基于协同进化的异构种群挖掘混沌迭代函数

郑皎凌[1),2)] 唐常杰[1)] 徐开阔[1)] 陈 瑜[1)] 杨 宁[1)] 段 磊[1)]

[1)](四川大学计算机学院数据库与知识工程研究所 成都 610065)

[2](成都信息工程学院软件工程系 成都 610225)

摘 要 混沌迭代序列是复杂系统动力学研究的一个分支,其序列值在不同参数条件下时会出现分叉及混沌现象.已有的方法不能同时挖掘拟合迭代序列的迭代函数的结构及其相应条件参量.文章则旨在同时挖掘出二者,主要工作包括:(1)提出了基于协同进化的异构种群挖掘模型,能融合不同种群的优势;(2)提出了新的适合挖掘迭代序列的适应度计算方式;(3)从理论上证明了多种群协同挖掘的进化难度远大于单种群进化难度,通过实验证实了在有效协同策略下,多种群进化得到的结果远优于单种群的进化结果;(4)提出 3 种协同进化策略,在对迭代序列的函数拟合以及参数拟合两方面,多路并行式结合策略能达到相对较优效果;(5)在合成数据和真实数据上进行了实验,证实了算法的正确性和有效性.

关键词 混沌迭代序列;协同进化模型;挖掘模型;异质种群;种群结合策略

中图法分类号 TP311 **DOI 号**: 10.3724/SP.J.1016.2010.00672

Mining Chaotic Iterative Functions by Co-evolution over Heterogeneous Populations

ZHENG Jiao-Ling[1),2)] TANG Chang-Jie[1)] XU Kai-Kuo[1)] CHEN Yu[1)] YANG Ning[1),2)] DUAN Lei[1)]

[1)](*Institute of Database and Knowledge Engineering*, *School of Computer Science*, *Sichuan University*, *Chengdu* 610065)

[2)](*Department of Software Engineering*, *Chengdu University of Information Technology*, *Chengdu* 610225)

Abstract Chaotic iterative sequence is a research direction in complex system kinetics research. The sequence may incarnate bifurcate or chaotic phenomenon under different parameter conditions. The existing methods cannot discover the iterative structure and parameters simultaneously. This study aims at mining the iterative functions and conditional parameters parallel. The main contributions include: (1) Proposes co-evolution model based on heterogeneous populations to integrate the advantages of those populations. (2) Proposes a new fitness function to mine the sequence in iterative style. (3) Theoretically proves that heterogeneous populations' co-evolution is more difficult than a single population's evolution. Experimentally proves that given effective co-evolution strategy, heterogeneous populations can obtain much better results than single population. (4) Proposes three co-evolution strategies. The cooperating strategy can archive relatively good results in terms of fitting the sequence's mathematic equation and the equation's parameters. (5) Conducts extensive experiments on both synthesized and real data to validate the correctness and efficiency of the algorithm.

Keywords chaotic iterative sequence; co-evolutionary model; mining model; heterogeneous population; population cooperating strategy

---

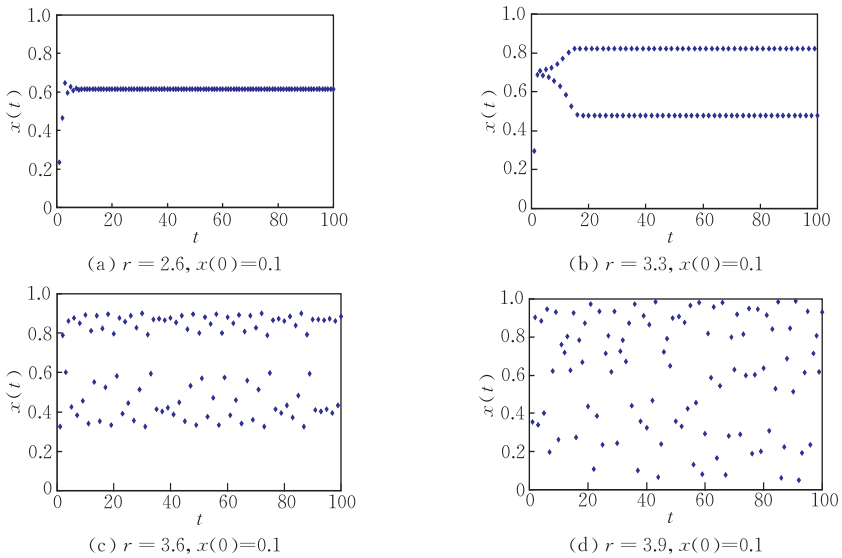
收稿日期:2009-05-09;最终修改稿收到日期:2010-03-11.本课题得到国家"十一五"科技攻关项目基金(2006BAI05A01)和国家自然科学基金(60473071)资助.**郑皎凌**,女,1981 年生,博士研究生,主要研究方向为数据库与知识工程、机器学习. E-mail: zhengjiaoling@gmail.com. **唐常杰**,男,1946 年生,教授,博士生导师,主要研究领域为数据库、知识工程与数据挖掘. **徐开阔**,男,1983 年生,博士研究生,主要研究方向为机器学习与数据挖掘. **陈 瑜**,男,1974 年生,讲师,博士,研究方向为数据库与知识工程. **杨 宁**,男,1974 年生,讲师,博士研究生,主要研究方向为机器学习与数据挖掘. **段 磊**,男,1981 年生,博士,主要研究方向为数据库与知识工程.

1　研究背景和目标

1.1　混沌现象

混沌是一种确定性的非周期现象，天气、地震、传染病扩散、城市道路交通流量观测数据中都可能呈现出非周期的混沌现象[1-3]. 挖掘其中的规律富有吸引力和挑战性.

观察表明：(1)混沌虽然是非周期现象，但却是由确定性的系统产生的，其本质是确定性的，有望通过数据挖掘手段将产生混沌的确定性系统挖掘出来. (2)很多复杂系统的研究认为迭代系统是描述混沌现象的有力工具. 例如，今年的天气会影响明年的天气(见 8.2 节的实验部分)，当前已经被传染的人口数量会影响下一时间被传染的人口数，某城市道路当前的交通流会影响下一时刻的交通流. Sarkovskii 定理和 Li 定理[4]都指出，对一维迭代映射动力系统，只要有周期 3 就有混沌. 迭代系统在不同的参数条件影响下，会逐渐从稳定的周期状态演变成非周期的混沌状态. 例如某些传染病的扩散在不同条件下会出现爆发、缓慢扩散或逐渐消亡的不同状态. 又如著名的用于预测生物种群数量的 logistic 函数 $x(t)=r\times x(t-1)\times(1-x(t-1))$，在 r 取不同值时，$x(t)$会出现从分岔到混沌的过程，如图 1.



(a) $r=2.6, x(0)=0.1$　(b) $r=3.3, x(0)=0.1$

(c) $r=3.6, x(0)=0.1$　(d) $r=3.9, x(0)=0.1$

图 1　logistic 序列的分叉混沌现象

1.2　挖掘目标

第 1.1 节中对 *logistic* 函数的分析表明，混沌迭代函数系统包括为两部分：(1)确定的迭代函数；(2)函数的调节参数，该参数可使函数从稳定变成混沌. 本文将分规律挖掘和参数挖掘两部分建立模型. 下面给出形式化的挖掘目标.

(1)一阶 n 元的耦合迭代序列. 设有一组时间序列数据$\{S_1,S_2,\cdots,S_n\}$，$S_i=\{x_i(0),x_i(1),\cdots,x_i(T)\}(i\in[1,n])$，如果任意 $x_i(t)(t\in[0,T])$由$\{x_1(t-1),x_2(t-1),\cdots,x_n(t-1)\}$决定，则称$\{S_1,S_2,\cdots,S_n\}$为一阶 n 元耦合迭代序列.

如气象中温度时间序列，前一年的温度影响下一年的温度，这是以年为迭代时间单位的一阶一元耦合迭代序列(见 8.2 节的实验部分).

(2)一阶 n 元 m 参量的耦合迭代函数 $F^{n,m}$. $F^{n,m}$是形如式(1)具有 n 个耦合变元$\{x_1(t),x_2(t),\cdots,x_n(t)\}$和 m 个参量$\{r_1,r_2,\cdots,r_m\}$的迭代函数. 其中，$F^{n,m}$迭代运算 n 次可以产生一阶 n 元耦合迭代序列.

$$\begin{cases}x_1(t+1)=f_1(x_1(t),x_2(t),\cdots,x_n(t),r_1,r_2,\cdots,r_m);\\ x_2(t+1)=f_2(x_1(t),x_2(t),\cdots,x_n(t),r_1,r_2,\cdots,r_m);\\ \cdots\\ x_n(t+1)=f_1(x_1(t),x_2(t),\cdots,x_n(t),r_1,r_2,\cdots,r_m)\end{cases}\tag{1}$$

(3)挖掘目标. 设一组待挖掘的数据是一阶 n 元耦合迭代序列$\{S_1,S_2,\cdots,S_n\}$，$S_i=\{x_i(0),x_i(1),\cdots,x_i(T)\}(i\in[1,n])$，该序列的初值为$\{x_1(0),x_2(0),\cdots,x_n(0)\}$. 挖掘目标要找出一个一阶 n 元 m 参量的耦合迭代函数 $F^{n,m}$，使得 $F^{n,m}$以初值$\{x_1(0),x_2(0),\cdots,x_n(0)\}$进行 T 次迭代后得到的一阶 n 元耦合迭代序列$\{P_1,P_2,\cdots,P_n\}$，$P_i=\{y_i(0),y_i(1),\cdots,y_i(T)\}(i\in[1,n])$，与目标序列拟合程度最高，即使得 $F^{n,m}$适应度最高($F^{n,m}$适应度在第 5 节详细给出).

