

进化策略求解 Hopfield 神经网络

黎明 严超华 刘高航

(南昌航空工业学院测试与控制工程系, 南昌 330034)

摘要 提出了一种进化策略求解 Hopfield 神经网络的方法. 该进化策略分三个阶段, 即第一阶段只在较小区间上求出局部优化解; 然后, 在此基础上, 由第二阶段求出较大区间上的局部优化解; 最后由第三阶段求出全局优化解. 同时采用 Hopfield 神经网络动态方程指导第一阶段的局部进化策略的进化方向, 因而大大加快了优化搜索速度. 在分阶段的进化策略中, 其第一阶段只需搜索较小区间, 第二和第三阶段的搜索则建立在其前一阶段的基础上, 因此可以采用较小的遗传群体规模, 从而明显地提高了求解 Hopfield 神经网络的速度, 减少了计算内存的需求.

关键词 模式识别 进化策略 Hopfield 神经网络 优化

中图分类号: TP183 文献标识码: A 文章编号: 1006-8961(2000)07-0603-06

Converging Hopfield Neural Network by Evolutionary Strategies

LI Ming, YAN Chao-hua, LIU Gao-hang

(Dept. of Applied Engineering, Nanchang Institute of Aeronautical Technology, Nanchang 330034)

Abstract The method of applying evolutionary strategies to converge Hopfield neural network is proposed in this paper. Genetic search of evolutionary strategies(ES) consists of three consecutive processes. ES performs local search in some small domain in the first process, and the search domain grows in the second process. Then ES performs the global search in whole domain in the third process. We use the dynamic equation to indicate the ES search direction to speed up the local search speed obviously. The ES search domain is small in the first process, and the second and the third process are based on the optimization results of their previous processes, therefore the convergence speed of the proposed method is much faster and it needs less memory space.

Keywords Pattern recognition, Evolutionary strategies, Hopfield neural network, Optimization

0 引言

遗传理论作为一种有效的优化理论越来越广泛地应用于求解各种优化问题. 这种遗传理论包括遗传算法、进化策略和进化规划^[1,2]三种算法, 它们都是在模仿生物遗传进化理论的基础上发展而来的. Hopfield 神经网络^[3]也是一种优化算法, 它模仿的是人类大脑的神经网络活动. 遗传理论与 Hopfield 神经网络有明显的共同点, 即它们都是模仿生命原理的模型, 都可以进行并行运算. 但是它们也有明显的差异, 这种差异主要表现在遗传理论是一种全局优化理论, 因此它首先需随机设置初始的遗传群体, 然后通过迭代遗传运算找到全局优化解, 其初始遗

传群体的设置只影响收敛速度, 一般不影响优化结果; 而 Hopfield 神经网络则是一种局部优化理论, 它具有很强的容错能力, 但它仅在神经网络的初始状态邻域内进行搜索, 因而使得网络的能量最低, 因此其求出的优化解与神经网络的初始状态有关.

许多研究者将遗传理论与 Hopfield 神经网络相结合, 以获得两种方法的优点. Lai 采用遗传理论优化设计了 Hopfield 神经网络的控制参数^[4], Toshio 则利用遗传算法优化了 Hopfield 神经网络的权重矢量^[5], 但是, 这两种方法虽然简化或优化了 Hopfield 神经网络设计过程, 可仍然未改变 Hopfield 神经网络的局部优化特性. Hiroyuki 利用遗传算法成功地求解了 Hopfield 神经网络^[6], 从而提高了网络收敛到全局能量最小点的概率. 但

Hiroyuki 方法难以求解较大规模的 Hopfield 神经网络, 例如, 求解用于图象边界检测的 Hopfield 神经网络时, 可能需用 $1\ 024 \times 1\ 024$ 个神经元, 由于遗传算法的二值编码太长, 使得每次遗传运算过程很长, 因而网络收敛速度较慢, 同时需要较大内存. 如果用缩小每个遗传运算过程中遗传群体规模的方法, 来提高每个遗传运算过程的运算速度和减少内存的需求, 则需要更多的遗传运算过程才能使网络收敛, 因而总的遗传收敛速度仍然较慢, 而且若遗传群体的规模太小, 甚至还会引起遗传算法不收敛.

