

# 常微分方程组并行演化建模的实验研究\*

曹宏庆<sup>1+</sup>, 康立山<sup>1</sup>, 陈毓屏<sup>1</sup>, 胡庆丰<sup>2</sup>

<sup>1</sup>(武汉大学 软件工程国家重点实验室,湖北 武汉 430072)

<sup>2</sup>(国防科学技术大学 计算机学院,湖南 长沙 410073)

## Experimental Study on the Parallel Evolutionary Modeling of System of Ordinary Differential Equations

CAO Hong-Qing<sup>1+</sup>, KANG Li-Shan<sup>1</sup>, CHEN Yu-Ping<sup>1</sup>, HU Qing-Feng<sup>2</sup>

<sup>1</sup>(State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, China)

<sup>2</sup>(School of Computer, National University of Defence Technology, Changsha 410073, China)

+ Corresponding author: Phn: 86-27-87653091, E-mail: jxyu@mail.whu.edu.cn

<http://nic.whu.edu.cn/~jxyu/caohq>

Received 2002-02-05; Accepted 2002-04-18

**Cao HQ, Kang LS, Chen YP, Hu QF. Experimental study on the parallel evolutionary modeling of system of ordinary differential equations. *Journal of Software*, 2003,14(3):443~450.**

**Abstract:** A distributed asynchronous parallel evolutionary modeling algorithm for solving the evolutionary modeling problem of system of ordinary differential equations is proposed in this paper. Performed on a simulated parallel environment consisting of 128 Pentium III 500 PCs connected by a 10Mbps Ethernet, large-scale experiments have been done to systematically test the effect of some important parallel control parameters, which include the degree of connectivity between processors, the migration rate and the migration interval of individuals, on the performance of the algorithm. Many new experimental results are achieved as well as some analysis and explanations are given. Especially the best modeling results for the sequential algorithm and the parallel one are compared at the end.

**Key words:** parallel genetic programming; parallel control parameter; evolutionary modeling; system of ordinary differential equations

**摘要:** 以常微分方程组的演化建模问题为主要研究对象,设计了分布式异步并行演化算法,并以 128 台 PIII 500 微机通过 10Mbps 的以太网互联而成的机群系统作为模拟实验环境进行了大规模的并行实验,系统地测试了算法中的一些重要的并行控制参数,包括处理机间的连通度、个体的迁移率和迁移代频等对算法性能的影响,

\* Supported by the National Natural Science Foundation of China under Grant Nos.70071042, 60073043, 60133010 (国家自然科学基金); the Assisting Project of Ministry of Education of China for Backbone Teachers of University and College (国家教育部高等学校骨干教师资助计划); the Natural Science Foundation of Hubei Province of China under Grant No.2001ABB062 (湖北省自然科学基金); the Youth Chenguang Project of Science and Technology of Wuhan City of China under Grant No.20015005037 (武汉市青年科技晨光计划); the Program of the State Key Laboratory of Parallel and Distributed Processing of China (并行与分布处理国家重点实验室资助项目)

第一作者简介: 曹宏庆(1972—),女,湖北咸宁人,博士,副教授,主要研究领域为演化建模,遗传程序设计,并行计算。

得到了一些崭新的实验结果,给出了一些结果分析,特别是对串行算法和并行算法的最好建模结果进行了比较.

关键词: 并行遗传程序设计;并行控制参数;演化建模;常微分方程组

中图法分类号: TP18 文献标识码: A

遗传程序设计(genetic programming,简称 GP)<sup>[1,2]</sup>是 20 世纪 90 年代初在遗传算法(genetic algorithm,简称 GA)<sup>[3]</sup>的基础上发展起来的演化计算的一个新分支,它为人们提供了进行自动程序设计的一种新思路和新方法,目前正得到越来越广泛的应用.我们曾基于此方法研究了动态系统的常微分方程组建模问题<sup>[4-7]</sup>.在研究中我们发现,由于在计算模型适应值的过程中需要进行微分方程组的多步积分,建模的时间开销直接与输入的历史数据量成正比,因此在很大程度上限制了所解决问题的规模和复杂性.另一方面,演化算法的一些控制参数,如种群的规模、最大演化代数、最大树深等,由于受到运行时间的限制也不可能设置得太大,因此,也在很大程度上制约了演化算法潜能的充分发挥,从而影响最终解的质量.为了能在较短的运行时间内找到精确度更高的模型,我们考虑采用并行计算方法实现常微分方程组的演化建模.

