 58.33%, 而方案 4 和方案 5 则提高到 79.17%。总体来看, 该算法是相对较稳定的算法, 这也是改

进算法优于传统神经网络方法的原因之一。

#### 3.2.2 规模测试

实验选取 4 种不同规模的 TSP 问题来验证本算法的健壮性, 测试方法是将问题规模分为 4 种级别: 规模 A, B, C 和 D, 相对应的结点数  $N$  分别为 10, 12, 14 和 16, 迭代次数分别选取 8 000 ~ 14 000 不等, 实验结果如表 3 所示。测试的评价指标是有效收敛率, 实验总次数为 40×4 次。

表3 规模测试的有效收敛率

Table 3 Effective convergence rate of scale test %

迭代次数	TSP 问题规模			
	A	B	C	D
8 000	100	80	90	90
10 000	80	60	60	90
12 000	90	80	80	90
14 000	40	70	40	50

从表 3 可看到, 规模 A 的最佳迭代次数为 8 000, B 的最佳迭代次数是 8 000 或 12 000, C 的最佳迭代次数是 8 000, D 的最佳迭代次数是 14 000 以下。4 种规模问题的迭代次数为 14 000 时, 有效收敛率下降较明显。因此, 利用连续 Hopfield 网络求解组合优化问题, 迭代次数不是越多越好, 太多的迭代次数不仅较难得到优化解, 且会增加系统时空开销。

#### 3.2.3 拓扑测试

实验选用较典型的 3 种拓扑结构: 均匀拓扑、对称拓扑和区域聚集拓扑。为了评价算法在不同拓扑结构下的运算性能, 采用路径质量做为评价指标。路径质量参数  $Q_L = (l_{ave} - l_{ans}) / (l_{ave} - l_{opt})$ , 其中:  $l_{ave}$  为路径的平均长度,  $l_{ans}$  为算法某次运算得到的解,  $l_{opt}$  表示问题的最优解。实验次数为 50×3 次, 实验数据如表 4 所示, 表中  $t_{ave}$  表示算法运行的平均时间。

表4 拓扑结构测试数据

Table 4 Topology test data

拓扑结构	路径长度			$Q_L$	$t_{ave}/s$	有效 收敛率 /%
	$l_{ave}$	$l_{opt}$	$l_{ans}$			
均匀	3.602	3.531 4	3.365 7	3.347	3.721	50
对称	3.876	3.531 4	4.061 4	-0.538	3.641	60
区域聚集	2.978	2.890 3	3.394 3	-4.746	3.627	70

从实验结果来看, 本算法性能较好, 3 种拓扑结构求解的平均运行时间均接近 3.6 s, 都能获得较满意的结果。通过对表 4 中实验数据的分析发现, 城市拓扑结构对神经网络性能影响较大, 这一点从有效收敛率和路径质量值可以看出来。可见, 结点拓扑分布越分散, 有效收敛率越小, 如均匀拓扑只有 50%; 而结点拓扑分布越集中, 有效收敛率越高, 如区域聚集拓扑达 70%。从路径质量方面来看, 网络对