本文提出了一种采用进化策略求解 Hopfield 神经网络的新方法. 该进化策略采用多值编码, 并对网络的每个神经元输出进行直接的操作. 在本文的方法中, 进化策略是分三个阶段搜索的, 即第一阶段首先将搜索空间分成一些离散的较小区间, 然后进化策略在各个较小的区间内进行搜索, 以找出各个区间内的局部优化解; 第二阶段则将第一阶段的较小区间相互合并, 形成较大的搜索空间, 并将第一阶段搜索到的局部优化解作为进化策略的初始群体, 再找出各个较大的区间内的局部优化解; 第三阶段再将第二阶段搜索到的局部优化解作为进化策略的初始群体, 然后找出全局优化解.

1 Hopfield 神经网络

设 i, j 表示 $M \times N$ 神经元阵列的水平方向坐标, k, l 表示垂直方向坐标. 设 u_{ik} 为神经元 (i, k) 的输入, V_{ik} 为神经元 (i, k) 的输出, τ, R_{ik}, I_{ik} 为能量函数参数, t 为时间变量, $\omega_{k, jl}$ 为神经元 (i, k) 和神经元 (j, l) 之间的权重系数, 则 Hopfield 神经网络的动态方程为

$$\frac{du_{ik}}{dt} = -\frac{u_{ik}}{\tau} + \sum_j \sum_l \omega_{k, jl} V_{jl} + I_{ik} \quad (1)$$

更新神经元输出状态的 Sigmoidal 变换为

$$V_{ik} = f_{ik}(u_{ik}) = \frac{1}{2} \left| 1 + \tanh \left| \frac{u_{ik}}{u_0} \right| \right| \quad (2)$$

式中, u_0 为规一化基准值, 当 $u_0 \rightarrow 0$ 时, $f_{ik}(\cdot)$ 成为硬限幅函数. Hopfield 网络的能量函数为

$$E = -\frac{1}{2} \sum_i \sum_k \sum_j \sum_l \omega_{k, jl} V_{ik} V_{jl} + \sum_i \sum_k \frac{1}{R_{ik}} \int_0^{V_{ik}} f_{ik}^{-1}(\xi) d\xi - \sum_i \sum_k I_{ik} V_{ik} \quad (3)$$

这里需指出, 当权重矢量设置为对称时, Hopfield 神经网络随着动态方程的迭代, 网络能量

将不断降低, 同时, 由于网络收敛结果与网络初始状态有关, 故可能使网络收敛于能量局部极小值, 从而得不到 Hopfield 神经网络的全局最优化解.

2 进化策略求解 Hopfield 神经网络

本文采用 (μ, λ) 进化策略来求解 Hopfield 神经网络, 以提高网络收敛到全局能量最小点的概率. 本文采用的进化策略是将 Hopfield 神经网络的解(神经元阵列的输出状态), 编码为实值码串, 然后由不同的码串组成遗传群体. 而进化策略则对码串进行进化操作, 以产生新的码串, 进而组成新的遗传群体, 并通过选择操作, 使新的遗传群体中的码串具有更高的适应度, 也就是使与新码串对应的 Hopfield 神经网络输出状态具有更低的能量.

其具体实现则是首先将 Hopfield 神经网络的神经元阵列的输出状态, 编码成实值码串, 则每个码串就表示一种求得的 Hopfield 神经网络的解. 对于网络输出状态 $\mathbf{V} = \{V_{ik}: i \in [1, M], k \in [1, N]\}$, 其进化策略的码串为 $\mathbf{p}(\mathbf{V}) \equiv \{V_{ik}, \sigma_{ik}: i \in [1, M], k \in [1, N]\}$, 它是一 $M \times N$ 维矢量, 其中, σ_{ik} 为对应于网络输出状态分量 V_{ik} 的进化步进方差. 码串 $\mathbf{p}(\mathbf{V})$ 的适应度为

$$fitness(\mathbf{V}) = 1/E(\mathbf{V}) \quad (4)$$

其中, $E(\mathbf{V})$ 是输出状态下的网络能量, 可由(3)式来进行计算. 进化策略的重组操作、变异操作和选择操作参见文献[1].