演化算法的基本结构和特征决定了它特别适合大规模并行.Bethke 于 1976 年对并行遗传算法开始了最早的研究工作<sup>[8]</sup>.此后人们根据不同的模型研究了各种不同类型的并行 GA,大致包括全局并行 GA、粗粒度并行 GA、细粒度并行 GA 和混合并行 GA 这 4 类<sup>[9]</sup>.其中,粗粒度并行 GA 又称为分布式 GA 或“孤岛”式 GA,其基本原理是将种群划分为若干子种群(demes),各自独立地进行演化,同时在演化过程中,子种群间按照一定的时间间隔进行个体迁移.由于这种并行方式原理清晰,对实验环境要求较低,很容易用 PVM 或 MPI 在工作站网络或机群系统上实现,因而已成为目前并行演化计算研究的主流.自从遗传程序设计技术出现以后,人们同时也开始了并行 GP 的研究.Andre 等人<sup>[10]</sup>首先采用 64 个结点构成的网络和一台 486 PC 机实现了细粒度的并行 GP,用于求解 5 位偶校验器问题,并获得了超线性加速比.Nordin 等人<sup>[11]</sup>采用 Parsytec Power Explorer 并行机实现了并行 AIM-GP 系统,求解了布尔校验、函数回归和图像分类 3 个问题,并对迁移率和迁移策略进行了一些实验研究.Punch 等人<sup>[12]</sup>则分别采用单种群和粗粒度多种群方法对 Royal Tree 问题进行了研究,得到的实验结果很出人意料:采用任何并行设置下的多种群方法得到的结果总是比单种群方式要差.近年来,尽管人们对于并行 GP 的研究已日益增多,但大多数都是针对特定问题的应用研究,对于算法内部的一些重要的控制参数,如处理机间的连通度、个体的迁移率和迁移代频(子种群间每迁移一次间隔的代数)等,则缺乏较系统的研究,而这些参数的不同设置往往直接影响到并行算法的性能.对于粗粒度的并行 GA,美国 Illinois 遗传算法实验室的 Cantú-Paz<sup>[13]</sup>曾对其部分参数作了较深入的研究,得到了一些指导性的理论和实验结果,对于一些较复杂的 GP 问题,这些结论是否依然成立则有待进一步的考察.

基于以上原因,本文以较复杂的常微分方程组的演化建模为典型的 GP 问题,设计了有效的分布式异步并行算法,通过大规模实验,研究了算法中一些重要的并行控制参数对算法性能的影响,并且对串行算法和并行算法所得到的最好结果进行了比较.

## 1 常微分方程组的演化建模

### 1.1 问题描述

假设一个动态系统是由  $n$  个相互关联的函数  $x_1(t), x_2(t), \dots, x_n(t)$  来加以描述的.已知该系统在时刻  $t_j$  ( $j=0, 1, 2, \dots, m-1$ ) 的观测数据为

$$X = \begin{pmatrix} x_1(t_0), & x_2(t_0), & \dots, & x_n(t_0) \\ x_1(t_1), & x_2(t_1), & \dots, & x_n(t_1) \\ \vdots & \vdots & \vdots & \vdots \\ x_1(t_{m-1}), & x_2(t_{m-1}), & \dots, & x_n(t_{m-1}) \end{pmatrix}, \quad (1)$$

其中  $t_j = t_0 + j \times \Delta t$  表示观测时间,  $\Delta t$  为时间步长,要求用一阶常微分方程组

$$\begin{cases} dx_i/dt = f_i(x_1, x_2, \dots, x_n, t) & (i=1, 2, \dots, n) \\ x_i^0 = x_i(t_0) \end{cases} \quad (2)$$

来描述(拟合)该组数据,即在模型空间中寻找一个形如式(2)的模型作为该动态系统的模型,使得 $\|X^*-X\|$ 最小,其中

$$\|X^*-X\| = \sqrt{\sum_{i=0}^{m-1} \sum_{j=1}^n (x_j^*(t_i) - x_j(t_i))^2}, \quad (3)$$

并可由此预报系统的未来状态.这就是动态系统的常微分方程组建模问题.

