

第八章 基于反馈网络的优化

内容

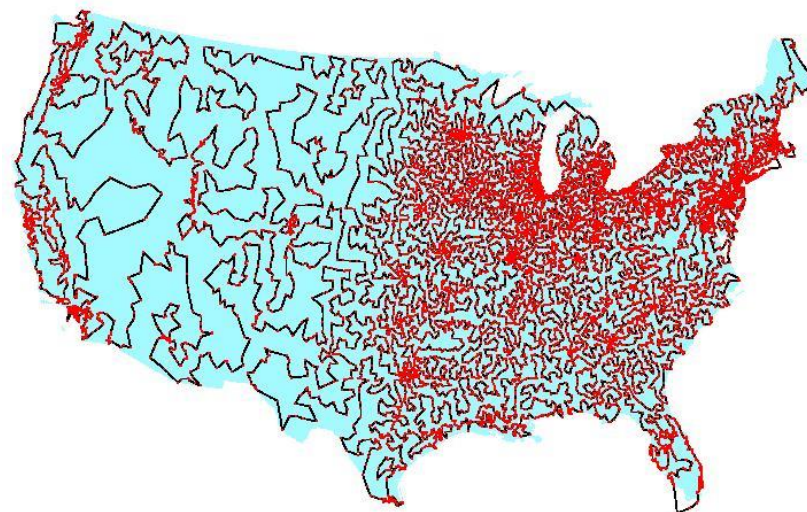
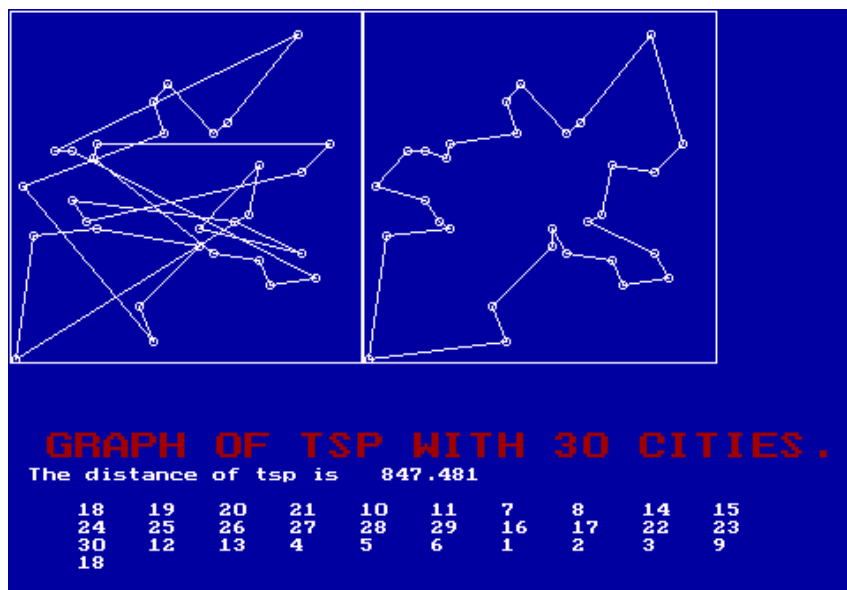
- 神经动力学
- 连续Hopfield网络
- 神经优化

概述

- 前向神经网络：
 - 从信息的观点来看，它是一个强有力的学习系统，系统结构简单、学习算法易于实现；
 - 从系统的观点来看，它是一个静态非线性映射，通过简单非线性处理元件的复合映射获得复杂系统的非线性处理能力；
 - 从计算的观点来看，它并不是一个动力学系统，缺乏丰富的动力学行为。
- 1982年，美国物理学家Hopfield提出了单层全互连的反馈神经网络(对称权值)，它是一个反馈动力学系统，具有很强的计算能力。
- Hopfield用能量函数的思想形成一种新的计算方法，阐明了神经网络与动力学的关系，并用非线性动力学的方法来研究神经网络的特性，建立了网络稳定性判据，并指出信息存储在网络中神经元的连接上，形成了离散Hopfield网络。

概述

- Hopfield还将反馈网络与统计物理中的Ising模型相类比，把磁旋的向上和向下方向视为神经元的激活和抑制状态，把磁旋的相互作用视为神经元的突触连接，这种类比的大量物理学理论以及大量物理学家进入NN领域铺平了道路。
- 1984年，Hopfield设计和研制了Hopfield网络模型的电路，指出神经网络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，称之为连续Hopfield网络。
- Hopfield基于连续反馈网络成功解决了旅行商TSP问题(NP-hard)，开创了神经优化计算的里程碑。



$$30! > 2.65 \times 10^{32}$$

从30!个解中寻找最优解，若计算机1秒列举1亿个解，则穷举需要约 8.4×10^{16} 年！若穷举20! 个解也需要771年多时间！（从元朝就得开始）

城市数	24	25	26	27	28	29	30	31
计算 时间	1 sec	24 sec	10 min	4.3 hour	4.9 day	136.5 day	10.8 year	325 year

神经动力学

- 研究神经网络非线性特性的学科称为神经动力学(Neurodynamics), 它可分为确定性神经动力学和统计性神经动力学。
 - 确定性神经动力学将神经网络作为确定性行为, 在数学上用非线性微分方程的集合来描述系统行为, 方程的解为确定解。
 - 统计性神经动力学将神经网络视为被噪声所干扰, 在数学上采用随机性的非线性微分方程来描述系统的行为, 方程的解用概率来表示。
- 动力学系统是状态随时间变化的系统。令 $V(t)=[v_1(t), v_2(t), \dots, v_N(t)]^T$ 为非线性动力学系统的状态变量, 其中 t 为独立的连续时间变量, N 为系统状态维数, 则系统的动力特性可用如下微分方程表示:

$$\frac{dv_i(t)}{dt} = F_i(v_i(t)), i = 1, 2, \dots, N$$
$$\text{or } \frac{dV(t)}{dt} = F(V(t))$$

其中, $F()$ 为包含自变量的非线性函数。

神经动力学

- 定义1：若矢量 V^* 满足 $F(V^*)=0$ ，则称之为系统的稳态或平衡态。
- 定义2：若对任意 $\varepsilon>0$ ，存在 $\delta>0$ ，当 $\|V(0)-V^*\|<\delta$ 时，对所有 $t>0$ ，均有 $\|V(t)-V^*\|<\varepsilon$ ，则称 V^* 是一致稳定的。
- 定义3：若存在 $\delta>0$ 满足 $\|V(0)-V^*\|<\delta$ ， $V(\infty)\rightarrow V^*$ ，则称 V^* 是收敛的。
- 定义4：若平衡态是稳定且收敛的，则称其为渐进稳定。
- 定义5：若平衡态是稳定的，且当时间趋于无穷大时，所有系统轨道均收敛到该平衡态，则称其为全局渐进稳定。

- 定理1：若平衡态 V^* 的小邻域内存在有界正函数 $E(V)$ ，该函数对时间的导数在区域中是有界非正函数，则 V^* 是稳定的。
- 定理2：若平衡态 V^* 的小邻域内存在有界正函数 $E(V)$ ，该函数对时间的导数在区域中是有界负函数，则 V^* 是渐进稳定的。