综上,本文的挖掘工作具有以下特点:(a)混沌迭代序列的挖掘目标.挖掘对象是一组一阶 n 元耦合迭代序列,它们由相同的迭代规律产生,有相同初值,但由于迭代规律中条件参量的不同使这组序列逐渐呈现出收敛或混沌状态,我们的算法将通过 $F^{n,m}$ 中的函数结构与参数的变化来挖掘这些不同序列;(b)联合了函数发现和参数优化的挖掘方式.$F^{n,m}$ 的挖掘既包含了函数结构发现又包含了函数中的参数优化,二者并行.本文尝试将两种挖掘任务结合在一起,并提出用 GEP 和粒子群协同进化来进行挖掘的算法.

2 相关工作

复杂迭代系统是一种存在自反馈的非线性动力系统,系统在动态运行过程中呈现出自组织现象、涌现现象和混沌现象.目前的相关研究主要是建模,仿真分析系统运行方式.复杂系统建模主要分两类:(1)基于多智能主体 Agent 的仿真,即通过建立智能主体、主体的运行环境以及主体之间在不同环境下的交互规则来仿真系统运行过程;(2)基于复杂函数系统的仿真,抽象出系统要素,建立要素之间的函数关系.通过系统自迭代仿真,两种方法都会产生出异常复杂的数据.文献[5]通过设定图节点之间的随机连接规则,产生出了具有不同拓扑性质的复杂网络,文献[6]通过调节 *SIR* 函数模型的参数,得到了完全不同的疾病传播方式,文献[7]则研究了在不同参数条件下复杂网络传染动力方程的迭代与震荡现象.

文献[8]研究了加权 Internet 访问直径时间序列的短期及长期预测,但其主要是基于 logistic 模型通过改变模型中的参量 r 来拟合序列数据,同样,文献[9-10]也只对产生混沌序列的模型中的参量进行了挖掘,没有进行函数结构本身的挖掘.文献[11]基于信息进行多参量混沌时间序列 LS-SVR 加权预测,文献[12]基于 EOF-SVD 模型进行多元时间序列相关性研究及预测,文献[13]进行了太阳黑子的时间序列预测,但它们主要是基于传统的数据拟合方式,没从真正的迭代意义上进行挖掘.

GEP[7]融合了遗传算法(GA)和遗传编程(GP)的优势.其富有特色的染色体头和尾定义,能保证 GEP 个体在进行各种遗传操作时始终产生有效语义个体,使 GEP 进化速率比 GP 平均快 2～4 个数量级,特别是在函数挖掘中显示了强大能力.如文献[14-16]研究了 GEP 在参数优化及函数发现等方面的应用.但它们都主要是对一些不存在自反馈并且各种参数都已确定的函数进行挖掘.

粒子群算法主要被用来进行参数优化,由于现实世界的优化问题日趋复杂,文献[17]采用粒子群进行了 benchmark 函数[18]的优化,文献[19-20]提出了一些改进的粒子群优化算法,但这些算法所依赖的函数系统是已经确定的,而本文是要并行地挖掘出函数系统及其参数.

本文第 3 节给出整个算法的模型框架;第 4 节提出算法的难点;第 5 节给出适应度的计算方法;第 6 节提出 3 种不同的挖掘算法;第 7 节分析各个算法的性能;第 8 节给出在合成数据和真实数据上的实验结果.

3 算法模型框架

迭代函数系统的函数结构 f 和参数 r 都是未知的,f 和 r 的描述有本质差异.我们的算法框架分成两部分,即函数结构挖掘模块和参数挖掘模块.如例 1 所示.

例 1. 考虑挖掘 2 元 2 参量一阶迭代函数,形式如下:

$$x_1(t+1)=f_1(x_1(t),x_2(t),\cdots,x_n(t),r_1,r_2);$$
$$x_2(t+1)=f_2(x_1(t),x_2(t),\cdots,x_n(t),r_1,r_2);$$

图 2 描述了一个可能的挖掘结果,由两部分组合得到.将这种组合方式简记为 $Join(StruChrom_i, ParamParticle_j)$,其中 *StruChrom* 和 *ParamParticle* 定义如下.

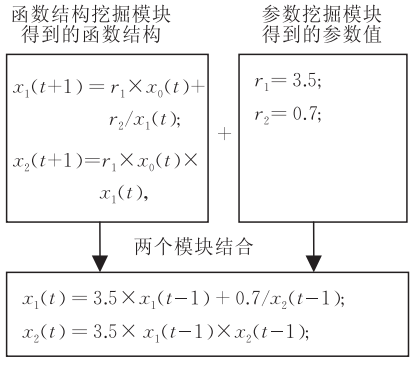


图 2 函数结构和参数挖掘模块的结合方式

3.1 基于 GEP 的函数结构挖掘模块

GEP 具有强大的函数发现能力[14-16],函数结构挖掘模块由 GEP 种群构成,种群中个体代表了一种完整的函数结构,用于挖掘函数结构的 GEP 个体定义如下.

定义 1(函数结构染色体 *StruChrom*). 挖掘

函数结构的 GEP 个体是多基因单染色体类型的 GEP 个体，简记为 $StruChrom$，是一个四元组 $\langle n, N, T, h\rangle$，其中：(1) n 是染色体包含的基因个数，n 为待挖掘函数系统耦合变元个数；(2) T 是基因的终结符集合 $T=\{x_1, x_2, \cdots, x_n, r_1, r_2, \cdots, r_m\}$，$x_1, x_2, \cdots, x_n$ 是耦合变元，$r_1, r_2, \cdots, r_m$ 是参数；(3) N 是基因的非终结符集合，N 可以是任意合法的函数运算符；(4) h 是基因的头部长度.

例 2.　设要挖掘例 1 中的函数系统，则 $StruChrom$ 所对应的四元组为 $\langle 2, \{x_1, x_2, r_1, r_2\}, \{+, -, \times, /\}, 3\rangle$，设由该四元组得到的一个 $StruChrom$ 为“$+\times x_2 r_2 x_1 x_2 x_1 \times + \times r_1 x_2 r_2 x_1$”，则其对应 K 表达式和相应的函数结构如图 3.

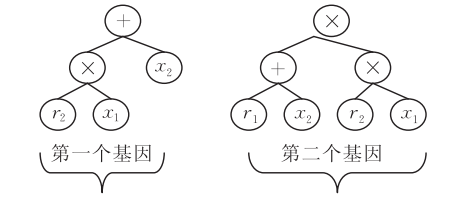


图 3　函数结构模块

3.2　基于粒子群的参数挖掘模块

粒子群算法在参数优化方面具有强大的能力[17-20]，参数挖掘模块采用粒子群技术实现，粒子群中每个粒子代表一组完整的参数值，其定义如下.

定义 2(函数参量粒子 $ParamParticle$).　挖掘参数的粒子群粒子是一个长为 m 的一维实数向量，$ParamParticle=\langle r_1, r_2, \cdots, r_m\rangle$，$r_i \in R$，($i \in [1, m]$)对应第 i 个参数.

例 3(参数粒子).　考虑从合成数据中挖掘 logistic 函数，由于其只含有 1 个参数，故 $ParamParticle$ 是一维向量，可能的结果形如 $ParamParticle=\langle 0.9\rangle$ 或 $ParamParticle=\langle 3.6\rangle$.

4　混沌迭代序列挖掘的难点

由于混沌序列的值在参数发生很小变化的情况下会大相径庭(如图 1). 所以采用异构种群分别挖掘拟合混沌迭代序列的函数结构及其参量存在以下难点：(1) GEP 种群正确地挖掘出了最优的函数结构，但粒子群没有挖掘出其对应的参量，则无法得到最优结果. (2) 即使两个种群都挖掘出了最优的函数结构和参数，但没能将二者结合在一起，也无法得到最优解. 所以如何结合不同种群，使得它们的挖掘结果组合在一起能达到最优拟合效果是挖掘的难点. 考虑下列极端情况.

例 4(种群的结合方式对挖掘混沌序列的影响).　设合成数据由例 1 中的 logostic 函数 $x(t)=3.6\times x(t-1)\times(1-x(t-1))$ 生成，此时迭代序列出现了较多的分岔情况. 设 GEP 种群和粒子群中的两个个体如图 4 所示. 如果 $Join(StruChrom_1, ParamParticle_2)$，则立刻得到最优解. 但如果 $Join(chromosome_1, ParamParticle_1)$，则会错过最优解，故种群的结合方式对种群的进化方向存在很大影响. 下面对寻找最优种群结合方式的难度进行了分析.

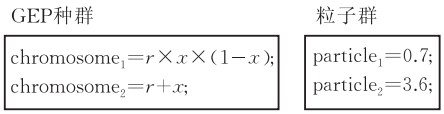


图 4　种群的结合方式

引理 2 证明了协同多种群寻找最优解的进化难度将会远大于单个种群的进化难度，为了证明的清晰性，首先给出证明中用到符号的含义和一个假设.

(1) 称 GEP 种群在 t 时刻包含的所有个体为 GEP 种群在 t 时刻的状态，记为 $GEP\_Pop\_Status(t)=\{StruChrom_1(t), StruChrom_2(t), \cdots, StruChrom_n(t)\}$；

(2) 称粒子群在 t 时刻包含的所有个体为粒子群在 t 时刻的状态，记为 $PSO\_Pop\_Status(t)=\{ParamParticl\,e_1(t), ParamParticl\,e_2(t), \cdots, ParamParticle_m(t)\}$；

(3) 称 GEP 种群和粒子群在 t 时刻结合后包含的所有个体为联合种群在 t 时刻的状态，记为 $Unit\_Pop\_Status(t)=(StruChrom_i(t)\bigcup ParamParticl\,e_j(t))(j\in[1,m], i\in[1,n])$.

(4) 种群所有可能状态组成的集合称为种群状态空间.