均匀拓扑结构所求得的路径质量最好。3种不同拓扑结构的路径寻优结果如图4~6所示。

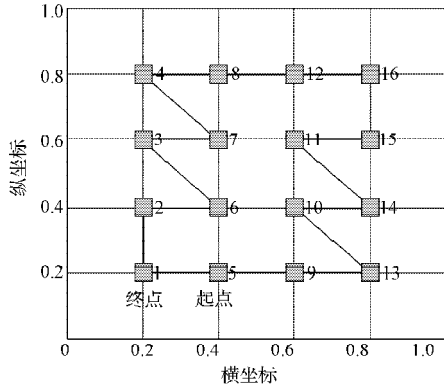


图4 均匀拓扑的最优解

Fig. 4 Optimal solution of uniform topology

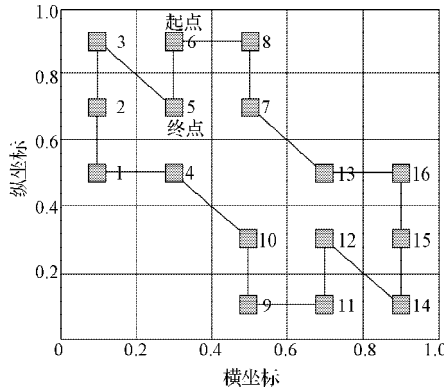


图5 对称拓扑的最优解

Fig. 5 Optimal solution of symmetry topology

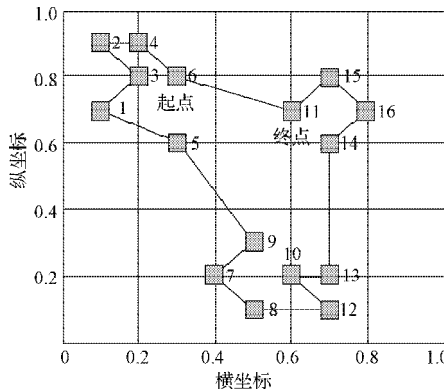


图6 区域聚集拓扑的最优解

Fig. 6 Optimal solution of regional aggregation topology

### 4 结语

组合优化在社会生产中的各个领域有着重要的应用，研究快速、准确、高效的组合优化问题求解算法意义重大。本文结合 Hopfield 神经网络的特点，利用改进的连续 Hopfield 网络求解了典型的组合优化问题（即 TSP 问题）。仿真实验和性能测试表明，改进算法在参数配置、稳定性及收敛速度等方面优

于传统方法，为组合优化问题求解方法的探索提供一定的借鉴作用。

### 参考文献:

[1] 张德丰. MATLAB 神经网络仿真与应用[M]. 北京: 电子工业出版社, 2009: 229-233.  
Zhang Defeng. MATLAB Neural Network Simulation and Application[M]. Beijing: Publishing House of Electronics Industry, 2009: 229-233.

[2] 兰兆青. Hopfield 神经网络在 TSP 问题中的应用[D]. 山西: 中北大学, 2008.  
Lan Zhaoqing. The Application of Hopfield Neural Network in the TSP Problem[D]. Shanxi: North University of China, 2008.

[3] 费春国, 韩正之, 唐厚君. 基于连续 Hopfield 网络求解 TSP 的新方法[J]. 控制理论与应用, 2006, 23(6): 907-912.  
Fei Chunguo, Han Zhengzhi, Tang Houjun. Novel Method to Solve TSP by Continuous Hopfield Neural Network[J]. Control Theory & Applications, 2006, 23(6): 907-912.

[4] 高 隽. 神经网络原理及仿真实例[M]. 2 版. 北京: 机械工业出版社, 2007: 99-101.  
Gao Jun. Principle of Artificial Neural Networks and Simulation Instances[M]. 2nd ed. Beijing: Machinery Industry Press, 2007: 99-101.

[5] 史 峰, 王小川, 郁 磊, 等. MATLAB 神经网络 30 个案例分析[M]. 北京: 北京航空航天大学出版社, 2010: 100-104.  
Shi Feng, Wang Xiaochuan, Yu Lei, et al. MATLAB Neural Network Analysis of 30 Cases[M]. Beijing: Beijing Aeronautics and Astronautics Press, 2010: 100-104.

[6] 张宗飞. 求解组合优化问题的改进型量子进化算法[J]. 计算机工程与设计, 2010, 31(17): 3891-3894.  
Zhang Zongfei. Novel Improved Quantum Evolutionary Algorithm for Combinatorial Optimization Problems[J]. Computer Engineering and Design, 2010, 31(17): 3891-3894.

[7] 孙守宇, 郑君里. Hopfield 网络求解 TSP 的一种改进算法和理论证明[J]. 电子学报, 1995, 23(1): 73-78.  
Sun Shouyu, Zheng Junli. A Modified Algorithm and Theoretical Analysis for Hopfield Network Solving TSP[J]. Acta Electronica Sinica, 1995, 23(1): 73-78.

[8] Jolai F, Ghanbari A. Integrating Data Transformation Techniques with Hopfield Neural Networks for Solving Travelling Salesman Problem[J]. Expert Systems with Applications, 2010, 37(7): 5331-5335.

[9] Marinakis Yannis, Marinaki Magdalene, Dounias Georgios. Honey Bees Mating Optimization Algorithm for the Euclidean Traveling Salesman Problem[J]. Information Sciences, 2010, 32: 1-15.

(责任编辑: 李玉珍)