- 满足上述条件的标量函数 $E(V)$ 称为平衡态 V^* 的Lyapunov函数。

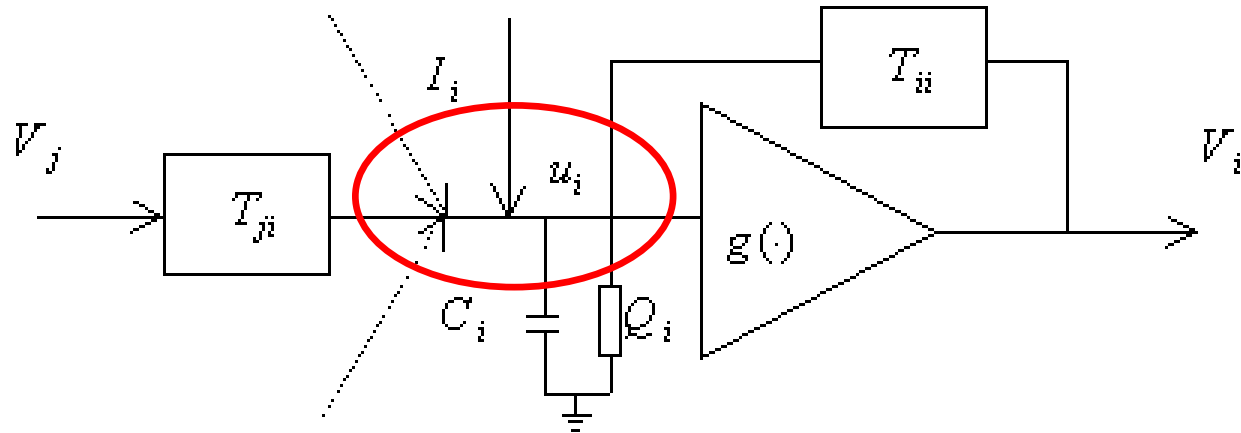
神经优化

- 连续HNN的稳定性分析
- 基于HNN模型优化的一般流程
- 基于HNN模型优化的缺陷
- 基于HNN模型的TSP优化
- 基于HNN模型优化的不稳健性
- 改进方案
- 混沌神经网络

连续Hopfield网络

- 1984年，Hopfield设计和研制了Hopfield网络模型的电路，网络中的激励函数为连续函数，网络的输入、输出均为模拟量，各神经元采用同步并行工作方式。
- 连续Hopfield网络可用于解决大量组合优化问题，譬如对旅行商TSP问题(NP-hard)的成功解决，开创了神经优化计算的里程碑。

连续Hopfield网络模型



$$\sum_{j=1}^n T_{ji}(V_j - u_i) + I_i = C_i \frac{du_i}{dt} + \frac{u_i}{Q_i}$$

$$\begin{cases} C_i \frac{du_i}{dt} = \sum_{j=1}^n T_{ji} v_j - \frac{u_i}{R_i} + I_i & 1/R_i = 1/Q_i + \sum_{j=1}^n T_{ji} \\ v_i = g(u_i) \end{cases}$$

$g(\cdot)$ 为具有连续且单调增性质的神经元激励函数

激励函数

$$v_i = g(u_i) = \tanh(a_i u_i / 2) = \frac{1 - \exp(-a_i u_i)}{1 + \exp(-a_i u_i)} \quad \text{即双曲正切函数}$$

特点：连续、有界、单调增

$$a_i / 2 = \left. \frac{dg(u_i)}{du_i} \right|_{u_i=0} \quad \text{为激励函数在原点的斜率}$$

$$u_i = g^{-1}(v_i) = -\frac{1}{a_i} \ln \left[\frac{1 - v_i}{1 + v_i} \right] \quad \text{连续、单调增}$$

连续Hopfield网络的稳定性

$$E = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n T_{ij} v_i v_j - \sum_{i=1}^n I_i v_i + \sum_{i=1}^n \int_0^{v_i} g^{-1}(v) dv / R_i$$

$$\frac{dE}{dt} = \sum_{i=1}^n \frac{\partial E}{\partial v_i} \cdot \frac{dv_i}{dt}$$

$$\begin{aligned} \frac{\partial E}{\partial v_i} &= -\frac{1}{2} \sum_{j=1}^n T_{ij} v_j - \frac{1}{2} \sum_{j=1}^n T_{ji} v_j + \frac{u_i}{R_i} - I_i = -\frac{1}{2} \sum_{j=1}^n (T_{ij} - T_{ji}) v_j - \left(\sum_{j=1}^n T_{ji} v_j - \frac{u_i}{R_i} + I_i \right) \\ &= -\frac{1}{2} \sum_{j=1}^n (T_{ij} - T_{ji}) v_j - C_i \frac{du_i}{dt} = -\frac{1}{2} \sum_{j=1}^n (T_{ij} - T_{ji}) v_j - C_i g^{-1'}(v_i) \frac{dv_i}{dt} \end{aligned}$$

当 $T_{ij} = T_{ji}$ 时, $\frac{dE}{dt} = -\sum_{i=1}^n C_i g^{-1'}(v_i) \cdot \left(\frac{dv_i}{dt}\right)^2 \leq 0$, 且当 $\frac{dv_i}{dt} = 0$ 时 $\frac{dE}{dt} = 0$ 。