为了构造简单而有效的模型我们需要下列的假设.

假设 1.　设 $GEP\_Pop\_Status(t)$ 和 $PSO\_Pop\_Status(t)$ 随时间产生的状态序列是马尔可夫链.

说明：由于 $GEP\_Pop\_Status(t)$ 和 $PSO\_Pop\_Status(t)$ 均是种群进化过程，而种群进化的状态序列在大量文献中[21]已被证明是马尔可夫链，所以这里假设是合理的.

引理 1.　$Unit\_Pop\_Status(t)$ 的状态序列是马尔可夫链.

证明.　由于 $GEP\_Pop\_Status(t)$ 和 $PSO\_Pop\_Status(t)$ 序列是马尔可夫链，所以任意

$StruChrom_i(t)$ 和 $ParamParticle_j(t)$ 只与 $StruChrom_i(t-1)$ 和 $ParamParticle_j(t-1)$ 相关，又因为 $Unit\_Pop\_Status(t)=(StruChrom_i(t)\cup Paramparticle_j(t))$，故 $Unit\_Pop\_Status(t)$ 只与 $Unit\_Pop\_Status(t-1)$ 相关，所以 $Unit\_Pop\_Status(t)$ 的状态序列是马尔可夫链.

引理 2. 设 GEP 种群和粒子群的状态空间大小分别为 N 和 M，种群大小分别为 n 和 m，联合种群 $Unit\_Pop\_Status$ 的种群大小为 h，则 $\boldsymbol{U}(t)$ 的状态空间大小为 $N\times M\times C_{m\times n}^{h}$.

证明. 由于 $Unit\_Pop\_Status(t)$ 的状态空间包含了 $GEP\_Pop\_Status(t_1)$ 和 $PSO\_Pop\_Status(t_2)$ (t_1，t_2 为任意合法时间)的任意 $N\times M$ 种组合，而在每种组合中又包含两个种群所有个体的任意 $C_{m\times n}^{h}$ 种组合，故为 $M\times N\times C_{m\times n}^{h}$.

从引理 2 可以看出，联合种群的状态空间大小远大于 GEP 种群和粒子群状态空间的大小，故联合种群的进化难度将会远大于单个种群的进化难度.

由于如何计算个体适应度实际上决定了种群的结合方式，下面给出了基于个体适应度的种群结合方式.

定义 3(种群的结合方式). 是一个四元组〈G，P，F_G，F_P〉，

(1) G 是 GEP 种群，$G=\{StruChrom_1,\cdots,StruChrom_n\}$；

(2) P 是粒子群，$P=\{ParamParticle_1,\cdots,ParamParticle_m\}$；

(3) F_G 是 G 中每个个体适应度的计算方式；

(4) F_P 是 P 中每个个体适应度的计算方式.

5 耦合迭代函数适应度的计算

本节给出耦合迭代函数适应度的计算方法，而异构种群个体的适应度将在此基础上得到. 通过例 5 来说明本文适应度计算方式与传统的适应度计算方式的区别. 具体计算方式见定义 4.

例 5. 设挖掘的目标序列为{0，1，2，3}，设挖掘出的迭代函数为 $x(t)=2\times x(t-1)$. 传统的方法是将 $f(t-1)$ 的值代入 g 中进行计算的，即 $g(t)=g(f(t-1))$，此时得到的迭代序列为{0，0，2，4}. 如文献[11-13]等都是采用的这种方法，但这显然是不合理的，因为迭代系统应当输入自身在上一时刻计算出的结果，即 $g(t)=g(g(t-1))$，此时得到的迭代序列示为{0，0，0，0}，而这正是本文所采用的计算适应度的方法.

定义 4(耦合迭代函数 $F^{n,m}$ 的适应度 $Fitness(F^{n,m})$). 设 $\{S_1,S_2,\cdots,S_n\}$，$S_i=\{x_i(0),x_i(1),\cdots,x_i(T)\}$ ($i\in[1,n]$) 是待挖掘的一阶 n 元耦合迭代序列，初值为 $\{x_1(0),x_2(0),\cdots,x_n(0)\}$. $F^{n,m}$ 是挖掘出的一阶 n 元 m 参量的耦合迭代函数，$F^{n,m}$ 在初值为 $\{x_1(0),x_2(0),\cdots,x_n(0)\}$ 的条件下通过 T 次迭代运算得到的一阶 n 元耦合迭代序列 $\{P_1,P_2,\cdots,P_n\}$，$P_i=\{y_i(0),y_i(1),\cdots,y_i(T)\}$ ($i\in[1,n]$)，则 $F^{n,m}$ 的适应度为

$$Fitness(F^{n,m})=\sum_{i=1}^{n}\sum_{t=0}^{T}H-\left|\frac{(y_i(t)-x_i(t))}{x_i(t)}\right|\times 100,$$

H 是适应度规范因子，T 是迭代次数. 表 1 给出了下文算法中需要用到的符号.

表 1 符号表

符号	含义
$GEP\_Pop(t)$	GEP 第 t 代种群
$Fitness(StruChrom_i(t))$	GEP 种群第 i 个个体在第 t 代的适应度
$Fitness(ParamParticle_j(t))$	粒子群第 j 个个体在第 t 代的适应度
$Best\_StruChrom(t)$	GEP 种群第 t 代最优个体
$PSO\_Pop(t)$	粒子群第 t 代种群
$Global\_Best\_ParamParticle(t)$	粒子群第 t 代全局最优个体
$Local\_Best\_ParamParticle_j(t)$	$ParamParticle_j(t)$ 的局部最优个体
$Global\_Best\_ParamParticle\_Fitness(t)$	粒子群在第 t 代全局最优个体的适应度，初值为 0
$Local\_Best\_\ ParamParticle_j\_Fitness(t)$	$ParamParticle_j(t)$ 的局部最优个体的适应度，初值为 0
x_{jp}	$ParamParticle_j$ 的第 p 维分量
v_{jp}	x_{jp} 的速度，初值为 1
w	惯性常量
c_1	全局最优分量的加速常量

6 协同进化的三种结合策略

如前所述，由于混沌迭代序列的演化方式对参量非常敏感，所以如何使 GEP 种群(粒子群)能够敏感地发现粒子群(GEP 种群)中的优质个体并与其结合，是协同进化的核心，我们提出了 3 种结合算法，具体如下.

6.1　朴素交替式种群结合策略

我们首先提出一种朴素交替式的种群结合策略来进行两个种群的联合挖掘，基本思想如下：(1) 选出粒子群当前代的最优个体 *Global_Best_ParamParticle*(*t*)，让 GEP 种群中所有 *StruChrom*(*t*)都与该个体结合并进化，得到当前最优的 *StruChrom*. (2) 选出 GEP 种群当前代的最优个体 *Best_StruChrom*(*t*)，让粒子群中所有 *ParamParticle*(*t*)与该最优个体结合并进化. (3) 循环地执行这个过程. 整个算法的基本框架如图 5 所示，具体运行流程如图 6 所示.

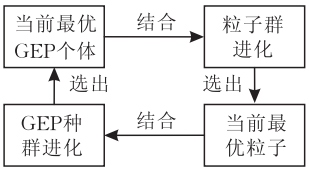


图 5　朴素交替式种群结合策略

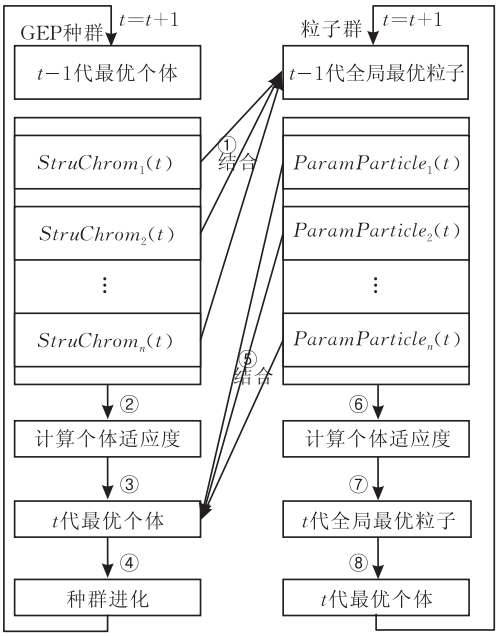


图 6　朴素交替式种群结合策略

(1) 种群结合方式：$\langle G, P, F_G, F_P \rangle$(见定义 3).

1) G, $G=\{StruChrom_1(t), StruChrom_2(t), \cdots, StruChrom_n(t)\}$;

2) P, $P=\{ParamParticle_1(t), ParamParticle_2(t), \cdots, ParamParticle_m(t)\}$;

3) F_G: $Fitness(StruChrom_i(t))=Fitness(Join(StruChrom_i(t), Global\_Best\_ParamParticle(t-1)))$;

4) F_P: $Fitness(ParamParticle_j(t))=Fitness(Join(Best\_StruChrom(t), ParamParticle_j(t)))$.

(2) 种群进化算法

如图 6 所示，箭头上的序号是种群的进化步骤.

步骤①. GEP 种群中的所有个体与粒子群中当前最优全局粒子结合，如 F_G 所示.

步骤②. 计算每个 *StruChrom*(*t*)的适应度.

步骤③. 得到种群当前最优个体 *Best_StruChrom* (*t*).

步骤④. GEP 种群进行新一轮进化.

步骤⑤. 粒子群中的所有个体与 GEP 种群中当前最优个体结合，如 F_P 所示.

步骤⑥. 计算每个 *ParamParticle*(*t*)的适应度.

步骤⑦. 得到当前最优全局粒子 *Global_Best_ParamParticle*(*t*).

步骤⑧. 粒子群进行新一轮进化.