正如第一节中所述, 采用 (μ, λ) 进化策略求解大规模的 Hopfield 神经网络的缺点是其码串很长, 因而使得每次进化运算过程的重组操作、变异操作所需计算时间很长, 且网络收敛速度较慢. 同时很长的码串也需要很大的内存, 例如求解 $1\ 024 \times 1\ 024$ 个神经元的 Hopfield 神经网络, 当遗传策略的参数 $\mu = 50$ 时, 需要大于 400MB 的内存, 从而使得难以在微机上实现进化策略. 因此, 本文提出了一种采用分阶段搜索的 (μ, λ) 进化策略来求解大规模的 Hopfield 神经网络的方法, 并且利用 Hopfield 神经网络动态方程求出进化策略的进化方向, 因而进一步加快了求解速度.

所谓分阶段搜索的 (μ, λ) 进化策略就是分三个阶段来搜索 Hopfield 神经网络的优化解, 其第一阶段在较小的范围内由动态方程指导搜索 Hopfield 神经网络的局部优化解; 第二阶段则是以第一阶段得

到的局部优化解为基础, 在更大的范围内搜索局部优化解; 第三阶段搜索又以第二阶段搜索得到的局部优化解为基础, 在全局范围内搜索 Hopfield 神经网络的优化解。

(μ, λ) 进化策略的三阶段搜索过程如图 1 所示。

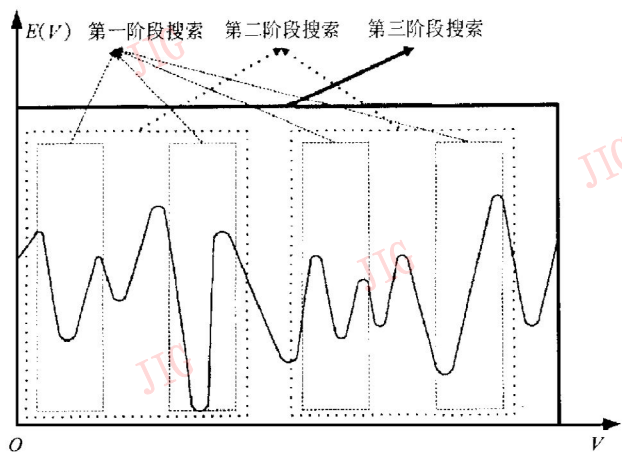


图 1 (μ, λ) 进化策略的三阶段搜索过程

(1) 第一阶段的(μ, λ) 进化策略

第 1 步 将神经元阵列输出状态编码为实值码串 $p(\mathbf{V})$, 并将码串值 \mathbf{V} 所有可能的取值形成 $M \times N$ 维矢量空间 $R^{M \times N}$: $R^i = [0, 1], i \in [1, (M \times N)]$, 再在 $R^{M \times N}$ 中选择多个离散区间组成集合 Φ :

$$\Phi = \left\{ \left| \mathbf{V}^d, \mathbf{V}^d + \Delta \mathbf{V}^d : d = \left\lfloor 0, \frac{1}{K^0 + K^1} - 1 \right\rfloor \right. \right\} \quad (5)$$

$$\mathbf{V}^d = \left| V_{ik}^d : i \in [1, M], k \in [1, N] \right| \quad (6)$$

$$V_{ik}^d = \frac{d}{K^0 + K^1} \quad (7)$$

$$\Delta \mathbf{V} = \left| \Delta v_{ik} = K^0 : i \in [1, M], k \in [1, N] \right| \quad (8)$$

其中, K^0 为码串每个分量搜索区间的尺寸, K^1 为码串每个分量搜索区间的间隔距离, 它们的取值范围为 $K^0 \in (0, 1), K^1 \in (0, 1)$, 这两个参数在进化策略运算过程之前设定. 因此第一阶段共在搜索空间选择了 $D = \frac{1}{K^0 + K^1}$ 个等大小和等间隔的搜索区间。

第 2 步 按顺序选择上述 D 个搜索区间, 进行以下的进化策略运算。

第 3 步 在被选择的搜索区间上, 随机产生由 μ 个码串组成的初始进化群体, 并作为此次运算过程的父母码串。

第 4 步 以均等的概率, 随机地从父母码串中选择两个码串, 先进行重组操作, 以产生一暂时的后代码串 $p(\mathbf{V}^A) = \left| V_{ik}^A, \sigma_{ik}^A : i \in [1, M], k \in [1, N] \right|$ 。