$$\frac{\partial E}{\partial v_i} = -C_i \frac{du_i}{dt}$$

基于HNN模型优化的一般流程

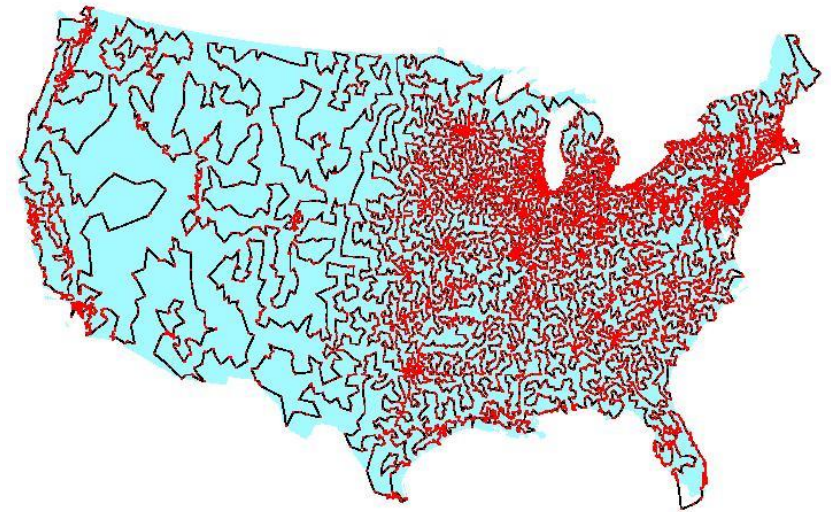
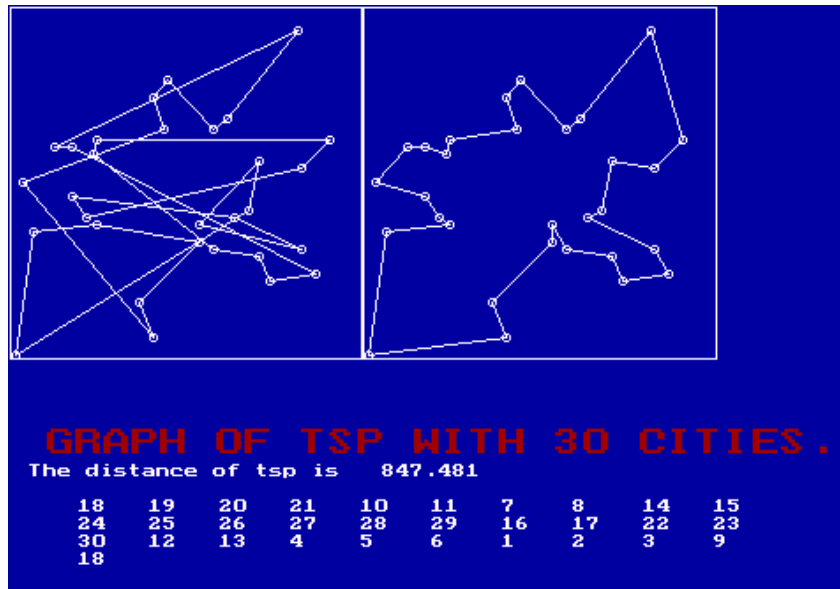
- Hopfield网络是一种非线性动力学模型，通过引入类似Lyapunov函数的能量函数概念，把神经网络的拓扑结构(用连接权矩阵表示)与所求问题(用目标函数描述)对应起来，转换成神经网络动力学系统的演化问题。因此，在用HNN求解优化问题之前，必须将问题映射为相应的神经网络。譬如对TSP问题的求解，首先将问题的合法解映射为一个置换矩阵，并给出相应的能量函数，然后将满足置换矩阵要求的能量函数的最小值与问题的最优解相对应。
- 对于一般性问题，通常需要以下几方面工作：
 - 选择合适的问题表示方法，使神经网的输出与问题的解相对应；
 - 构造合适的能量函数，使其最小值对应问题的最优解；
 - 由能量函数和稳定条件设计网络参数，如连接权值和偏置参数等；
 - 构造相应的神经网络和动态方程；
 - 用硬件实现或软件模拟。

基于HNN模型优化的缺陷

- 基于HNN模型设计优化算法的出发点在于以下两方面：
 - 神经网络是稳定的，网络势必收敛到渐近平衡点；
 - 神经网络的渐近平衡点恰好是能量函数的极小点。
- 在Lyapunov稳定理论中，平衡点与函数极点预先已对应，必然把运动轨迹趋于平衡点与Lyapunov函数趋于极小值相对应，从而保证稳定理论的正确性。稳定性分析仍然是动态反馈网络应用的理论依据。
- 从稳定性理论分析，HNN模型优化方法并不严格，虽借助Lyapunov法的思想构造能量函数，但并不知道网络平衡点的位置，因此不能保证能量函数的极值点就是问题真正的全局极小点，从而往往出现网络不能收敛到全局极小的情况。

- 利用HNN优化求解，若网络参数选择不好，往往得不到最优解，或次优度很差，甚至出现“非法”或“冻结”现象。
 - “非法”指结果不符置换矩阵要求
 - “冻结”意味着算法不收敛
- 因此，研究参数对算法性能的影响很有意义，尽管分析权矩阵特征值和特征向量可给出一些参数选择的约束，但目前尚无一般而实用的选择方法。
- 鉴于理论上的不严格性，同时由于传统HNN网络仍采用梯度下降策略，因此基于HNN网络的优化计算通常会导致以下问题：
 - 网络最终收敛到局部极小解，而非问题的全局最优解；
 - 网络可能会收敛到问题的不可行解；
 - 网络优化的最终结果很大程度上依赖于网络的参数，即参数鲁棒性较差。

TSP优化



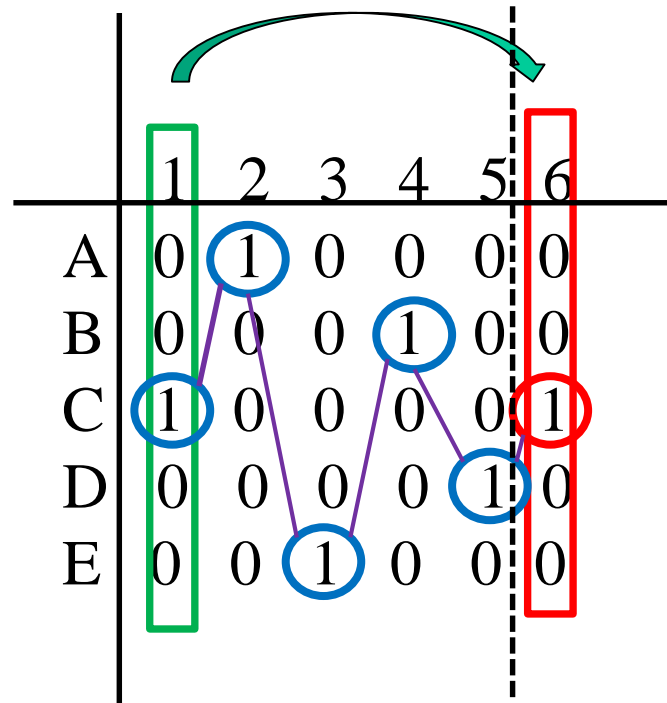
$$30! > 2.65 \times 10^{32}$$

从30!个解中寻找最优解，若计算机1秒列举1亿个解，则穷举需要约 8.4×10^{16} 年！若穷举20! 个解也需要771年多时间！（从元朝就得开始）

城市数	24	25	26	27	28	29	30	31
计算 时间	1 sec	24 sec	10 min	4.3 hour	4.9 day	136.5 day	10.8 year	325 year

基于HNN模型的TSP优化

0-1矩阵编码：



旅行线路：C→A→E→B→D→C

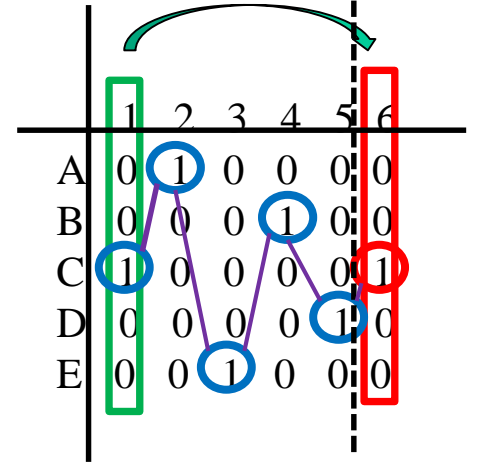
基于HNN模型的TSP优化

$$E_0 = \frac{A}{2} \sum_x \sum_i \sum_{j \neq i} v_{xi} v_{xj} + \frac{B}{2} \sum_i \sum_x \sum_{y \neq x} v_{xi} v_{yi} + \frac{C}{2} (\sum_x \sum_i v_{xi} - n)^2$$

$$+ \frac{D}{2} \sum_x \sum_i \sum_{y \neq x} d_{xy} v_{xi} (v_{y,i+1} + v_{y,i-1})$$

$$E = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n T_{ij} v_i v_j - \sum_{i=1}^n I_i v_i + \sum_{i=1}^n \int_0^{v_i} g^{-1}(v) dv / R_i$$

$$= E_0 + \sum_{i=1}^n \int_0^{v_i} g^{-1}(v) dv$$



$$\begin{cases} T_{xi,yj} = -A\delta_{xy}(1-\delta_{ij}) - B\delta_{ij}(1-\delta_{xy}) - C - Dd_{xy}(\delta_{j,i+1} + \delta_{j,i-1}) \\ I_{xi} = C \cdot n \end{cases}$$

$$\frac{\partial E}{\partial v_i} = -C_i \frac{du_i}{dt} \begin{cases} \frac{du_{xi}}{dt} = -\frac{u_{xi}}{\tau} - A \sum_{j \neq i} v_{xi} - B \sum_{y \neq x} v_{yi} - C(\sum_x \sum_i v_{xi} - n) - D \sum_{y \neq x} d_{xy} (v_{y,i+1} + v_{y,i-1}) \\ v_{xi} = g(u_{xi}) = \frac{1}{2} \cdot [1 + \tanh(u_{xi} / u_0)] \end{cases}$$