具体算法如下，由于篇幅原因，GEP 种群的进化方式参见文献[13]，GEP 种群和粒子群当前代全局最优个体的生成算法、粒子群进化算法及适应度计算方法见附录.

算法 1.　基于朴素交替式种群结合策略的挖掘算法(Naïve alternating cooperating strategy)

输入：初始 *GEP_Pop*(0), *PSO_Pop*(0), *Best_StruChrom*(0), *Global_Best_ParamParticle*(0)

输出：满足停止条件的种群最优个体 *Best_StruChrom*(*t*), *Global_Best_ParamParticle*(*t*)

```
1.  t=0;
2.  While(stop_condition)
3.    For each StruChrom_i(t) in GEP_Pop(t);
4.      f=Fitness(StruChrom_i(t))=Fitness
          (Join(StruChrom_i(t),
           Global_Best_ParamParticle(t−1)));
5.        Generate_Best_StruChrom
                          (StruChrom_i(t),f);
6.    End For
7.    Evolute_GEP_Population(GEP_Pop(t));
8.    For each ParamParticle_j(t) in PSO_Pop(t);
9.      f=Fitness(ParamParticle_j(t))=Fitness(Join
                (Best_StruChrom(t), ParamParti-
                cle_j(t));
10.       Generate_Best_Global_ParamParticle
                (ParamParticle_j(t),f);
11.   End For
12.   Evolute_ PSO_Population(PSO_Pop(t));
13.   t=t+1;
14. End While
15. return Best_StruChrom(t);
16. return Global_Best_ParamParticle(t);
```

其中，第 4,9 行描述了种群的结合方式，第 5,10 行描述了如何生成 GEP 种群和粒子群的最优个体，第 7,12 行描述种群的进化方式，具体算法见附录.

6.2 单路并行式种群结合策略

在交替式策略中,GEP 种群中所有个体都只与一个 *ParamParticle* 结合,同样粒子群中所有个体都只与一个 *StruChrom* 结合,这会使种群很快收敛到局部最优,故提出了下列改进算法:(1)对 GEP 种群中的任意个体 $StruChrom_i(t)$,将 $ParamParticle_i(t)$ 与其进行结合,用 $ParamParticle_i(t)$ 来指导 $StruChrom_i(t)$ 的进化;(2)对粒子群中的任意个体 $ParamParticle_i(t)$,将 $StruChrom_i(t)$ 与其进行结合,用 $StruChrom_i(t)$ 来指导 $ParamParticle_i(t)$ 的进化.这样就使得种群的结合方式从一种变成了 n 种(n 为种群的大小).

具体结合方式如下.

(1)种群结合方式:$\langle G,P,F_G,F_P\rangle$(见定义 3),

1) G,$G=\{StruChrom_1(t),StruChrom_2(t),\cdots,StruChrom_n(t)\}$.

2) P,$P=\{ParamParticle_1(t),ParamParticle_2(t),\cdots,ParamParticle_m(t)\}$.

3) F_G:$Fitness(StruChrom_i(t))=Fitness(Join(StruChrom_i(t),ParamParticle_i(t)))$.

4) F_P:$Fitness(ParamParticle_j(t))=Fitness(Join(StruChrom_j(t),ParamParticle_j(t)))$.

如图 7,将对应位置的 *StruChrom* 与 *ParamParticle* 相结合,两个种群的大小应相同.值得指出的是,当 GEP 种群第 j 个位置的个体 $StruChrom_j$ 在进化中被变异或者被淘汰,而由另一个个体占据第 j 个位置,此时粒子群第 j 个粒子 $partcile_j$ 仍然与 GEP 种群的第 j 个个体结合.

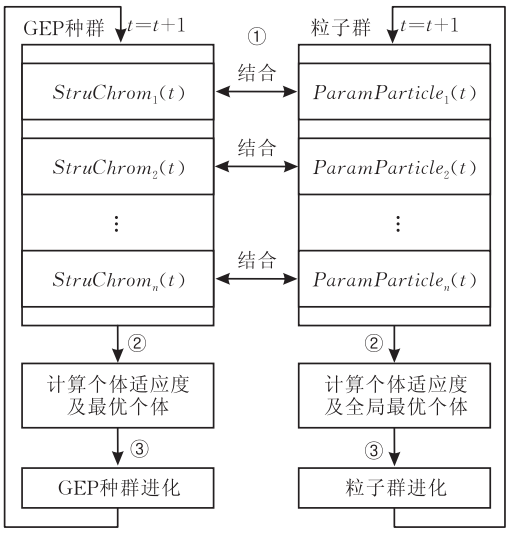


图 7 单路并行式种群结合策略

(2)种群进化算法

如图 7 所示,若两个箭头上序号相同,表示两个过程可以并行进行.

步骤①. 表示两个种群中的个体按照 F_G,F_P 的方式结合;

左步骤②:表示 GEP 种群与粒子群结合后计算每个 *StruChrom* 的适应度和 *GEP* 种群的当前最优个体 $Best\_StruChrom(t)$.

右步骤②:表示粒子群与 GEP 种群结合后计算每个 *ParamParticle* 的适应度和粒子群的当前全局最优个体 $Global\_Best\_ParamParticle(t)$.

左步骤③:表示 GEP 种群按照进化规则进行进化.

右步骤③:表示粒子群按照进化规则进行进化.

算法 2. 基于单路并行式种群结合策略的挖掘算法(Single way paralleling cooperating strategy)

输入:初始 $GEP\_Pop(0)$,$PSO\_Pop(0)$,$Best\_StruChrom(0)$,$Global\_Best\_ParamParticle(0)$

输出:满足停止条件的种群最优个体 $Best\_StruChrom(t)$,$Global\_Best\_ParamParticle(t)$

```
1. t=0;
2. While(stop_condition)
3.    For each StruChrom_i(t) in GEP_Pop(t) and
          ParamParticle_i(t) in PSO_Pop(t)
4.       f=Fitness(StruChrom_i(t))=Fitness(Param
          Particle_i(t))=Fitness(Join(StruChrom_i(t),
          ParamParticle_i(t)));
5.          Generate_Best_StruChrom
              (StruChrom_i(t),f);
6.          Generate_Best_Global_ParamParticle
              (ParamParticle_i(t),f);
7.    End For
8.    Evolute_ GEP_Population(GEP_Pop(t));
9.    Evolute_ PSO_Population(PSO_Pop(t));
10.   t=t+1;
11. End While
12. return Best_StruChrom(t);
13. return Global_Best_ParamParticle(t);
```

第 3~7 行描述了种群的结合方式,第 5,6 行描述了如何生成 GEP 种群和粒子群的最优个体,第 8,9 行描述种群的进化方式.同样由于篇幅原因,GEP 种群的进化方式参见文献[13],GEP 种群和粒子群当前代全局最优个体的生成算法、粒子群进化算法及适应度计算方法见附录.

6.3 多路并行式种群结合策略

虽然相对于朴素结合策略,单路并行策略将种群的结合可能性从一种提高到了 n 种(n 为种群大小),但还可能错过很多好的结合机会,例如考虑例 4 中挖掘 logistic 函数的例子.显然 $StruChrom_1$ 与 $ParamParticle_2$ 结合会得到最优解,但单路并行策

略很可能会错过这种结合方式.

基于上述考虑，提出了如下解决思路：(1) 对 GEP 种群中的任意个体 $StruChrom_i(t)$，首先让 $StruChrom_i(t)$ 与粒子群中的所有粒子分别进行结合，此时 $StruChrom_i(t)$ 会得到多个适应度，取最高的一个作为 $StruChrom_i(t)$ 在当前的适应度. (2) 对粒子群任意个体 $ParamParticle_j(t)$，采取相同策略计算 $ParamParticle_j(t)$ 当前的适应度. 如图 8 所示. 首先给出种群的结合方式.

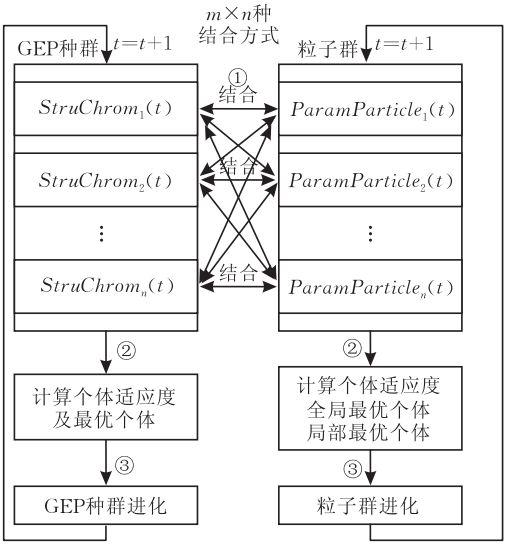


图 8　多路并行式种群结合策略

(1) 种群的结合方式：$\langle G, P, F_G, F_P \rangle$（见定义 3），

1) G，$G=\{StruChrom_1(t), StruChrom_2(t), \cdots, StruChrom_n(t)\}$.

2) P，$P=\{ParamParticle_1(t), ParamParticle_2(t), \cdots, ParamParticle_m(t)\}$.

3) F_G：$Fitness(StruChrom_i(t))=Fitness(Join(StruChrom_i(t), ParamParticle_k(t)))$，当对任意 $ParamParticle_{k'}(t) \in P$，都有 $Fitness(Join(StruChrom_i(t), ParamParticle_k(t))) \geq Fitness(Join(chromosome_i(t), ParamParticle_{k'}(t)))$.

(4) F_P：$Fitness(ParamParticle_j(t))=Fitness(Join(StruChrom_q(t), ParamParticle_j(t)))$，当对任意 $ParamParticle_{q'}(t) \in P$，都有 $Fitness(Join(StruChrom_q(t), ParamParticle_j(t))) \geq Fitness(Join(chromosome_{q'}(t), ParamParticle_j(t)))$.