## 1.2 串行演化建模算法

常微分方程组串行演化建模算法采用的是两级演化建模的思想,即在以 GP 技术优化模型结构的同时嵌套以 GA 优化模型参数的过程.算法的基本结构如下:

**PROCEDURE** 常微分方程组的串行演化建模

begin

  输入观测数据矩阵  $X(0)$ ;

  初始化模型种群  $P(0)$ ;

  Geno := 0;

  repeat

    采用 GP 的模型的结构优化过程;

    模型的简洁化和规范化处理;

    采用 GA 的模型参数优化过程;

    Geno := Geno + 1;

  until 满足终止条件;

  输出  $P(\text{Geno})$ 中的最好模型并进行系统预测;

end

其中,采用 GP 技术进行模型结构优化的过程请参见文献[4]中第 4.2 节“自适应的演化建模过程”部分,进行参数优化的 GA 算法详见文献[6]中第 5 节“III类算法:GP+GA 算法”的有关描述.由于篇幅有限,这里不再赘述.

## 1.3 异步并行演化建模算法

并行演化建模算法的设计采用的是分布式异步并行演化算法,其基本步骤如下:

**Step 1.** 由一台主处理机读入建模的数据文件、演化算法的一些控制参数(包括种群规模  $n$ 、最大演化代数  $\text{Maxgeno}$ 、最大树深  $D$  等)以及并行算法的一些控制参数(包括创建的进程数  $P$ 、处理机间的连通度  $\delta$ 、迁移率  $\rho$ 、迁移代频  $g$  等),然后创建进程,将这些信息发送到其他  $P$ 台从处理机.

**Step 2.** 主处理机和  $P$  台从处理机各自独立地进行演化建模的过程,包括种群的初始化、模型的结构优化和参数优化(执行选择-杂交-变异过程)和个体的适应值评估等.同时,每台处理机在演化建模的过程中还与其他处理机进行信息交换,包括:

(1) 每隔  $g$  代将当前种群中的  $m$  个最好的个体发送到其他  $\delta$ 台邻接的处理机(在原种群中仍保留其副本).其中  $m$  由迁移率  $\rho$ 和子种群大小  $n_d$  决定,即  $m = \rho \times n_d (n_d = n/(P+1))$ .

(2) 在每一演化代多次检查有无其他处理机发送来的个体.如果有,则用接收到的  $m$  个个体替换当前种群中的  $m$  个最差个体;如果没有,则继续执行下面的操作.

**Step 3.** 当主处理机运行到最大代数  $\text{Maxgeno}$  时,则向其他  $P$  台从处理机发送终止消息,并将自身种群中最好的模型作为最终模型,输出其拟合与预测结果.而其他  $P$  台从处理机接收到主处理机的终止命令后即停止运行.

上述算法具有以下几个特点:

(1) 虽然处理机间个体的发送是每隔固定代数进行的,但各处理机内个体的接收是采用每代频繁检测的方

式,有则接收,无则继续,从而处理机间无须等待,实现了完全地异步并行.这是算法最重要的一点.

(2) 算法直接选取主处理机自身种群里最好的个体作为最终模型,而没有采取主处理机从各从处理机中收集最好个体再进行择优的常用方法.因为若采用后者会使得主处理机的消息负荷太重,尤其是当邻接处理机数目很大时.实际上,经过频繁的个体迁移之后,主处理机已从其他处理机那里获得足够的局部最优信息,保证了自身种群的多样性.

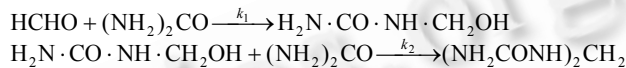
(3) 算法中所考虑的并行控制参数较多且灵活可调,有利于开展较系统的实验研究.