优化流程

- (1) 确定网络参数 A 、 B 、 C 、 D 、 λ 和初值 u_0 、 u_{xi} 、 δu_{xi} 。
- (2) 由 Sigmoid 激励函数计算每个神经元的输出 v_{xi} 。
- (3) 计算能量函数 E 和 du_{xi} / dt 的值。
- (4) 以 $u_{xi}^{k+1} = u_{xi}^k + \lambda \cdot du_{xi} / dt$ 确定新的 u_{xi} 和 v_{xi} 。
- (5) 判断能量函数是否满足稳定条件。
若到一定条件仍不稳定则认为“冻结”而退出算法，否则转(2)；
若满足稳定条件则转(6)。
- (6) 输出最优路径、路径长度以及执行时间。

假设 $n = 10$ ，各点横、纵坐标均取(0,200)内随机数，

选取 $A = B = D = 500$ ， $C = A / \sqrt{n}$ ， $u_0 = 0.02$ ， $\lambda = 10^{-5}$

表 u_0 对算法性能的影响

参数 u_0	优化结果
0.03	“冻结”
0.02	d=521.00, t=11.81s
0.01	d=521.00, t=9.01s
0.005	d=563.63, t=7.42s
0.002	“非法”

u_0 的影响： u_0 下降导致寻优时间缩短，但 u_0 太小会导致路径不优或“非法”路径的出现； u_0 太大又可能出现“冻结”现象。原因在于 u_0 相当于放大器增益， u_0 太小使 Sigmoid 激励函数趋于离散阶跃函数，从而不易获得优解；反之使激励函数过于平坦，神经元状态不易收敛到“0”或“1”。

$$\begin{cases} \frac{du_{xi}}{dt} = -\frac{u_{xi}}{\tau} - A \sum_{j \neq i} v_{xi} - B \sum_{y \neq x} v_{yi} - C \left(\sum_x \sum_i v_{xi} - n \right) - D \sum_{y \neq x} d_{xy} (v_{y,i+1} + v_{y,i-1}) \\ v_{xi} = g(u_{xi}) = \frac{1}{2} \cdot [1 + \tanh(u_{xi} / u_0)] \end{cases}$$

表 λ 对算法性能的影响

参数 λ	优化结果
1.5×10^{-5}	“非法”
1.0×10^{-5}	d=521.00, t=11.81s
0.5×10^{-5}	d=521.00, t=16.20s
0.1×10^{-5}	“冻结”

λ 的影响： λ 减小导致寻优时间增加，太小的 λ 会引起“冻结”现象或路径不优； λ 太大则会引起“非法”路径。从 u_{xi} 的迭代式知， λ 的大小决定 u_{xi} 变化大小的快慢， λ 大使 u_{xi} 变化剧烈而易出现“非法”现象， λ 小使 u_{xi} 更新过于慢而易出现“冻结”现象。

$$u_{xi}^{k+1} = u_{xi}^k + \lambda \cdot du_{xi} / dt$$

表 C 对算法性能的影响

参数 C	优化结果
0.6 A	“非法”
0.4 A	D=521.00, t=5.05s
$A / \sqrt{10}$	D=521.00, t=11.8s
0.2 A	“冻结”

参数 C 的影响： C 下降导致寻优时间增加，过小的 C 值会引起“冻结”现象； C 太大而导致“非法”路径的出现。原因在于 C 的大小影响能量函数中置换矩阵约束项的权重， C 太大使能量函数变化剧烈，尽管计算过程缩短，但路径可能出现“非法”；相反则导致收敛太慢而出现“冻结”。

$$\begin{aligned}
 E_0 = & \frac{A}{2} \sum_x \sum_i \sum_{j \neq i} v_{xi} v_{xj} + \frac{B}{2} \sum_i \sum_x \sum_{y \neq x} v_{xi} v_{yi} + \frac{C}{2} (\sum_x \sum_i v_{xi} - n)^2 \\
 & + \frac{D}{2} \sum_x \sum_i \sum_{y \neq x} d_{xy} v_{xi} (v_{y,i+1} + v_{y,i-1})
 \end{aligned}$$

表 D 对算法性能的影响

参数 D	优化结果
1.5 A	“非法”
1.2 A	d=521.00, t=16.2 s
0.8 A	D=521.00, t=3.46s
0.1 A	D=584.73, t=2.31s

参数 D 的影响： D 下降使寻优时间缩短， D 太小则引起路径不优； D 过大则会引起“非法”路径。由能量函数知， D 的改变即能量函数中目标项权重改变， D 太大使目标项权重过大而导致与约束项不一致，从而出现“非法”； D 太小使目标项权重降低，势必出现路径不优的现象。

$$\begin{aligned}
 E_0 = & \frac{A}{2} \sum_x \sum_i \sum_{j \neq i} v_{xi} v_{xj} + \frac{B}{2} \sum_i \sum_x \sum_{y \neq x} v_{xi} v_{yi} + \frac{C}{2} \left(\sum_x \sum_i v_{xi} - n \right)^2 \\
 & + \frac{D}{2} \sum_x \sum_i \sum_{y \neq x} d_{xy} v_{xi} (v_{y,i+1} + v_{y,i-1})
 \end{aligned}$$

基于HNN模型优化的不稳健性

- 初值的不稳健性，各神经元初值大小和分布情况影响寻优结果；
- 模型参数的不稳健性，参数直接影响能量函数中约束项和目标项在优化过程中的地位和重要程度；
- 问题结构的不稳健性，相同模型参数对不同结构的问题会导致不同的收敛性；
- 激励函数的不稳健性，主要是函数形态的影响；
- 算法收敛性标志的不严格性。

改进方案

- 在迭代过程中设置神经元输出变“0”和“1”的阈值，一旦达到阈值相应的输出就设置为“0”或“1”，从而提高算法的收敛速度；
- 当每行或每列出现一个“1”时，相应行或列的其余元置“0”；
- 取消神经元动态方程中的自反馈项，以减小能量的消耗；
- 用离散化或其它形态的激励函数代替连续型Sigmoid函数，改善收敛速度和性能；
- 改变能量函数形式或置换矩阵的定义等。
- 引入全局优化技术，譬如SA、GA、混沌等。