(2) 种群进化算法

步骤①：两个种群中的个体按照 F_G，F_P 的方式结合；

左步骤②：GEP 种群与粒子群结合后计算每个 $StruChrom$ 的适应度和 GEP 种群的当前最优个体 $Best\_StruChrom(t)$.

右步骤②：粒子群与 GEP 种群结合后计算每个 $ParamParticle$ 的适应度和粒子群的当前全局最优 $Global\_Best\_ParamParticle(t)$ 和每个粒子 $ParamParticle_j$ 的局部最优 $Local\_Best\_ParamParticle_j(t)$.

左步骤③：GEP 种群按照进化规则进行进化.

右步骤③：粒子群按照进化规则进行进化.

算法 3.　基于多路并行式种群结合策略的挖掘算法（Multi way paralleling cooperating strategy）

输入：初始 $GEP\_Pop(0)$，$PSO\_Pop(0)$，$Best\_StruChrom(0)$，$Global\_Best\_ParamParticle(0)$

输出：满足停止条件的种群最优个体 $Best\_StruChrom(t)$，$Global\_Best\_ParamParticle(t)$

1. $t=0$;
2. While($stop\_condition$)
3.　For each $StruChrom_i(t)$ in $GEP\_Pop(t)$
4.　 For each $ParamParticle_j(t)$ in $PSO\_Pop(t)$
5.　　$f=Fitness(StruChrom_i(t))=Fitness(Join(StruChrom_i(t), ParamParticle_j(t)))$;
6.　　$Generate\_Best\_StruChrom(StruChrom_i(t), f)$;
7.　　$Generate\_Best\_Global\_ParamParticle(ParamParticle_j(t), f)$;
8.　 End For
9.　End For
10.　$Evolute\_GEP\_Population(GEP\_Pop(t))$;
11.　$Evolute\_PSO\_Population(PSO\_Pop(t))$;
12.　$t=t+1$;
13. End While
14. return $Best\_StruChrom(t)$;
15. return $Global\_Best\_ParamParticle(t)$;

3～8 行描述了种群结合策略，第 6，7 行描述了生成 GEP 种群和粒子群的最优个体的方式，GEP 种群进化方式见文献[13]，GEP 种群和粒子群当前代全局最优个体的生成算法和粒子群进化算法及适应度计算方法见附录.

7　算法的有效性分析

7.1　算法 3 与算法 1，2 的比较

本节将从马尔可夫链的角度分析算法 1～3 找到最优解的可能性. 通过定理 1 证明算法 3 优于算法 1，2.

为了证明的简洁、清晰，引入下列符号和术语.

(1) 联合种群的最优个体记为 $BestInd=\{StruChrom_i(t) \cup ParamParticle_j(t)\}$，其中

$StruChrom_{ii}(t)$和 $ParamParticle_j(t)$是所有结合后能够达到迭代函数系统最优适应度阈值的个体.

(2)最优种群 $BestPop(t)$和非最优种群 $NbestPop(t)$.若种群中存在 $BestInd$,则称该种群为最优种群,记为 $BestPop(t)$,否则为非最优种群,记为 $NbestPop(t)$.

(3)完全状态概率转移矩阵 $\boldsymbol{M}$.$\boldsymbol{M}$ 是 $Unit_Pop_Status(t)$序列所构成的马尔可夫链的状态转移概率矩阵,$\boldsymbol{M}$ 中每个状态是任意 $Unit_Pop_Status(t)$对应的状态.

(4)规约状态概率转移矩阵 $Reduced\_M$. $Reduced\_M$ 含两个状态 B 和 N,$B=\bigcup\{BestPop(t)\}$,$N=\bigcup\{NbestPop(t)\}$,P_{BB},P_{BN},P_{NB},P_{NN} 是状态间转移概率.

注意,$Reduced\_M$ 实际上是将 M 中的所有状态归纳成两种状态后得到的状态转移矩阵,由于 M 是马尔可夫链,所以 $Reduced\_M$ 仍然是马尔可夫链.下面将对算法 1,2,3 中所对应的 $Reduced\_M$ 的转移概率的大小关系进行假设.

假设 1. 设 $Reduced\_M_1$,$Reduced\_M_2$,$Reduced_M_3$和 P^1,P^2,P^3分别是算法 1,2,3 各自的种群序列进化过程中得到的规约状态概率转移矩阵和概率,则有 $P^3_{NB}>P^1_{NB}$,$P^3_{NB}>P^2_{NB}$,$P^3_{BN}<P^1_{BN}$,$P^3_{BN}<P^2_{BN}$.

解释:算法 1 中 $U_i(t)=StruChrom_i(t)\bigcup ParamParticle_i(t)$,算法 2 中 $U_i(t)=StruChrom_i(t)\bigcup ParamParticle_B(t)$,算法 3 中 $U_i(t)=StruChrom_B(t)\bigcup ParamParticle_B(t)$,其中 $StruChrom_B(t)$和 $ParamParticle_B(t)$分别是当前代中 F_module 种群和 P_module 种群的最优个体,从三种联合方式易知采用 $StruChrom_B(t)\bigcup ParamParticle_B(t)$的方式更容易发展出更优秀的个体,而 P_{NB} 和 P_{BN} 则是这种趋势的概率表达,故假设是合理的.

由于 $P^3_{BN}+P^3_{BB}=1$,$P^2_{BN}+P^2_{BB}=1$,$P^1_{BN}+P^1_{BB}=1$,又根据假设 2,即 $P^3_{BN}<P^1_{BN}$,$P^3_{BN}<P^2_{BN}$,故有 $P^3_{BB}>P^1_{BB}$,$P^3_{BB}>P^2_{BB}$.

定理 1. 设算法 1,2,3 初始种群大小均为 N,包含最优个体数均为 $N_B(1)$,设 $N_1(t)$,$N_2(t)$,$N_3(t)$,是 3 个种群 t 代后包含最优个体数量的期望值,则 $N_3(t)>N_2(t)$,$N_3(t)>N_1(t)$.

证明. (1)三种算法第 2 代种群包含最优个体数量分别为 $N^i_B(2)=(N-N_B(1))\times P^i_{NB}+N_B(1)\times(P^i_{BB}-P^i_{BN})(i=1,2,3)$,上式很容易按照概率转移矩阵得到,又由假设 2 及 $P^3_{BB}>P^1_{BB}$,$P^3_{BB}>P^2_{BB}$,易知 $N^3_B(2)>N^1_B(2)$,$N^3_B(2)>N^2_B(2)$.(2)由于在第二代由算法 3 得到的最优个体数期望量已经领先,故在 2 代以后,由算法 3 得到最优个体数的期望值仍将最大.故概率意义上算法 3 的性能要优于算法 1 和 2.

7.2 算法 3 与基于单种群 GEP 挖掘算法的比较

基于单种群的 GEP 挖掘算法(简记为算法 4)指用 B-GEP[13] 来挖掘函数模块,用 GEP-PO[13] 来挖掘参数,参见文献[13].本节旨在证明算法 3 找到最优解的概率大于算法 2.首先给出假设 2.

假设 2. 设 $P^3_{NB}>P^4_{NB}$,$P^3_{BN}<P^4_{BN}$,P^3_{NB},其中 P^3_{NB},P^4_{NB}以及 P^3_{BN},P^4_{BN}的含义与假设 1 相同.

解释:(1)算法 3 与算法 4 的区别在参数发现模块,算法 3 用 PSO 进行参数发现,而算法 4 用 GEP 进行参数发现.文献[17]和[14]分别用 GEP 和 PSO 对 CEC2005[22] 提出的 12 个 benchmark 函数($F_1\sim F_{12}$)进行了优化测试.对比两个文献的结果可知,PSO 的参数优化能力远优于 GEP.(2)由于混沌迭代函数对参数非常敏感,当挖掘出的参数与最优值距离稍远,会导致整个挖掘结果的适应度急剧下降.而 PSO 的参数微调能力远强于 GEP,所以基于以上分析,有假设 2.

定理 2. $N_3(t)>N_4(t)$,$N_3(t)$,$N_4(t)$的含义与定理 1 相同.

证明. 整个证明过程与定理 1 的证明相似,这里省略.

故概率意义上算法 3 的性能要优于算法 4.

通过 7.1 和 7.2 节的分析,证明了算法 3 无论是相对于单纯的 GEP 挖掘算法(算法 4),还是对于其它不同策略的异质种群协同进化算法(算法 1,2),都更为有效.

8 实验和性能分析

8.1 合成数据上的实验

(1)实验数据

构造 2 组共 6 个函数 $F_1\sim F_6$,结构和参数取值如表 2 所示.①混沌迭代序列:$F_1\sim F_3$ 含有一个变量 $x(t)$和一个参数 r,$x(0)=0.1$,r 从 2.6 变化到 3.9 时,$x(t)$从稳定走向混沌,$F_4\sim F_6$ 含有两个耦合变量 $x(t)$,$y(t)$和一个参数 r,$x(0)=0.1$,$y(0)=0.2$,r 从 3.9 变化到 5.5 时,$x(t)$,$y(t)$从稳定走向混沌.②训练数据:$F_1\sim F_3$ 中,取 $x(0)\sim x(19)$作为训练数据.$F_4\sim F_6$ 中取 $x(0)\sim x(19)$,$y(0)\sim y(19)$

作为训练数据. ③ 测试数据：$F_1 \sim F_3$ 中，取 $x(20) \sim x(99)$ 作为测试数据. $F_4 \sim F_6$ 中取 $x(20) \sim x(99)$，$y(20) \sim y(99)$ 作为测试数据.