## 2 并行建模实验

### 2.1 建模实例

我们选取化学反应动力学中的一个典型的连串反应作为计算实例:

甲醛( $x_1$ )在水溶液中与尿素反应生成羟甲基脲( $x_2$ ),再与尿素反应生成亚甲基二脲( $x_3$ ),其反应式为



反应在 308.15K 和过量的尿素  $c_{2(0)} = 2\text{mol} \cdot \text{dm}^{-3}$  的条件下进行,催化剂 HCl 的浓度为  $0.0008\text{mol} \cdot \text{dm}^{-3}$ ,甲醛的初始浓度  $x_{1(0)}$  为  $0.1\text{mol} \cdot \text{dm}^{-3}$ . 反应体系物质浓度满足如下微分方程组

$$\begin{cases} dx_1/dt = k'_1 x_1 \\ dx_2/dt = k'_1 x_1 - k'_2 x_2 \\ dx_3/dt = k'_2 x_2 \end{cases} \quad (4)$$

式中  $k'_1 = k_1 c_{2(0)}$ ,  $k'_2 = k_2 c_{2(0)}$ . 其中  $k_1 = 0.007\text{dm}^3 \cdot \text{mol}^{-1} \cdot \text{min}^{-1}$ ,  $k_2 = 0.021\text{dm}^3 \cdot \text{mol}^{-1} \cdot \text{min}^{-1}$ . 根据连串反应的浓度解

$$\begin{cases} x_1 = x_{1(0)} e^{-k'_1 t} \\ x_2 = \frac{k'_1 x_{1(0)}}{k'_2 - k'_1} (e^{-k'_1 t} - e^{-k'_2 t}) \\ x_3 = x_{1(0)} - x_1 - x_2 \end{cases} \quad (5)$$

我们计算出反应发生 110 分钟内每间隔 2 分钟各物质的浓度值作为仿真数据,取前 50 个点为建模样本点演化建立常微分方程组,并以后 5 个点作为待估样本点.

### 2.2 实验设置

本文的模拟并行实验环境是由 128 台 PIII 500(128M 内存)微机构成的机群系统,计算机之间通过 10Mbps 以太网连接,用 PVM3.3 编程实现.

为了将并行结果与串行结果作对比,我们首先运行了 10 次串行演化建模程序,演化建模的有关参数设置见表 1.

Table 1 Parameter settings of the evolutionary modeling

表 1 演化建模的参数设置

Structure optimization of model (GP)	Function set: $F = \{+, -, *, /, \wedge, \sin, \cos, \exp, \ln\}$ (where $x^k$ denotes $x^k$ ) Terminal set: $T = \{x_1, x_2, x_3, t, c\}$ (where $c$ is a constant) Other parameters: $n=6400, D=3, \text{Maxgeno}=100$
Parameter optimization of model (GA)	$N=30, M=8, a=-0.5, b=1.5$ The max number of crossovers: $\text{MAX}=200$ (The definitions of all parameters are referred to reference[6])

在所有的并行实验中,我们所使用的处理机总数固定为 128,即子种群数为 128,各子种群独立进行演化建模的参数设置同表 1,只是规模  $n_d = 6400/128=50$ . 其他并行控制参数的设置因实验目的不同而由所不同,具体请参见第 2.3 节的有关内容.

对于并行算法的性能,我们主要从最终解的质量和并行加速比两方面来进行评估.其中评价解质量的好坏主要是通过计算模型的拟合误差 FE(即适应值):

$$FE = \sqrt{\sum_{i=1}^{50} \sum_{j=1}^3 [x_j^*(t_i) - x_j(t_i)]^2}, \quad (6)$$

拟合误差越小,则解越好.此外,我们还计算了模型的预测误差 PE:

$$PE = \sqrt{\sum_{i=51}^{55} \sum_{j=1}^3 [x_j^*(t_i) - x_j(t_i)]^2}. \quad (7)$$

对于并行算法的加速比  $S_p$ ,则定义为串行执行时间  $T_s$  与并行执行时间  $T_p$  之比,即

$$S_p = T_s / T_p. \quad (8)$$

对于每组实验均独立运行 10 次,计算平均拟合误差 AFE 和平均运行时间,进而求得并行加速比  $S_p$ .

### 2.3 有关实验及结果分析

#### 2.3.1 并行控制参数对最终解质量的影响

这部分实验主要考察决定算法通信开销的 3 个并行控制参数,即处理机间的连通度、个体的迁移率和迁移代频对解质量的影响及其相互关系.