改进研究

- Wilson等针对随机扰动的缺点，归一化城市坐标到区间，引入一个偏置到初值来改善优化性能；

$$bias(i, j) = \cos[\arctan(\frac{y_i - 0.5}{x_i - 0.5}) + \frac{2\pi(j-1)}{n} \sqrt{(x_i - 0.5)^2 + (y_i - 0.5)^2}]$$

- Brandt等针对问题的表征方式提出三下标法，用 v_{ijk} 表示从第 i 个城市到第 j 个城市恰是第 k 次访问，这样总路径长度可表示为 $\sum_i \sum_j \sum_k d_{ij} v_{ijk}$ ，数据项 d_{ij} 仅是能量函数中一次项的系数，与权矩阵无关，但需要 n^3 个神经元数，计算量较大；
- Szu等在能量函数方面提出了一个强化约束公式，它表示置换矩阵每行每列只有一个“1”，它能及时唤醒全零元行(列)中的神经元，抑制过多的“1”出现，减少非法路径；

$$E = \sum_x (\sum_i v_{xi} - 1)^2 + \sum_i (\sum_x v_{xi} - 1)^2$$

改进研究

- Xu等改变置换矩阵的定义，用 v_{xy} 表示 x 与 y 之间的连接， $v_{xy}=1$ 表示 x 与 y 相连，否则不连，这样可用 $n(n-1)/2$ 个神经元建立模型，能量函数定义如下，神经元间的连接复杂性较HNN有所降低，并消除了一些冗余项，提高了计算效率；

$$E = \frac{A}{2} \sum_x (\sum_{y \neq x} v_{xy} - 2)^2 + \frac{B}{2} \sum_y (\sum_{x \neq y} v_{xy} - 2)^2 + C \sum_{x=1}^n \sum_{y=x+1}^n v_{xy} d_{xy}$$

- Wang等用时变能量函数(其中 $S=1+K\exp(-at)$ ， K 、 a 为常数)，通过自适应改变能量函数中某个参数改善算法性能， S 的缓慢变化可避免陷入局部极小，且随 t 加大而最终趋于1，从而得到有效优解，并较大幅度提高得到优解的比例。

$$E = \frac{A}{2} \sum_x (\sum_i v_{xi} - S)^2 + \frac{B}{2} \sum_i (\sum_x v_{xi} - S)^2 + \frac{C}{2} \sum_x \sum_y \sum_i d_{xy} v_{xi} (v_{y,i+1} + v_{y,i-1})$$

$$+ \frac{A+B}{2} \sum_x \sum_i v_{xi} (1 - v_{xi})$$

混沌神经网络

Chaotic neural networks

- CNN提出原因
- CNN模型
 - 内模型
 - 外模型
- ACNN算法
- TSP优化

概述

- 混沌是一种普遍的非线性现象，其行为复杂且类似随机，但存在精致的内在规律性。混沌的发现，对科学的发展具有空前而深远的影响。近年来，混沌控制、混沌同步和混沌神经网络受到了广泛关注，并展现出诱人的应用与发展前景。
- 混沌具有其独特性质：
 - 随机性，即混沌具有类似随机变量的杂乱表现；
 - 遍历性，即混沌能够不重复地历经一定范围内的所有状态；
 - 规律性，即混沌是由确定性的迭代式产生的。
- 介于确定性和随机性之间，混沌具有丰富的时空动态，系统动态的演变可导致吸引子的转移。最重要的是，混沌的遍历性特点可作为搜索过程中避免陷入局部极小的一种优化机制，这与模拟退火的概率性劣向转移和禁忌搜索的禁忌表检验存在明显的区别。
- 混沌已成为一种新颖的优化工具，并受到广泛重视和大量研究。

CNN提出原因

- 鉴于在高强度连接下的神经网络依靠集体协同作用能自发产生计算行为，Hopfield等提出了反馈型的HNN神经网络，并用于求解组合优化问题。但是，基于HNN的优化计算通常会导致以下问题，从而HNN的应用受到了很大的限制，尤其是复杂或大规模的问题。
 - 单纯梯度下降策略易使网络最终收敛到局部极小解，而非问题的全局最优；
 - 网络可能会收敛到问题的不可行解；
 - 与能量函数相关的参数难以确定，参数鲁棒性和初值鲁棒性较差。
- 受生物神经元混沌特性的启发，通过在HNN中引入混沌动态，Aihara等提出了混沌神经网络模型。基于Euler离散化的HNN，通过增加一个大的自反馈项，Nozawa得到了类似的CNN模型。借鉴混沌动态的遍历性特点，搜索过程不受能量障碍的限制，从而可有效避免优化过程陷入局部极小解。对TSP的优化研究，验证了CNN相对随机算法的有效性。从而，混沌神经网络成为了改进HNN的优化效率和质量，乃至解决组合优化问题的有效工具，并在近几年受到优化领域的广泛关注与研究。

HNN模型回顾

$$dx_i / dt = -\partial E / \partial y_i = -x_i / \tau + \sum_j w_{ij} y_j + I_i$$

$$x_i(k+1) = (1 - \Delta t / \tau) x_i(k) + \Delta t \cdot (\sum_j w_{ij} y_j + I_i)$$

CNN内模型

- Chen等：先利用瞬时混沌动态进行全局遍历性搜索，而后再转入梯度下降搜索。

$$\begin{cases} x_i(k+1) = \gamma x_i(k) + \alpha \cdot \left(\sum_{j \neq i} w_{ij} y_j(k) + I_i \right) - z(k) \cdot (y_i(k) - I_0) \\ z(k+1) = (1 - \beta) z(k), \quad \beta \in [0, 1] \end{cases}$$

CNN内模型

- Wang等：时间步长作为分岔参数用以控制网络的动态，通过对时间步长的衰减控制，网络动态同样能够经历一个由混沌到逆分岔再到稳定搜索的过程，而前一阶段的搜索恰好用以避免优化过程陷入局部极小。

$$\begin{cases} x_i(k+1) = (1 - \Delta t(k) / \tau) x_i(k) + \Delta t(k) \cdot (\sum_j w_{ij} y_j(k) + I_i) \\ \Delta t(k+1) = \beta \cdot \Delta t(k), \quad \beta \in (0,1] \end{cases}$$

CNN外模型

- 通过在HNN中引入外部机制产生的混沌噪声来构成混沌神经网络，是其用于组合优化问题的另一种途径。这种方法基于混沌时间序列永不可精确重复的规律，当外部引入的混沌的幅度足够小时，能量函数曲面上的寻优过程在具有丰富时空特性的混沌动态的激励下能够产生避免陷入局部极小的能力，并保持对问题的原始描述不变。

$$x_i(k+1) = (1 - \Delta t / \tau) x_i(k) + \Delta t \cdot \left(\sum_j w_{ij} y_j + I_i - \eta_i(k) \right)$$

CNN的其它研究

- 在“内模型”方面：
 - Zhou等提出的混沌退火算法也采用了类似的优化机制，但其混沌动态是通过引入非线性项而产生的。
 - 在理论研究方面，Chen等对混沌神经网络和离散回归网络的吸引集及其特性进行了分析，并得到了网络具有全局搜索能力的充分条件。
 - Tokuda等建议以转换期引起的间歇动态现象作为混沌在局部极小间切换的潜在机制。