表 2　合成数据的函数及参数

	函数结构	参数
F_1	$x(t+1)=r\times x(t)\times(1-x(t))$	2.6
F_2	$x(t+1)=r\times x(t)\times(1-x(t))$	3.3
F_3	$x(t+1)=r\times x(t)\times(1-x(t))$	3.9
F_4	$x(t+1)=r\times(x(t)/y(t))\times(1-x(t)/y(t))$ $y(t)=1+x(t)/y(t)$	3.9
F_5	$x(t+1)=r\times(x(t)/y(t))\times(1-x(t)/y(t))$ $y(t)=1+x(t)/y(t)$	5.2
F_6	$x(t+1)=r\times(x(t)/y(t))\times(1-x(t)/y(t))$ $y(t)=1+x(t)/y(t)$	5.5

(2) 算法和参数

表 2 列出了实验用到的算法，所有算法在 CPU 为 Intel 1.6GHz，512MB 内存，Windows XP 的 PC 机上完成. 所有算法在 zGEP[23] 基础上进行扩展，用 Java 实现.

表 3 列出了各个算法使用的参数，都是经过大量实验后的最优值. 其中，A_1 表示基于单路并行式种群结合策略的挖掘算法；A_2 表示基于朴素交替式种群结合策略的挖掘算法；A_3 表示基于多路并行式种群结合策略的挖掘算法；A_4 表示纯 GEP 挖掘算法，即用 B-GEP[13] 来挖掘函数模块，用 GEP-PO[13] 来挖掘参数.

表 3　参数表

(a) GEP 及 PSO 在 $F_1 \sim F_6$，$A_1 \sim A_4$ 中相同的参数设置

参数	值	参数	值
GEP 及 PSO 种群大小	100	GEP 插串率	0.1
种群进化代数	100000	GEP 根插串率	0.1
GEP 变异率	0.044	GEP 基因插串率	0.1
PSO 中 v_{jp}(粒子群个体速度)	1	GEP 选择算子	锦标赛
PSO 中 w(个体的惯性常量)	0.9	PSO 中 c_2(全局最优分量的加速常量)	2

(b) GEP 在 $F_1 \sim F_6$，$T_1 \sim T_6$，$A_1 \sim A_4$ 中有差别的参数设置

参数	值		
	$F_1 \sim F_3$，$A_1 \sim A_4$	$F_4 \sim F_6$，$A_1 \sim A_4$	$T_1 \sim T_6$，$A_1 \sim A_4$
终止符集	{a,b}	{a,b,c}	{a,b}
函数集	{+,−,×,/}	{+,−,×,/}	{+,−,×,/,sin,cos,tan,exp}
头长	6	9	9
基因个数	1	2	1
染色体长度	13	19	19
单点重组率	0.3	0.5	0.3
双点重组率	0.3	0.5	0.3
基因重组率	0.1	0.7	—
最优适应度	2000	4000	2000

(3) 实验结果

① 序列拟合情况. 表 4，5 是 A_1 和 A_3 对 $F_4 \sim F_6$ 在 10 次实验中的最优挖掘结果. 表 6 是表 4，5 中挖掘出的函数迭代 100 次得到的序列与原序列的欧氏距离. 可知，A_3 拟合出的序列要优于 A_1，当然，A_3 仍然是收敛到了局部最优.

② 算法收敛情况. 图 9 列出了 $A_1 \sim A_4$ 挖掘 $F_1 \sim F_6$ 的平均实验结果(每种算法对每个函数分别运行 10 次)，横坐标是进化代数的 10 的对数值，纵坐标是适应度. 为保持图的清晰性，只画出了当算法达到某一适应度时的最先的代数. 对 $F_1 \sim F_3$，A_1 和 A_3 能在 100～1000 代的数量级上达到最优适应度，A_2，A_4 会很快地收敛到局部最优. 对 $F_4 \sim F_6$，A_1 和 A_3 能够在 1000～10000 代的数量级上达到相对较高的适应度，但适应度的绝对值会随着分岔数的增加而降低. 同样，A_2 和 A_4 出现了快速收敛到局部最优的困境.

表 4　A_1 对 $F_1 \sim F_6$ 的最优挖掘结果

	函数结构	参数	适应度
F_1	$x(t+1)=r\times x(t)\times(1-x(t))$	2.6	2000
F_2	$x(t+1)=r\times x(t)\times(1-x(t))$	3.3	2000
F_3	$x(t+1)=r\times x(t)\times(1-x(t))$	3.9	2000
F_4	$x(t+1)=(((x(t))/(x(t)))+(r+(x(t))))/((r-(x(t)))-(x(t)))$ $y(t+1)=((x(t))+(y(t)))/(y(t))$	−62.7	3959
F_5	$x(t+1)=((r/(x(t)))-((x(t))+(x(t))))/((r/(y(t)))+r)$ $y(t+1)=((y(t))+(x(t)))/(y(t))$	−9.7	3687
F_6	$x(t+1)=r/((((x(t))\times r)\times(x(t)))+(r+(y(t))))$ $y(t+1)=((y(t))+(x(t)))/(y(t))$	−5.0	3456

表 5　A_3 对 $F_1 \sim F_6$ 在 10 次实验中的最优挖掘结果

	函数结构	参数	适应度
F_1	$x(t+1)=r\times x(t)\times(1-x(t))$;	2.6	2000
F_2	$x(t+1)=r\times x(t)\times(1-x(t))$;	3.3	2000
F_3	$x(t+1)=r\times x(t)\times(1-x(t))$;	3.9	2000
F_4	$x(t+1)=((((r)+(y(t)))-((x(t))-(y(t))))-((x(t))+(y(t))))/(r)$; $y(t+1)=((y(t))+(x(t)))/(y(t))$;	−62.7	3986
F_5	$x(t+1)=(y(t))/((y(t))+((((x(t))+r)+((x(t))\times(x(t))))\times((x(t))\times(x(t)))))$; $y(t+1)=((y(t))+(x(t)))/(y(t))$;	−9.7	3726
F_6	$x(t+1)=((y(t))+(y(t)))/(((x(t))\times((x(t))\times(y(t))))+((y(t))+(x(t))))$; $y(t+1)=((y(t))+((x(t))\times r))/(y(t))$;	−5.0	3500

表 6　挖掘出的迭代序列与原序列的欧式距离

	F_4		F_5		F_6	
	x	y	x	y	x	y
A_1	0.08	0.07	2.24	1.88	7.58	5.73
A_3	0.05	0.02	1.51	0.78	3.52	2.70

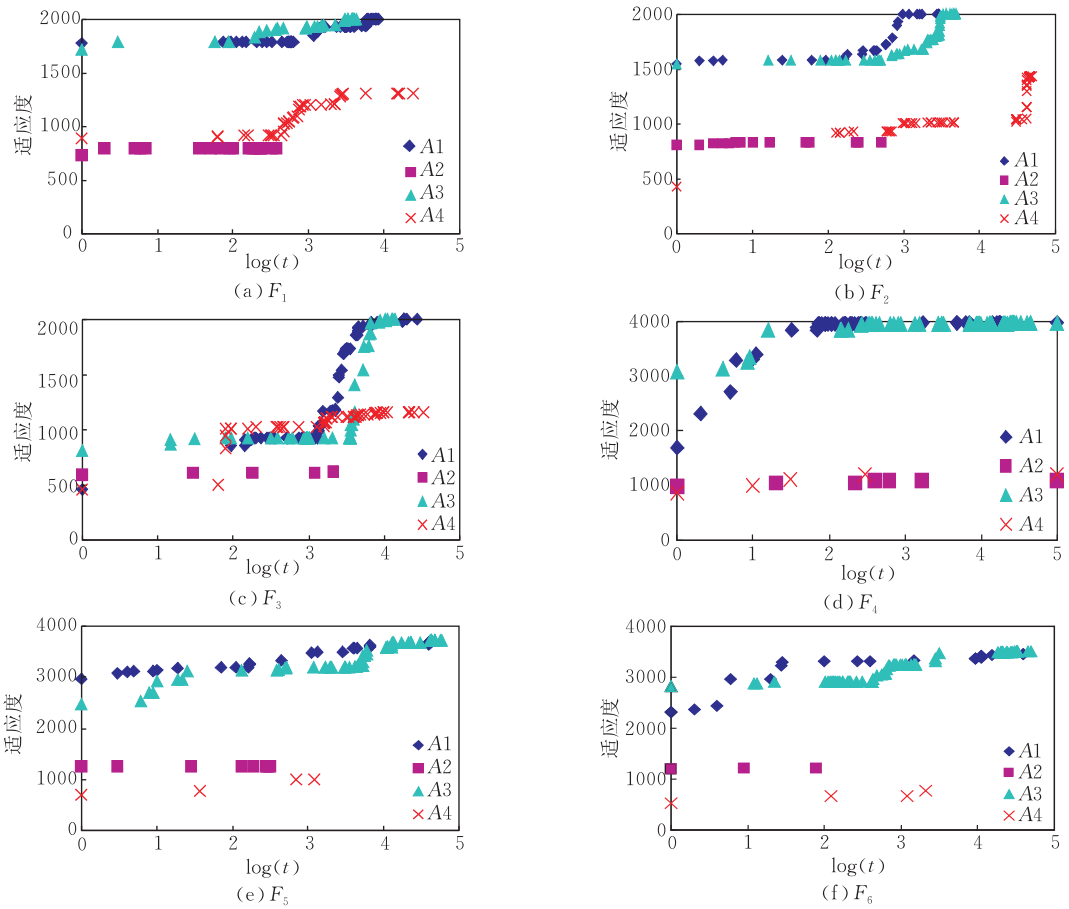


图 9 算法在 $F_1 \sim F_6$ 上的收敛性

8.2 真实数据上的实验

(1) 实验数据

在引言部分已经指出天气中的温度序列是一种具有混沌特性的迭代动力系统，在大尺度宏观趋势上(如 100 年)具备较强的周期规律，而在小尺度微观趋势上(如 3 年)则呈现出非周期的混沌状态[24].