根据文献[13]的研究结果:具有相同的图连通度的不同通信拓扑结构所获得的解的质量几乎相同,我们在实验中进行处理机间的信息交换时只需考虑拓扑的连通度  $\delta$ ,即每台处理机的邻接处理机数,而不必关心具体采用的是何种拓扑结构.为方便起见,这里考虑的是一种最简单的拓扑连接方式,即若某台处理机的编号为  $k$ ,则与其邻接的  $\delta$  台处理机分别为  $\{k+1, k+2, \dots, k+\delta\} \bmod (P+1)$ .不失一般性,对于  $\delta$ ,我们考虑了两种极端情况和一种中间情况,即  $\delta=1$ (单连接),127(全连接)和 64(半连接).对于每种情况,我们均通过设置不同的迁移代频  $g=1, 5, 10, 20$  和迁移率  $\rho=0\%, 2\%, 6\%, 10\%, 14\%$ (即  $m=0, 1, 3, 5, 7$ )进行了组合实验,所得到的 3 种情况下的 AFE 值随  $g$  值和  $\rho$  值变化的曲线如图 1 所示.

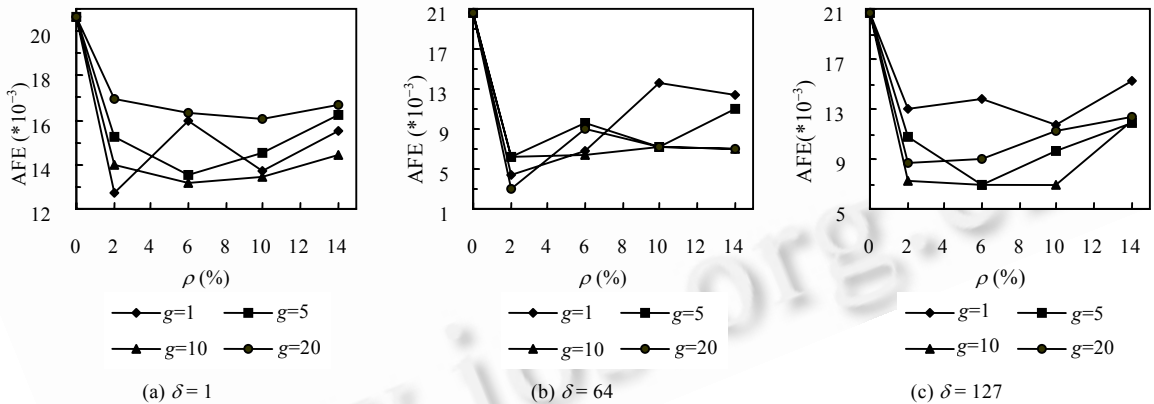


Fig.1 The average fitting errors (AFE) in three cases of connectivity ( $\delta$ ) by varying the migration rate ( $\rho$ ) and the migration generation interval ( $g$ )

图 1 处理机间连通度  $\delta$  不同时,迁移率  $\rho$  和迁移代频  $g$  的变化对平均拟合误差 AFE 的影响

对于此实验结果,我们有如下讨论:

(1) 由图 1 可见,当子种群间无个体迁移(即  $\rho=0\%$ )时,AFE 值最大.实际上,这个值 0.020 678 比串行实验的平均结果 0.016 009 要大得多.而一旦子种群间发生了个体迁移,则无论  $g$  和  $\rho$  取值如何,解的平均质量均有明显改善.由此可见,采用这种无个体迁移的分种群的并行方式无多大实际意义,因为它获得的解比串行程序还要差.

(2) 对图 1 中各图作进一步的观察与分析后可见,为确保获得较好的解,这 3 个并行参数之间明显地存在着一种制约与平衡关系.这可以由以下实验现象来说明:

首先,由图中各曲线所呈现的上下起伏以及不同曲线间错综交杂的现象可见,解的平均质量的好坏并不是孤立地受某一并行参数的影响并单调变化的,它是以上 3 个控制参数综合作用的结果,因而不能简单地认为:个体迁移得越多、越频繁或者邻接的处理机数越多,所获得的解就越好.