CNN的其它研究

- 在“外模型”方面：
 - Zhou等利用Henon映射产生的混沌时间序列作为外部噪声，算法解决100城市TSP的性能优于Boltzmann机。
 - Hayakawa等利用不同分岔参数下的Logistic映射产生外部噪声，研究了噪声对优化的影响，并得到了一些能够产生良好优化能力的分岔参数值。
 - 通过在HNN中引入不同类型的混沌噪声，Asai等进一步研究了混沌噪声出现时网络对局部极小的搜索动态，并观察和比较了混沌噪声与随机噪声下算法对最优解的跟踪能力。
 - Hasegawa等认为，不同方式产生的混沌噪声只要具有相同的自相关性就能产生几乎相同的优化能力。

CNN的其它研究

- 王凌等将“内模型”和“外模型”集成为一个模型，并对控制参数提出了另一种控制策略。
- Kwok等对内、外方法的CNN结构及其对优化过程的影响建立了一种统一的研究方法，并提出了CNN的统一框架，并对能量函数进行了统一描述。

讨论

- 由于基于混沌的粗搜索的进行，CNN算法最终梯度下降的范围缩小到最优解或满意解的局部邻域，因此在优化性能方面CNN相对HNN具有较大的优越性，这也通过大量数值研究得到了验证。
- 但是，基于CNN的优化也存在一定的缺陷：
 - 首先，基于CNN的优化过程和应用仍受到HNN的限制，CNN仍需将问题的解映射成适合网络描述的结构，如置换码映射到矩阵码；
 - 其次，CNN包容了HNN的所有参数，并增加了混沌项及其控制过程，算法复杂性相对增加。

ACNN模型

- HNN模型：

$$du_{xi}(t)/dt = -u_{xi}(t)/\tau + \beta \cdot [\sum_y \sum_j w_{xi,yj} v_{yj}(t) + I_{xi}]$$

$$\begin{aligned} u_{xi}(k+1) &= (1 - \Delta t / \tau) \cdot u_{xi}(k) + \Delta t \cdot \beta \cdot [\sum_y \sum_j w_{xi,yj} v_{yj}(k) + I_{xi}] \\ &= \alpha \cdot u_{xi}(k) + \beta \cdot [\sum_y \sum_j w_{xi,yj} v_{yj}(k) + I_{xi}] \end{aligned}$$

- ACNN模型：

$$\begin{cases} u_{xi}(k) = \alpha \cdot u_{xi}(k-1) + \beta \cdot [\sum_y \sum_j w_{xi,yj} v_{yj}(k-1) + I_{xi} - s_1] \\ \quad - z(k-1) \cdot [v_{xi}(k-1) - s_2] \\ v_{xi}(k) = 1 / [1 + \exp(-\mu \cdot u_{xi}(k))], \quad x, i, y, j = 1, 2, \dots, n \\ z(k) = z(k-1) / \ln[e + \lambda \cdot (1 - z(k-1))], \quad k = 1, 2, \dots, m \end{cases}$$

ACNN模型的优化机制

- ACNN模型综合了随机性和确定性算法的优点，优化过程分为基于混沌的“粗搜索”和基于HNN网络的“细搜索”两个阶段。
- 粗搜索中的混沌搜索同时具有随机性和轨道遍历性，具有克服局部极小的能力。随机性保证大范围搜索能力，轨道遍历性使算法能按系统自身的行为不重复地遍历所有可能状态，有利于克服一般随机算法中以分布遍历性为机制搜索带来的局限性。由此，算法在机制上保证了同时具有克服局部极小的能力和良好的搜索效率。

粗搜索阶段

- 选定控制参数使系统一开始就进入大范围意义下的较强混沌动态，利用混沌搜索的随机性和轨道遍历性在大范围内按其自身规律进行遍历性搜索。搜索过程按混沌轨道进行遍历，不受目标函数限制，从而有很强避免局部极小的能力。
- 伴随混沌搜索的进行，对控制参数“退温”，退温到一定程度并由系统的自组织能力，混沌动态将退出而进入逆分岔阶段。此后，搜索过程将限制在一些周期解，且能随控制参数的进一步减小而缩小搜索区域。
- 当逆分岔现象随控制参数的进一步退温而达到周期1形态时，算法得到一个易于得到全局意义下最优解的状态，粗搜索过程结束，转入HNN模型的梯度搜索。

细搜索阶段

- 随着粗搜索结束，系统方程中控制参数决定项的作用减弱，优化过程基本按能量函数的梯度下降方向进行。
- 寻优过程将由粗搜索得到的状态在小范围内按HNN模型的梯度下降机制进一步搜索，从而较快收敛到全局意义下较满意的解。

基于ACNN的TSP优化

- Hopfield等原始HNN模型，20次随机实验得到16次合法解，10次满意解；
- Wilson等重复Hopfield的实验，100次随机实验只有15次在1000步迭代内收敛到合法解，且仍为次优解；
- 庄镇泉等的改进HNN模型，100次随机实验只得到47次合法解，12次满意解。

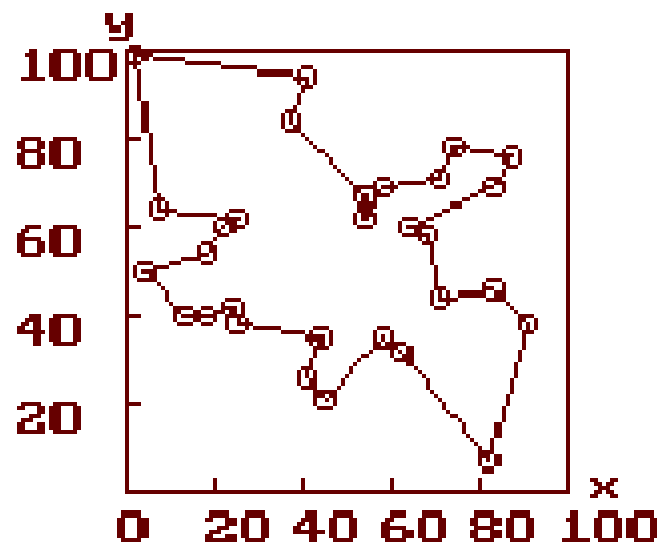
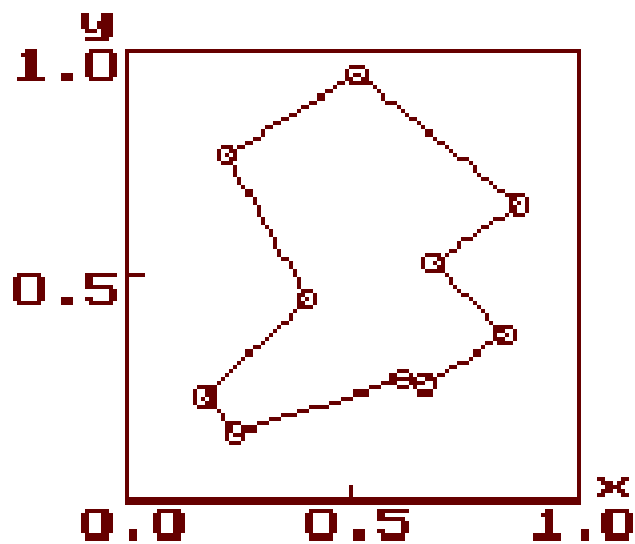
表 4.5.1 10 城市 TSP 的 ACNN 模型优化统计性能