第 5 步 在 Hopfield 神经网络动态方程指导下, 对此暂时的后代码串 $p(\mathbf{V}^A)$ 进行变异操作, 以产

生一个后代码串, 即首先将与码串 $p(\mathbf{V}^A)$ 对应的网络输出状态 $\mathbf{V}^A = \left| V_{ik}^A : i \in [1, M], k \in [1, N] \right|$ 放入 Hopfield 神经网络, 作为网络的初始输出状态, 然后根据式 (1) 和式 (2) 描述的 Hopfield 神经网络动态方程和输出状态更新方程, 对网络的初始输出状态进行 R 次的迭代运算, 迭代次数 R , 一般可选择为 $R = [5, 10]$. 设 R 次迭代后, 网络输出状态为 $\mathbf{V}^B = \left| V_{ik}^B : i \in [1, M], k \in [1, N] \right|$, 由于 Hopfield 神经网络随着动态方程的迭代, 网络能量不断降低, 因此与输出状态 \mathbf{V}^B 对应的适应度 $fitness(\mathbf{V}^B)$ 大于与输出状态 \mathbf{V}^A 对应的适应度 $fitness(\mathbf{V}^A)$, 也就是说, 网络动态方程给出了进化策略的进化方向. 在网络动态方程指导下的变异操作, 则将码串 $p(\mathbf{V}^A) = \left| V_{ik}^A, \sigma_{ik}^A : i \in [1, M], k \in [1, N] \right|$ 变异为码串 $p(\mathbf{V}^C) = \left| V_{ik}^C, \sigma_{ik}^C : i \in [1, M], k \in [1, N] \right|$:

$$\Delta v_{ik} = V_{ik}^B - V_{ik}^A \quad \forall i \in [1, M], k \in [1, N] \quad (9)$$

$$\sigma_{ik}^C = \sigma_{ik}^A \cdot N \left[0, \left| 1 + \frac{|\Delta v_{ik}|}{R} \right| \cdot \Delta \sigma \right] \quad (10)$$

$$\forall i \in [1, M], k \in [1, N]$$

$$V_{ik}^C = V_{ik}^A + N(\Delta v_{ik}, \sigma_{ik}^C) \quad (11)$$

$$\forall i \in [1, M], k \in [1, N]$$

式 (10) 中的 $N \left[0, \left| 1 + \frac{|\Delta v_{ik}|}{R} \right| \cdot \Delta \sigma \right]$ 是一个均值为 0, 标准偏差为 $\left| 1 + \frac{|\Delta v_{ik}|}{R} \right| \cdot \Delta \sigma$ 的 Gaussian 随机变量. 网络动态方程指导下的变异操作与标准进化策略的变异操作相比, 网络动态方程的指导也给出了各编码分量标准偏差的变化方向, 即如果 $\frac{|\Delta v_{ik}|}{R}$ 较大, 则表明网络能量多维曲线在输出状态 \mathbf{V}^A 的附近变化较快, 且与 \mathbf{V}^A 相应的码串的进化步进方差也应较大. 式 (11) 中的 $N(\Delta v_{ik}, \sigma_{ik}^C)$ 是一个均值为 Δv_{ik} , 标准偏差为 σ_{ik}^C 的 Gaussian 随机变量, 而均值 Δv_{ik} 则给出了由网络动态方程指导的进化策略的变异进化方向。

如果由网络动态方程指导的变异操作, 使对应的输出状态 \mathbf{V}^C 超出了所搜索的区间, 则说明与本区间外的输出状态对应的网络能量比输出状态 \mathbf{V}^A 的网络能量低, 此时将 $p(\mathbf{V}^A)$ 作为一个满足下述第 7 步适应度收敛条件的后代码串, 并将 λ 减 1. 如果 λ 被减为零, 则说明除了边界以外, 与本区间内的输出状态对应的网络能量无极小值。

第 6 步 重复第 3 步和第 5 步, 直到获得 λ 个后代码串, 通过选择操作, 从此 λ 个后代码串中选择

μ 个码串作为下次运算过程的父母码串。

第7步 重复第3步至第6步,直到 λ 个后代码串的适应度收敛到一定的精度。

第8步 重复第2步至第7步,分别求出 D 个搜索区间上的局部最优解。

在上述第一阶段搜索步骤中,本文提出的方法从下面三个方面加快了 (μ, λ) 进化策略的收敛速度:

① 由于第一阶段每个搜索区间较小,因此对每个搜索区间所实施的进化策略的参数 μ 和 λ 都可以选得较小,从而使得进化策略对每个区间的运算次数减少;

② 由于搜索区间是离散分布的,而且其中有 $D = \frac{1}{K^0 + K^1}$ 个等大小的区间不搜索,因此实际上进

化策略只搜索 $\frac{K^0}{K^0 + K^1} \times 100\%$ 的全部空间;③ 在网络动态方程的指导下,式(9)~式(11)可使得进化策略迅速收敛至局部最优值。

(2) 第二阶段的 (μ, λ) 进化策略

第二阶段的 (μ, λ) 的进化策略与第一阶段的 (μ, λ) 进化策略类似,都是进行局部的搜索.所不同的有以下三方面:局部搜索区域的选择;初始进化群体的选择;变异操作不同。

该阶段进化策略主要是将第一阶段的多个相邻的搜索区间合并,并包含这些区间之间的未搜索区间,以形成如图1所示的第二阶段搜索区间.由于这些较大的搜索区间互不交叠,并且等大小、等间隔,因此所有第二阶段搜索区间的集合包含了所有第一阶段搜索区间的集合。

若设第二阶段的每个搜索区间由 K 个相邻的第一阶段的搜索区间合并,则它就继承了这 K 个第一阶段搜索区间的 $K \times \lambda$ 个局部优化解,再从这 $K \times \lambda$ 个局部优化解中,随机选择 μ 个码串就组成了第二阶段的初始进化群体,然后进行标准的进化策略的重组操作、变异操作和选择操作,以得到 λ 个适应度收敛到一定精度的后代码串,即第二阶段的变异操作不再采用动态方程的指导,以便搜索出更大范围内可能存在的适应度更高的局部优化解。

第二阶段搜索以第一阶段搜索得到的局部优化解为基础,但仍然是局部搜索,每个搜索区间仍较小,因此进化策略的参数 μ 和 λ 仍可以选得较小,这将使得第二阶段进化策略收敛速度很快。

(3) 第三阶段的 (μ, λ) 进化策略

第三阶段的 (μ, λ) 的进化策略是进行全局搜索,

即将第二阶段搜索得到的所有区间的局部优化解组成一个集合,再从这个集合中随机选择 μ 个码串组成第三阶段初始进化群体,然后进行标准的进化策略的重组操作、变异操作和选择操作,以使进化策略运算收敛,从而得到Hopfield神经网络的全局优化解.同样,第三阶段搜索是以第二阶段搜索得到的局部最优解为基础,其进化策略的参数 μ 和 λ 仍可以选得较小,以便使得第三阶段进化策略收敛速度很快。

3 实验结果与结论

采用Hopfield神经网络方法检测图象边界^[7],需要使用较大规模的Hopfield神经网络,检测如图2所示的 252×238 象素点的Lena图象,可将整幅被测图象看作为一个Hopfield神经网络,每一象素点是一神经元,每一神经元都与其它神经元相连,但不与自身连接,而且任意两象素点之间的灰度梯度及对应的两象素之间距离即定义了对应两神经元之间的连接权重系数^[7]。

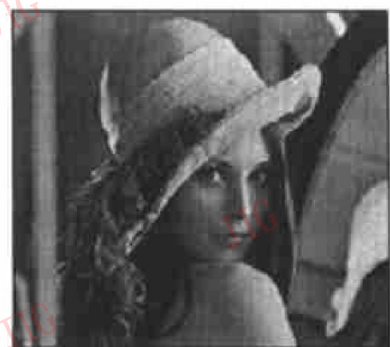


图2 原始Lena图象

在传统的Hopfield神经网络求解方法中,每一神经元的初始状态由其对应的被测图象中象素点的归一化灰度所决定,然后每一神经元的状态被迭代改变,直到整个神经网络收敛.网络收敛后得到的图象边界如图3所示。



图3 传统方法求出的边界

为了对实值编码进行直接操作和便于与本文方法试验结果对比, 因此将 Hiroyuki^[6] 求解 Hopfield 神经网络方法中的遗传算法改为 (μ, λ) 进化策略, 称之为直接的 (μ, λ) 进化策略求解 Hopfield 神经网络方法. 它不进行分段搜索, 也不采用 Hopfield 神经网络动态方程指导, 而是采用直接的 (μ, λ) 进化策略方法求解上述 Hopfield 神经网络, 其检测到的图象边界如图 4 所示. 此方法参数选择为 $\mu=40, \lambda=50$, 在一定进化代数的进化过程中, 如果个体适应度提高不到 0.3% 时, 则判定进化策略收敛, 或者说这种进化策略方法满足了 0.3% 的收敛精度.