其次,从各图最差解的情况来看:当 $\delta=1$ 时, $g=20$ 所得解最差,对于 $\rho$ 而言,则其值为2%或14%时易于导致较差解;当 $\delta=127$ 时, $g=1$ 所得解最差,而 $\rho=14\%$ 时总是一致地获得最差解;当 $\delta=64$ 时,则结果不明显.由此可见,当处理机间连接疏松时,不宜采用不频繁或者迁移个体数太少的交换方式;而当处理机间连接紧密时,过于频繁或者迁移个体数太多的交换方式则应当加以避免.

因此,在连通度、迁移率和迁移代频这三者之间应该保持一种平衡,以保证获得最好的解所必需的基本通信量.至于选取何值最为合适,从图中的实验结果来看,通常采取半连接的方式以及适中的迁移代频和迁移率( $\delta=64, g=10, \rho=6\%$ )所得到的解较好.

### 2.3.2 并行控制参数对并行加速比的影响

设计并行算法的一个重要目的是力求在更少的执行时间内找到至少与串行解同等质量的解.对于串行执行时间与并行执行时间的比较也就涉及到并行加速比问题.类似于图1,我们将算法在3种不同连通度下改变 $g$ 值和 $\rho$ 值得到的加速比曲线绘制于图2.

由图2可见,对于不同的连通度,其加速比曲线呈现出明显不同的特征:

由图2(a)可见,当处理机间仅存在单连接时,在同一 $\rho$ 值下,总是在 $g=1$ 时获得的加速比最小.这是可解释的,因为交换越频繁,处理机间的通信开销越大,从而并行执行时间越长.但在其他所有情况下,改变 $\rho$ 和 $g$ 的值对加速比无明显影响.

由图2(b)可见,当处理机间为半连接时,随着 $\rho$ 值的改变,加速比发生较大变化的是当 $g=1$ 时的情况,并且其加速比值并不随 $\rho$ 值的增大而一致地减小,而是当 $\rho$ 大于某值(这里为6%)时,其值反而增大.

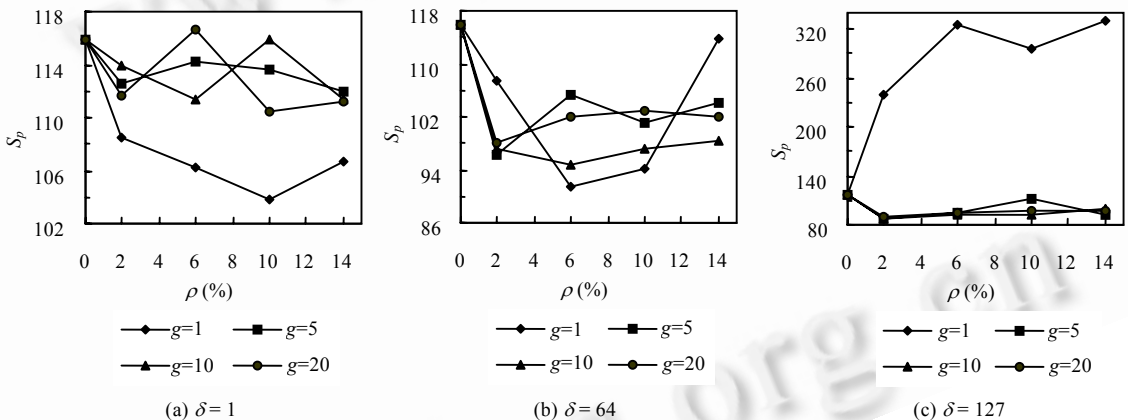


Fig.2 The parallel speedups ( $S_p$ ) in three cases of connectivity ( $\delta$ ) by varying the migration rate ( $\rho$ ) and the migration generation interval ( $g$ )

图2 处理机间连通度 $\delta$ 不同时,迁移率 $\rho$ 和迁移代频 $g$ 的变化对于并行加速比 $S_p$ 的影响

这种变化在 $\rho=14\%$ 时最为明显.这是由迁移率的增大所带来的双面效应所引起的.一方面,迁移个体数的增多使得处理机间的通信时间增大;另一方面,来自其他子种群的大量个体的迁入使得原子种群很快收敛到一个较小的多样性范围,虽然演化的最大代数保持不变,但由于子种群内存在大量的同类模型,而同类模型只进行一次参数优化,所以实际上每一代的运行时间大大缩短.因此,当 $\rho$ 值增大到一定程度,使得其对算法收敛速度的影响更甚于通信时间的代价时,则加速比呈现上升的趋势.