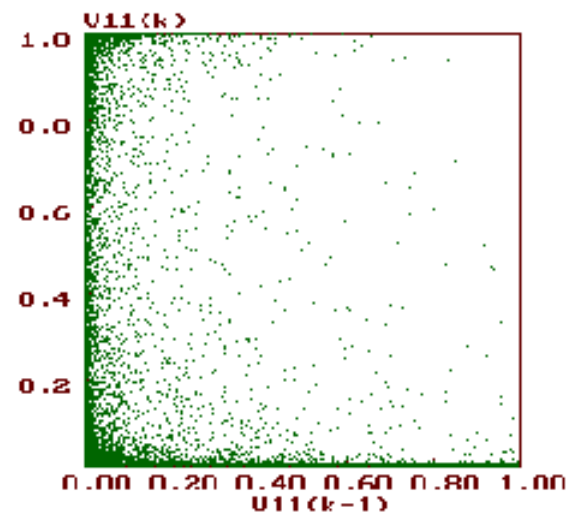
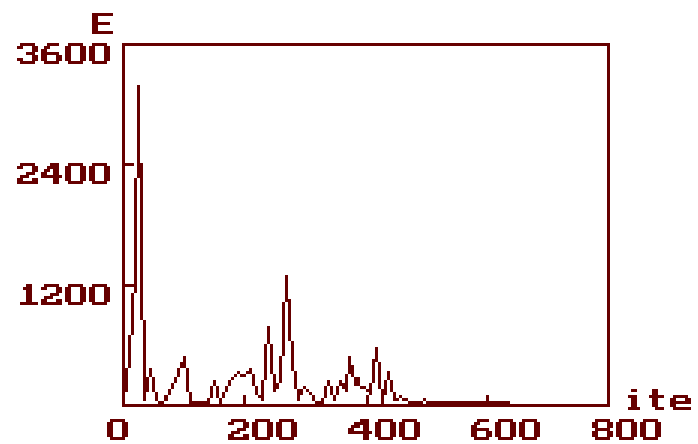
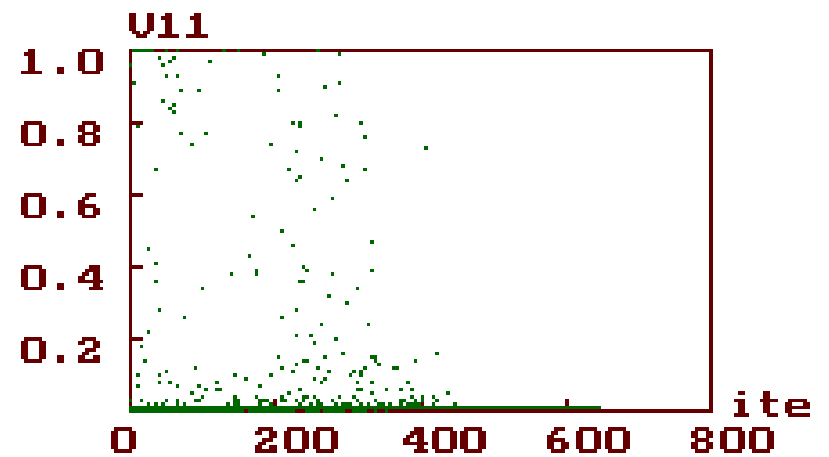
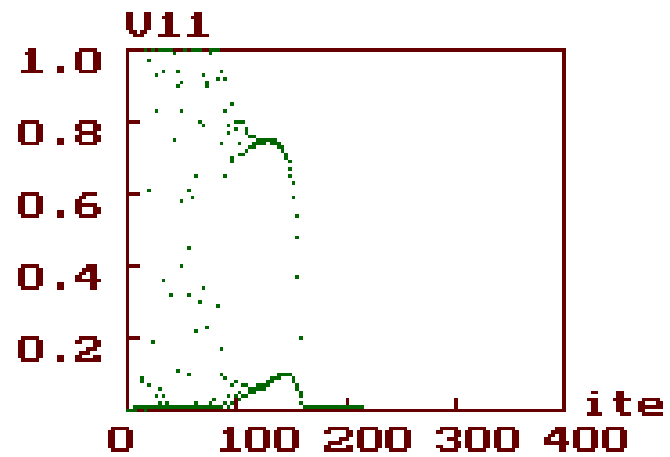
序号	β	α	λ	$z(0)$	最优解 数目	次优解 数目	非法解 数目	波动率 %	平均迭代 步数
1	0.02	0.88	0.01	0.05	50	0	0	0	216
2	0.018	0.88	0.01	0.05	49	1	0	0.22	239
3	0.015	0.88	0.01	0.05	42	5	3	1.26	293
4	0.022	0.88	0.01	0.05	43	7	0	1.71	194
5	0.02	0.90	0.01	0.05	50	0	0	0	200
6	0.02	0.86	0.01	0.05	47	3	0	0.93	225
7	0.02	0.88	0.008	0.05	48	2	0	0.63	259
8	0.02	0.88	0.012	0.05	47	3	0	0.74	187
9	0.02	0.88	0.01	0.06	50	0	0	0	267
10	0.02	0.88	0.01	0.04	39	11	0	1.93	147