图 4 直接进化策略求出的边界

采用本文提出的 (μ, λ) 进化策略方法求解上述 Hopfield 神经网络, 其检测到的图象边界如图 5 所示. 本文提出的 (μ, λ) 进化策略方法, 其第一阶段进化策略参数选择为 $\mu=10, \lambda=15, K^0=0.015, K^1=0.005$, 收敛精度选为 3%; 第二阶段进化策略参数选择为 $\mu=10, \lambda=15, K=5$, 收敛精度选为 1%; 第三阶段进化策略参数选择为: $\mu=15, \lambda=25$, 收敛精度选为 0.3%.



图 5 本文方法求出的边界

由于标准的 (μ, λ) 进化策略方法的初始进化群体尺寸选得太小, 会使初始进化群体中个体不具有多样性, 因而导致进化策略不收敛. 在试验中, 当用直接的 (μ, λ) 进化策略方法求解上述 Hopfield 神经网络时, 当选择 $\mu=20$ 时, 则 (μ, λ) 进化策略发散,

致使搜索不到 Hopfield 神经网络的最优解; 而用本文提出的分阶段和动态方程指导下的 (μ, λ) 进化策略求解上述 Hopfield 神经网络时, 其参数 μ 和 λ 都选得小得多, 但仍然可使 (μ, λ) 进化策略收敛.

由图 3~ 图 5 可见, 传统的 Hopfield 神经网络求解方法、求解上述 Hopfield 神经网络的直接 (μ, λ) 进化策略方法和本文提出的分阶段和动态方程指导下的 (μ, λ) 进化策略方法, 其检测到的图象边界质量基本相同. 这三种方法的计算时间和内存需求如表 1 所示. 实验是在 Pentium II / 233, 64MB RAM 的微机实现的. 实验结果表明, 求解较大规模 Hopfield 神经网络时, 本文提出的 (μ, λ) 进化策略方法比传统的方法和直接的 (μ, λ) 进化策略方法收敛速度快得多, 所需内存也比直接的 (μ, λ) 进化策略方法少得多, 但同样求出了 Hopfield 神经网络的全局最优解.

表 1 三种求解 Hopfield 神经网络方法的计算时间和内存需求对比

	计算时间(s)	内存需求(MB)
传统的求解方法	322	约 1
直接的 (μ, λ) 进化策略方法	68	约 21
本文提出的 (μ, λ) 进化策略方法	14	约 9

笔者还将这三种方法用于求解 30 节点的 Traveling salesman problem (TSP) 问题, 这是一个较小规模的 Hopfield 神经网络求解问题. 传统求解方法的结果与其初始状态有关, 在实验中, 一次随机选择初始状态后, 传统求解方法的优化结果如图 6 所示. 而采用直接的 (μ, λ) 进化策略方法和本文提出的进化策略方法来进行 Hopfield 神经网络的求解,