基于以上的先验知识，我们取了北半球 1851 年～1983 年共 133 年的平均温度序列数据[25]，将温度序列数据分别进行了尺度为 4,3,2 的小波变换以造成数据从稳定到混沌的趋势，其中当尺度为 2 时，已经基本接近原始数据. 同时，由于 1919 年是地球温度变暖的标志年，温度变化的迭代规律有所变化[25]，故我们将数据序列分成两部分，前一部分从 1854 年开始，后一部分从 1921 年开始. ① 混沌迭代序列. 共 6 组数据 $T_1 \sim T_6$，采样方式如表 7. 其中，a 表示小波变换的尺度，t_0 表示采样数据的起始年分，每隔 3 年采样一个数据，共采样 20 个数据. c 表示原始数据，c' 表示对原始数据作的预处理，这主要是为了避免气候数据绝对值过小造成适应度计算偏差. 由于有一个尺度参数，故待挖掘耦合函数含有一个参量. ② 训练数据：$T_1 \sim T_6$ 中，取 $x(0) \sim x(9)$ 为训练数据. ③ 测试数据：$T_1 \sim T_6$ 中，取 $x(10) \sim x(19)$ 为测试数据.

表 7 $T_1 \sim T_6$ 的采样方式

数组	采样方式
T_1	$a=4$, $t_0=1854$, $c'=(c+0.2)\times 100$
T_2	$a=4$, $t_0=1924$, $c'=c+0.13$
T_3	$a=3$, $t_0=1854$, $c'=c+0.15$
T_4	$a=3$, $t_0=1924$, $c'=c+0.12$
T_5	$a=2$, $t_0=1854$, $c'=c$
T_6	$a=2$, $t_0=1924$, $c'=c+0.2$

(2) 实验结果

① 序列拟合情况. 表 8 是 A_3 在 10 次挖掘中得到的最优迭代函数和相应参数，图 10 是函数迭代 20 次产生的拟合数据与原数据的对比. 可知，挖掘出的函数虽然在序列中某些点的绝对数值上拟合没有完全吻合，但在序列的周期趋势上和拐点分布上能够较好拟合原序列. ② 算法收敛情况：与合成数据的实验结果类似，A_1，A_3 能在 1000～10000 代的数量级上达到较高的适应度，A_2，A_4 很难拟合出与原序列近似的迭代序列，但算法 A_1 虽然能够达到与

A_3 相近的适应度，但其拟合出的序列无法拟合 T_1～T_6 中的各个峰值和拐点，总是收敛到一组较平直的序列，由于篇幅原因此处不再详细列出.

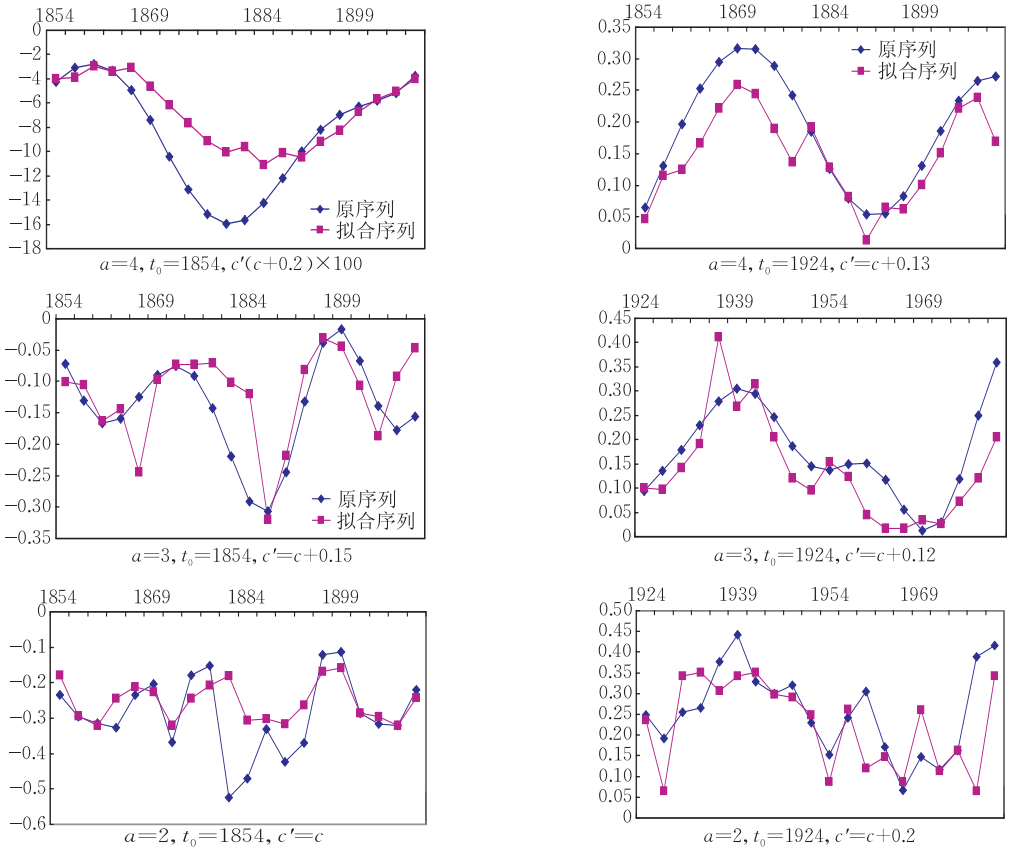


图 10　A_3 对 T_1～T_6 挖掘结果的迭代序列

表 8　A_3 对 T_1～T_6 在 10 次实验中的最优挖掘结果

	函数结构	参数
T_1	$x(t+1)=(x(t))-(\tan(\cos(((\cos(x(t)))\times((x(t))\times(x(t))))+(x(t)))))$;	−7.9
T_2	$x(t+1)=((r)\times(\cos(\tan((r)/((x(t))\times(x(t)))))))+(x(t))$;	0.1
T_3	$x(t+1)=(\exp(\cos((\sin(\exp(x(t))))/(\tan(x(t))))))\times(\tan(x(t)))$;	32.1
T_4	$x(t+1)=(x(t))/(\exp(\sin(\tan(((\sin(r))/(x(t)))+(\sin(x(t))))))))$;	−1.9
T_5	$x(t+1)=\sin(((\sin((r)/(x(t))))/(((r)+(r))+(x(t))))+(\tan(r)))$;	6.0
T_6	$x(t+1)=\exp((r)+(\sin(\tan(\sin(\exp(((x(t))+(x(t)))-(r)))))))$;	−2.0

9　总结及今后的工作

本文采用 GEP 种群和粒子群协同进化的方式进行混沌迭代序列的挖掘. 从两个种群结合发多样性和稳定性等角度，提出了三种不同的协同进化算法，并且从理论上分析了三种算法的性能. 在实验部分，在合成数据和真实数据上进行了翔实的实验，证实了联合式种群结合算法能得到相对较好的函数结构和参数.

由于多路并行式种群结合算法也是收敛到局部最优，故下一步的工作除了继续优化种群的结合方式，还将深入到种群个体自身的构造上，通过相辅相成的改进种群个体构造及种群结合方式挖掘出最优的结果.

参　考　文　献

[1] Chen Heng-Hui. Chaos synchronization between two different chaotic systems via nonlinear feedback control. Nonlinear Analysis: Theory, Methods & Applications, 2009, 70(12): 4393-4401

[2] Wang Hua. Finite-time chaos synchronization of unified chaotic system with uncertain parameters. Communications in Nonlinear Science and Numerical Simulation, 2009, 14(5): 2239-2247

[3] Ranjit Kumar. Dynamics of an ecological model living on the edge of chaos. Applied Mathematics and Computation, 2009, 210(2): 455-464

[4] Devaney R L. An introduction to chaotic dynamical systems. Menlo Park, Benjamin/Cummings Publishing Company,

1986

[5] Boccaletti S. Complex networks: Structure and dynamics. Physics Report, 2006, 424(7): 175-308

[6] Schneckenreither G, Popper N, Zauner G, Breitenecker F. Modeling SIR-type epidemics by ODEs, PDEs, difference equations and cellular automata — A comparative study. Simulation Modeling Practice and Theory, 2008, 16(8): 1014-1023

[7] Ramani A, Carstea A S, Willox R, Grammaticos B. Oscillating epidemic: A discrete-time model. Physica A, 2005, 333(1-2): 278-292

[8] Zhao Hai, Xu Ye, Su Wei-Ji, Zhang Wen-Bo, Zhang Xin. Analysis of short-term and long-term forecast of weighted Internet traveling diameter. Journal of Computer Research and Development, 2006, 43(6): 1027-1035(in Chinese)
(赵海, 徐野, 苏威积, 张文波, 张昕. 加权 Internet 访问直径短期及长期预测行为分析. 计算机研究与发展, 2006, 43(6): 1027-1035)

[9] Liu Wei, Wang Ke-Jun, Shao Ke-Yong. Predicting chaotic time series using hybrid particle swarm optimization algorithm. Control and Decision, 2007, 22(5): 562-565(in Chinese)
(刘伟, 王科俊, 邵克勇. 混沌时间序列的混合粒子群优化预测. 控制与决策, 2007, 22(5): 562-565)

[10] Li Tian-Shu, Tian Kai, Li Wen－Xiu. Chaotic time series prediction based on transiently chaotic neural network. Journal of Harbin Engineering University, 2007, 28(2): 165-168(in Chinese)
(李天舒, 田凯, 李文秀. 基于暂态混沌神经网络的低阶混沌时间序列预测. 哈尔滨工程大学学报, 2007, 28(2): 165-168)