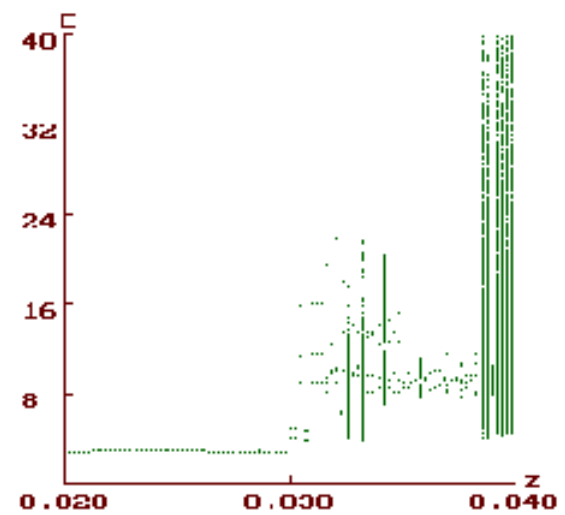
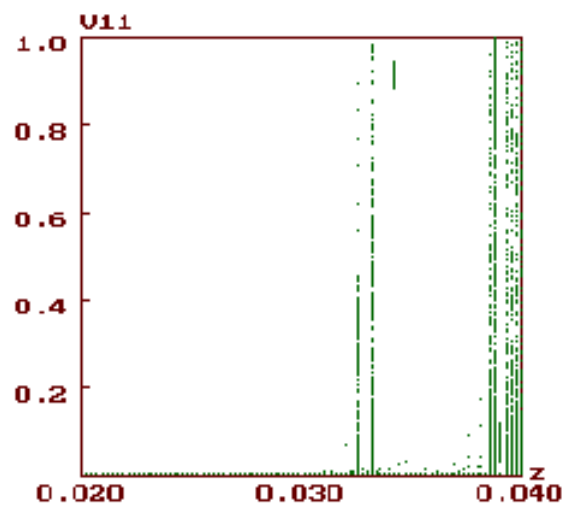
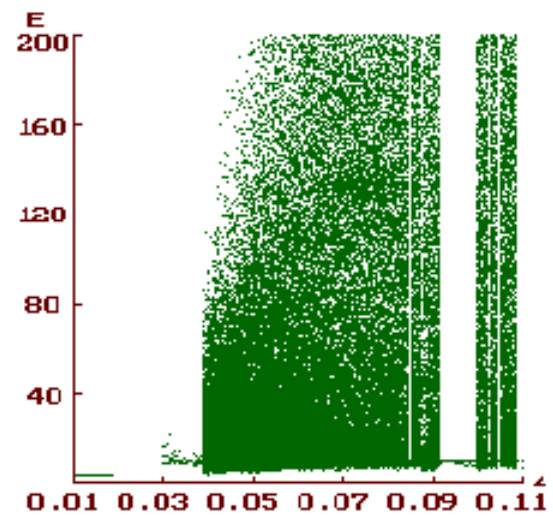
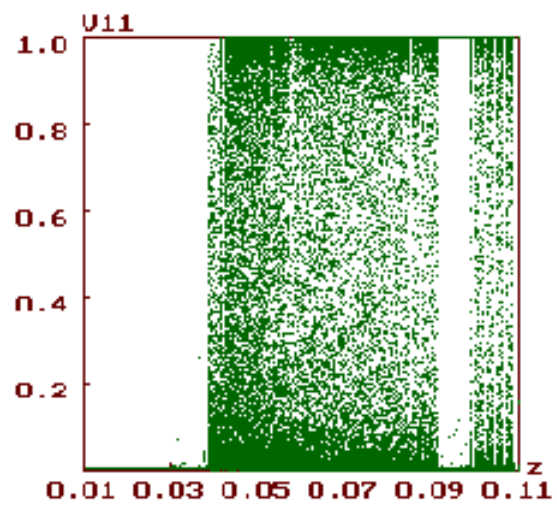
基于ACNN的TSP优化

表 4.5.2 基于 30 城市 TSP 的 ACNN 模型与 SA 的性能比较

算法	最优解 次数	相对误差小于 3%的次优解次数	非法解 次数	波动率 %	平均 迭代步数
ACNN	15	15	0	6.27	653
SA	3	10	0	10.47	28000







模型参数对算法性能的影响

1. 尺度参数 β 的影响

β 增大有利于加速收敛，但优化质量会有所下降。从网络动态方程知， β 增大使能量函数对动态方程的影响增大。若影响过大，将不能产生充分的混沌动态，从而影响搜索性能，如次优解增多；反之，太小的 β 使能量函数的变化不能充分影响动态的演变，从而搜索过程难以收敛到对应最小能量值的最优解，同时过分的混沌搜索将导致优化过程变长。

收敛，但会出现类似 β 过大的现象。

$$\begin{aligned} u_{xi}(k+1) &= (1 - \Delta t / \tau) \cdot u_{xi}(k) + \Delta t \cdot \beta' \cdot [\sum_y \sum_j w_{xi,yj} v_{yj}(k) + I_{xi}] \\ &= \alpha \cdot u_{xi}(k) + \beta \cdot [\sum_v \sum_i w_{xi,yj} v_{yj}(k) + I_{xi}] \\ u_{xi}(k) &= \alpha \cdot u_{xi}(k-1) + \beta \cdot [\sum_y \sum_j w_{xi,yj} v_{yj}(k-1) + I_{xi} - s_1] \\ &\quad - z(k-1) \cdot [v_{xi}(k-1) - s_2] \end{aligned}$$

模型参数对算法性能的影响

2. 阻尼因子 α 的影响

α 增大有利于加速收敛，并减少算法收敛到次优解的概率。通过对下式分析知， α 减小相当于增大仿真步长 Δt ，若算法中 β 未变，则 Δt 增大相当于 β' 减小，从而使能量函数的作用变弱，进而导致迭代步数增大和优化性能变差。同理， α 过大虽能加速收敛，但会出现类似 β 过大的现象。

$$\begin{aligned} u_{xi}(k+1) &= (1 - \Delta t / \tau) \cdot u_{xi}(k) + \Delta t \cdot \beta' \cdot [\sum_y \sum_j w_{xi,yj} v_{yj}(k) + I_{xi}] \\ &= \alpha \cdot u_{xi}(k) + \beta \cdot [\sum_y \sum_j w_{xi,yj} v_{yj}(k) + I_{xi}] \\ u_{xi}(k) &= \alpha \cdot u_{xi}(k-1) + \beta \cdot [\sum_y \sum_j w_{xi,yj} v_{yj}(k-1) + I_{xi} - s_1] \\ &\quad - z(k-1) \cdot [v_{xi}(k-1) - s_2] \end{aligned}$$

模型参数对算法性能的影响

3. 退温参数 λ 的影响

λ 增大有利于加速收敛，但优化质量会下降。 λ 的大小反映控制参数 z 下降速率的快慢，过大的 λ 会使混沌动态消失过快，从而 ACNN 模型过早转化为 Hopfield 模型，进而收敛到局部极小或非法状态；反之，混沌动态持续太久，将严重影响收敛速度。

$$\left\{ \begin{array}{l} u_{xi}(k) = \alpha \cdot u_{xi}(k-1) + \beta \cdot [\sum_y \sum_j w_{xi,yj} v_{yj}(k-1) + I_{xi} - s_1] \\ \quad - z(k-1) \cdot [v_{xi}(k-1) - s_2] \\ v_{xi}(k) = 1/[1 + \exp(-\mu \cdot u_{xi}(k))], \quad x, i, y, j = 1, 2, \dots, n \\ z(k) = z(k-1) / \ln[e + \lambda \cdot (1 - z(k-1))], \quad k = 1, 2, \dots, m \end{array} \right.$$

模型参数对算法性能的影响

4. 初始控制参数 $z(0)$ 的影响

$z(0)$ 减小有利于加快收敛，但算法收敛到次优解的概率将增加。由动态方程和退温函数知， $z(0)$ 减小使 $z(k+1)/z(k)$ 减小，即加快了 z 的下降过程，但一旦混沌行为不充分，则必然影响优化性能。反之， $z(0)$ 过大将使其下降过程过慢，从而混沌对系统动态演化过程的影响过大，收敛必然缓慢。

此外，若问题的规模增大，神经元数量和系统动态的复杂性均将增加。仿真表明，此时参数 B 应选择得比参数 A 小些，以均衡能量函数中各项的作用；同时， $z(0)$ 需适当增加， β 和 λ 需适当减小。

$$\begin{cases} u_{xi}(k) = \alpha \cdot u_{xi}(k-1) + \beta \cdot [\sum_y \sum_j w_{xi,yj} v_{yj}(k-1) + I_{xi} - s_1] \\ \quad - z(k-1) \cdot [v_{xi}(k-1) - s_2] \\ v_{xi}(k) = 1/[1 + \exp(-\mu \cdot u_{xi}(k))], \quad x, i, y, j = 1, 2, \dots, n \\ z(k) = z(k-1) / \ln[e + \lambda \cdot (1 - z(k-1))], \quad k = 1, 2, \dots, m \end{cases}$$