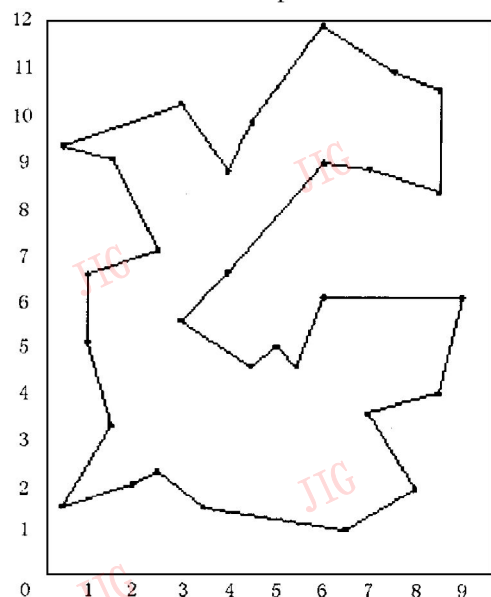


图 6 传统 Hopfield 神经网络方法求解 TSP 结果

其优化结果相同(如图7所示). 传统的 Hopfield 神经网络求解方法只获得了初始状态邻域内的局部优化解, 而直接的 (μ, λ) 进化策略方法和本文提出的 (μ, λ) 进化策略方法则获得了全局的最优解. 当然, 实验中传统 Hopfield 神经网络求解方法若适当选择初始状态也可获得图7所示的全局最优解. 用直接的 (μ, λ) 进化策略方法和本文提出的 (μ, λ) 进化策略方法对 TSP 问题求解, 其收敛时间分别为 13.5s 和 6.7s, 由此可见, 在求解较小规模的 Hopfield 神经网络时, 本文提出的 (μ, λ) 进化策略方法收敛速度仍快得多. 无论对于求解较大规模或较小规模的 Hopfield 神经网络, 本文提出的分段搜索和有动态方程指导的 (μ, λ) 进化策略方法都容易进行并行运算, 以实现实时求解.

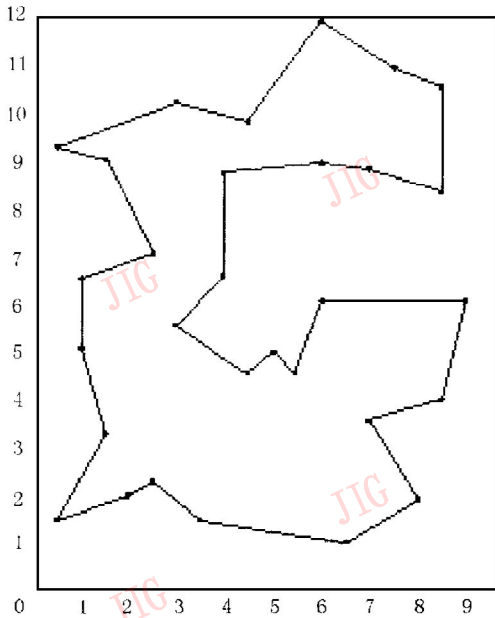


图7 进化策略结合 Hopfield 神经网络求解 TSP 结果

本文提出的一种新的进化策略求解 Hopfield 神经网络方法, 可通过分阶段搜索和利用 Hopfield 神经网络动态方程的指导, 还可以采用较小的遗传群体规模参与运算, 从而明显地提高了求解 Hopfield 神经网络的速度, 且减少了计算内存的需求.

参考文献

1 David B F. An introduction to simulated evolutionary optimization. IEEE Transactions on Neural Networks, 1994, (5): 3~ 14.

- 2 Yao X. A reviews of evolutionary artificial neural networks. International Journal of Intelligent Systems, 1993, (8): 539~ 567.
- 3 Hopfield J J, Tank D E. "Neural" computation of decisions in optimization problems. Biol. Cybern., 1985, (52): 141~ 152.
- 4 Lai W K, Coghill G G. Genetic breeding of control parameters for the Hopfield/Tank neural net. In: Proc. Of International Joint Conference on Neural Networks, Baltimore, USA, 1992, 618 ~ 623.
- 5 Toshio F, Tadashi K, Takanori S. Heuristic leaning by genetic algorithm for recurrent neural network. In: Proc. of the 32nd IEEE Conference on Decision and Control, San Antonio, USA, 1993, 71~ 77.
- 6 Hiroyuki S, Atsushi I, Shunji K, Tsuneo T. A solution of combinatorial optimization problem by uniting genetic algorithms with Hopfield's model. In: Proc. of 1994 IEEE International Conference on Neural Networks, Orlando, USA, 1994, 4704~ 4709.
- 7 Chao C H, Dhawan A P. Edge detection using a Hopfield neural network. Optical Engineering, 1994, (33): 3739~ 3747.



黎明 博士, 副教授. 主要从事图象处理、模式识别、神经网络和遗传算法方面的研究工作, 已发表学术论文 30 余篇.



严超华 副教授. 主要从事图象测试、神经网络和遗传算法方面的研究工作.

刘高航 教授. 主要从事图象处理、模式识别、神经网络方面的研究工作.