[11] Guo Yang-Ming, Zhai Zheng-Jun, Jiang Hong-Mei. Weighted prediction of multi-parameter chaotic time series using Least Squares Support Vector Regression (LS-SVR). Journal of Northwestern Polytechnical University, 2009, (1): 1125-1130(in Chinese)
(郭阳明, 翟正军, 姜红梅. 基于新息的多参量混沌时间序列 LS-SVR 加权预测. 西北工业大学学报, 2009, (1): 1125-1130)

[12] Han Min, Li De-Cai. Multiple time series correlation extraction and prediction based on EOF-SVD model. Journal of System Simulation, 2008, 20(7): 1669-1672(in Chinese)
(韩敏, 李德才. 基于 EOF-SVD 模型的多元时间序列相关性研究及预测. 系统仿真学报, 2008, 20(7): 1669-1672)

[13] Candia Ferreria. Gene Expression Programming Mathematical Modeling by an Artificial Intelligence. Berlin: Springer, 2006

[14] Xu Kaikuo, Liu Yintian, Tang Rong, et al. A novel method for real parameter optimization based on Gene Expression Programming. Journal of Applied Soft Computing, 2009, 9(2): 725-737

[15] Jiang Yue, Tang Chang-Jie, Zheng Hai-Chun, et al. Adaptive gene expression programming algorithm based on cloud model//Proceedings of the 2008 International Conference on BioMedical Engineering and Informatics. Beijing, Springer, 2008: 226-230

[16] Li Tai-Yong, Tang Chang-Jie, Wu Jiang, et al. GEP-NFM: Nested function mining based on gene expression programming//Proceedings of the 4th International Conference on Natural Computation. Anhui, China, 2008, 6: 283-287

[17] Liang J J, Qin A K, Suganthan P N, et al. Comprehensive learning ParamParticle swarm optimizer for global optimization of multimodal functions. IEEE Transactions on Evolutionary Computation, 2006, 10(3): 281-295

[18] Wilke D N, Kok S, Groenwold A A. Comparison of linear and classical velocity update rules in ParamParticle swarm optimization: Notes on diversity. International Journal for Numerical Methods in Engineering, 2007, 70(8): 962-984

[19] Makoto Koshino, Hiroaki Murata, Haruhiko Kimura. Improved ParamParticle swarm optimization and application to portfolio selection. Electronics and Communications in Japan (Part III: Fundamental Electronic Science), 2007, 90(3): 13-25

[20] Xu Kaikuo, Tang Changjie, Tang Rong, et al. Application of gene expression programming to real parameter optimization//Proceedings of the 4th International Conference on Natural Computation. Anhui, China, 2008, 6: 273-277

[21] Greenberg S, Randall D. Convergence rates of Markov chains for some self-assembly and non-saturated Ising models. Theoretical Computer Science, 2006, 34(15): 1417-1427

[22] Suganthan P N, Hansen N, Liang J J, et al. Problem definitions and evaluation criteria for the CEC 2005 Special Session on Real-Parameter Optimization. Nanyang Technological University, Singapore: Technical Report 2005005, 2005

[23] Zuo jie. Study on congestion control of high-speed network [Ph. D. dissertation]. Sichuan University, Chengdu, 2004 (in Chinese)
(左劼. 基因表达式编程核心技术研究[博士学位论文]. 四川大学, 成都, 2004)

[24] Fu Zun-Tao, Liu Shi-Da, Chen Jiong, Liu Shi-Kuo, Liang Shuang. Period 2,3,5 and their prediction for climatic jump points. Earth Science Frontiers, 2003, 10(2): 415-418(in Chinese)
(付遵涛, 刘式达, 陈炯, 刘式适, 梁爽. 周期 2,3,5 及其气候突变点预测. 地学前缘, 2003, 10(2): 415-418)

[25] Jones P D. Northern Hemisphere surface air temperature variations. 1851～1984. Climate Application, 1986, 25(2): 161-179

附录：异构种群协同进化的混沌迭代函数挖掘的若干辅助算法细节

(1) GEP 种群当前代全局最优个体的生成算法

算法：$Generate\_Best\_StruChrom(StruChrom_i(t), f)$；

//判断 $StruChrom_i(t)$是否是最优个体

输入：GEP 种群 t 代第 i 个个体 $StruChrom_i(t)$，$StruChrom_i(t)$的适应度 f

输出：GEP 种群第 t 代全局最优个体 $Best\_StruChrom(t)$

1. If($f>Best\_StruChrom\_Fitness(t)$)
2. $Best\_StruChrom\_Fitness(t)=f$;
3. $Best\_StruChrom(t)=StruChrom_i(t)$;
4. End If
5. return ($Best\_StruChrom(t)$);

(2) 粒子群当前代全局最优个体的生成算法

算法：$Generate\_Best\_Global\_ParamParticle(ParamParticle_j(t), f)$；//判断 $Param\ Particle_j(t)$是否最优个体

输入：第 t 代粒子群 $PSO\_Pop(t)$

输出：第 t 代全局最优个体 $Global\_Best\_ParamParticle(t)$

1. If ($f>Global\_Best\_ParamParticle\_Fitness(t)$)
2. $Global\_Best\_ParamParticle\_Fitness(t)=f$;
3. $Global\_Best\_ParamParticle(t)=ParamParticle_j(t)$;
4. End If
5. return($Global\_Best\_ParamParticle(t)$);

(3) 粒子群进化算法

算法：$Evolute\_PSO\_Population(t)$；

输入：第 t 代粒子群 $PSO\_Pop(t)$

输出：进化后的 $t+1$ 代种群 $PSO\_Pop(t+1)$

1. For each $ParamParticle_j(t)$ in $PSO\_Pop(t)$
2. For each dimention $x_{jp}(t)$ of $ParamParticle_j(t)$
3. $v_{jp}(t+1)=w\times v_{jp}(t)+c_1\times Random()\times(Global\_Best\_ParamParticle(t)-x_{jp}(t))$;
4. $x_{jp}(t+1)=x_{jp}(t)+v_{jp}(t)$;
5. End For
6. $ParamParticle_j(t+1)=\{x_{j1}(t+1),\cdots,x_{jp}(t+1)\}$;
7. $PSO\_Pop(t+1)+=ParamParticle_j(t+1))$;
8. End For
9. return($PSO\_Pop(t+1)$);

注：由于 $ParamParticle_j(t)$所对应的 $StruChrom_j(t)$每一代都可能发生变化，故在每个粒子 $ParamParticle_j$的学习公式中屏蔽了 $ParamParticle_j$的局部最优个体 $Local_Best_ParamParticle_j(t)$的影响.

(4) 计算挖掘出的耦合迭代函数的适应度

算法：计算耦合迭代函数的适应度 $Fitness(StruChrom_i(t), ParamParticle_j(t))$

输入：(1) 构造迭代函数的 $StruChrom_i(t)$，$ParamParticle_j(t)$；(2) 待挖掘的一阶 n 元的耦合迭代序列 $\{S_1, S_2, \cdots, S_n\}$，$S_i=\{x_i(0), x_i(1), \cdots, x_i(T)\}$ ($i\in[1,n]$)；(3) 迭代次数 T，适应度规范因子 H；

输出：由 $StruChrom_i(t)$和 $ParamParticle_j(t)$构成的迭代函数的适应度 $Fitness$

1. $Fitness=0$;
2. $\{f_1, f_2, \cdots, f_n\}=Join(StruChrom_i(t), ParamParticle_j(t))$;

//结合 $StruChrom_i(t)$，$ParamParticle_j(t)$得到的

//n 个一阶 n 元 m 参量的耦合迭代函数

3. For $i=1$ To n
4. $y_i(0)=x_i(0)$;
5. End For
6. For $t=0$ To $T-1$
7. For $k=1$ To n
8. $y_k(t)=f_k(y_1(t-1), y_2(t-1), \cdots, y_n(t-1))$;
9. $Fitness+=H-|(y_k(t)-x_k(t))/x_k(t)|\times 100$;
10. End For
11. End For
12. return $Fitness$;

ZHENG Jiao-Ling, born in 1981, Ph. D. candidate, lecturer. Her research interests include data base and knowledge engineering.

TANG Chang-Jie, born in 1981, professor, Ph. D. supervisor. His research interests include data base and knowledge engineering.

XU Kai-Kuo, born in 1983, Ph. D. candidate. His research interests include machine learning and data mining.

CHEN Yu, born in 1974, Ph. D., lecturer. His research interests include data base and knowledge engineering.

YANG Ning, born in 1974, Ph. D. candidate. His research interests include machine learning and data mining.

DUAN Lei, born in 1981, Ph. D., lecturer. His research interests include data base and knowledge engineering.

Background

This study tries to conduct regression on chaotic time series data by co-evolutionary computing. In the filed of chaos theory, researchers always first construct a mathematic model and then study the chaotic properties of the model. Few studies discuss the reverse problem, i. e. regression of chaotic time series data generated by a mathematic model or by a real chaotic dynamic system. In the filed of co-evolutionary computing, many studies concentrate on the co-evolution of homogeneous populations. Few researchers investigate the co-evolution of heterogeneous populations. The experimental results of this study show that co-evolution of heterogeneous populations can take the advantages of different populations. The regression precision on chaotic time series data is comparably high.

This study is supported by the National Natural Science Foundation of China under grant No. 60473071 and the National Key Technologies R&D Program of China during the 11th Five-year Plan Period under grant No. 2006BAI05A01. The National Key Technologies R&D Program is to find the intervention rules in birth defect. The NSFC program is to find the dynamic characteristics of sub complex system under artificial interventions. This study is derived from the NSFC program. It tries to discover the regularities behind those chaotic time series data and to investigate the characteristics of co-evolution populations. The institute of database and knowledge engineering of Sichuan University has done a lot of previous researches both in evolutionary computing and data mining.