由图2(c)可见,当处理机间为全连接时,图中曲线最显著的特征是当 $g=1$ 时出现了一致的超线性加速比,并且随 $\rho$ 值的增大,加速比呈上升趋势;而当 $g$ 取其他值时则无此现象.对此实验现象,我们的解释是:在这种处理机间的连通度已达最大的情况下,再进行最频繁的个体交换(每隔1代迁移一次)势必会大大加快子种群的收敛速度,并且迁移率越大,收敛越快,从而使得并行算法的执行时间大大缩短,由此出现了超线性加速比.

### 2.3.3 串行结果与并行结果的比较

我们采用串行算法得到的最好的模型为

$$\begin{cases} dx_1/dt = ((89.040329/((-31.643286)/x_1)) \\ dx_2/dt = (((t-x_3)-(x_1/(-0.646973)))-((x_2/x_1)-x_1)), \\ dx_3/dt = (\exp(x_2 \cdot 36.886494))/13.766514) \end{cases} \quad (9)$$

( $FE=0.010541, PE=0.002981$ , 运行时间=39745 秒(约 11 小时))其拟合与预测曲线如图 3(a)所示。

采用并行算法得到的最好模型为

$$\begin{cases} dx_1/dt = ((x_1/0.993726) \cdot (-2.782618)) \\ dx_2/dt = ((\sin(x_1) + (x_2 \cdot (-2.992554))) \cdot (\exp(x_2) + 1.798706)), \\ dx_3/dt = ((x_2 \cdot 8.366699) + 0.000295) \end{cases} \quad (10)$$

( $FE=0.000145, PE=0.00004$ , 运行时间=377 秒(约 6 分钟))并行参数设置为 $\delta=64, g=20, \rho=10\%$ ,其拟合与预测曲线如图 3(b)所示。

由以上结果可见,采用并行算法后不仅运行时间大大减少(减小了 100 多倍),并且搜索到的模型也比串行模型的精确度要高得多(拟合误差约减小了 100 倍),其拟合与预测曲线和原产生数据的仿真曲线几乎完全重合。

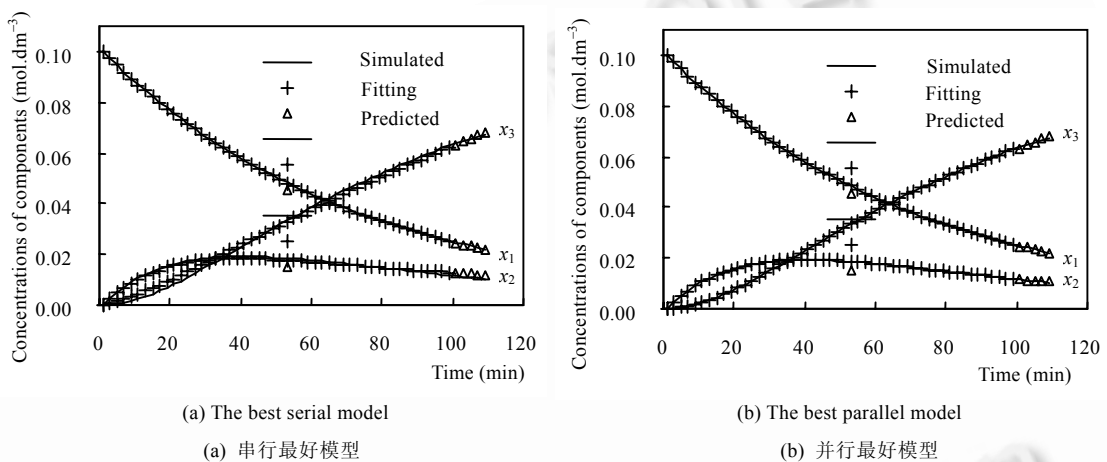


Fig.3 The fitting and predicted curves of the best models under two algorithms

图 3 两种算法下所得最好模型的拟合与预测曲线

### 3 结束语

本文以常微分方程组的演化建模问题为主要研究对象,设计了分布式异步并行演化算法,并以 128 台 PIII 500 PC 机互联而成的机群系统作为模拟的并行环境进行了大规模实验,系统地测试了算法中一些重要的并行控制参数,包括处理机间的连通度、个体的迁移率和迁移代频等对算法性能(包括解的平均质量和并行加速比两方面)的影响,得到了一些崭新的实验结果,并作了一些合理的分析和解释.当然,这些结论的得出是基于我们所解决的具体问题的,尚不能完全从理论上作出严格的证明,它们对于并行 GP 的研究者在解决其他问题时是否适用还有待进一步验证.我们希望本文的研究能为这些研究者的算法设计提供一定的指导性。

**致谢** 本文在实验过程中得到了国防科学技术大学的刘杰、卢宇彤、何新芳、王正华、赵文涛等老师的热心帮助,在此一并表示衷心感谢。

### References:

- [1] Koza JR. Genetic Programming: on the Programming of Computers by Means of Natural Selection. Cambridge, MA: MIT Press, 1992.
- [2] Koza JR. Genetic Programming II: Automatic Discovery of Reusable Programs. Cambridge, MA: MIT Press, 1994.
- [3] Holland JH. Adaptation in Natural and Artificial System. Ann Arbor, MI: University of Michigan Press, 1975.

- [4] Kang LS, Cao HQ, Chen YP. The evolutionary modeling algorithm for system of ordinary differential equations. Chinese Journal of Computers, 1999,22(8):871~876 (in Chinese with English Abstract).
- [5] Cao HQ, Kang LS, Chen YP. A hybrid evolutionary modeling algorithm for dynamic systems. Journal of Computer Research and Development, 1999,36(8):923~931 (in Chinese with English Abstract).
- [6] Cao HQ, Kang LS, Chen YP. The modeling of system of ordinary differential equations for dynamic systems: the experimental study based on different search techniques. Journal of Computer Research and Development, 2001,38(6):746~753 (in Chinese with English Abstract).
- [7] Cao HQ, Kang LS, Chen YP, Yu JX. Evolutionary modeling of system of ordinary differential equations with genetic programming. Genetic Programming and Evolvable Machines, 2000,1(4):309~337.
- [8] Bethke AD. Comparison of genetic algorithms and gradient-based optimizers on parallel processors: efficiency of use of processing capacity. Technical Report, No.197, Ann Arbor, MI: University of Michigan, 1976.
- [9] Cantú-Paz E. A survey of parallel genetic algorithms. IlliGAL Technical Report, No.97003, Urbana, IL: Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, 1997.
- [10] Koza JR, Andre D. Parallel genetic programming on a network of transputers. Technical Report, No.CS-TR-95-1542, Department of Computer Science, Stanford University, 1995.
- [11] Nordin P, Hoffmann F, Francone FD, Brameier M, Banzhaf W. AIM-GP and parallelism. In: Angeline PJ, Michalewicz Z, Schoenauer M, Yao X, Zalzal A, eds. Proceedings of the Congress on Evolutionary Computation, Vol 2. Piscataway, NJ: IEEE Press, 1999. 1059~1066.
- [12] Punch WF, Zongker D, Goodman ED. The royal tree problem: a benchmark for single and multiple population genetic programming. In: Angeline PJ, JrKinnear KE, eds. Advances in Genetic Programming 2. Cambridge, MA: MIT Press, 1996. 299~316.
- [13] Cantú-Paz E. Topologies, migration rates, and multi-population parallel genetic algorithms. IlliGAL Technical Report, No.99007. Urbana, IL: Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, 1999.

#### 附中文参考文献:

- [4] 康立山,曹宏庆,陈毓屏.常微分方程组的演化建模.计算机学报,1999,22(8):871~876.
- [5] 曹宏庆,康立山,陈毓屏.动态系统的演化建模.计算机研究与发展,1999,36(8):923~931.
- [6] 曹宏庆,康立山,陈毓屏.动态系统的常微分方程组建模——基于不同搜索技术的实验研究.计算机研究与发展,2001,38(6):746~753.