基于EDA的并行基因表达式程序设计方法

杜 欣[1,2] 丁立新[1] 谢承旺[1] 陈 莉[1,3]
(武汉大学软件工程国家重点试验室 武汉 430074)[1] (石家庄经济学院 石家庄 050031)[2]
(空军雷达学院 武汉 430019)[3]

摘 要 将分布评估算法(EDA)引入基因表达式程序设计方法中,以提高其收敛速度。为减少计算时间,提高解质量,在加入EDA的基因表达式程序设计方法的基础上设计了同步和异步分布式并行算法,同时比较了同步和异步并行算法。实验结果表明,并行算法提高了运行速度和解质量。最后通过实验分析了迁移代频对并行算法的影响。

关键词 基因表达式程序设计方法,分布评估算法,并行算法,MPI
中图法分类号 TP301 **文献标识码** A

Parallel Gene Expression Programming Based on EDA

DU Xin[1,2] DING Li-xin[1] XIE Cheng-wang[1] CHEN Li[1,3]
(State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430074, China)[1]
(Department of Information and Engineering, Shijiazhuang University of Economics, Shijiazhuang 050031, China)[2]
(Air Force Radar Academy, Wuhan 430019, China)[3]

Abstract In order to reduce the computation time and improve the quality of solutions of Gene Expression Programming (GEP), synchronous and asynchronous distributed parallel GEP algorithm based on Estimation of Distribution Algorithm (EDA) was proposed. The idea of introducing EDA into GEP is to accelerate the convergence speed. Moreover, the improved GEP was implemented by synchronous and asynchronous distributed parallel method based on the island parallel model. Some experiments were done on distributed network connected by twenty computers. The best results of sequential and parallel algorithms were compared, speedup and performance influence of some important parallel control parameters to this parallel algorithm were discussed. The experimental results show that parallel algorithms may approach linear speedup and have better ability to find optimal solution and higher stability than sequential algorithm.

Keywords GEP, EDA, Parallel algorithm, MPI

1 引言

基因表达式程序设计方法(GEP)[1]是在遗传算法(GA)[2]和遗传程序设计(GP)[3]基础上提出的一种基于基因型和表现型的新型演化算法。GEP综合了GA和GP的优点,其基因型采用与GA相似的定长编码方式,表现型采用与GP相似的树结构,实现了简单编码表示复杂结构。基因型与表现型之间可相互转换,最后表现型转换为表达式。

GEP在实际应用中尤其是符号回归方面表现得相当出色,其独特的线性编码方式实现了编码长度的固定和统一,可以方便地进行选择、交叉、变异等遗传操作,同时依照其编码规则又可以解析为树状形态,克服了传统遗传算法在表示方法上的局限和传统遗传程序设计复杂的解构造和复杂的遗传操作的缺点,具有更简单的编码表示方法、更强大的遗传算子(遗传算子的种类和数量不存在任何限制)、更易于遗传操作、能产生更高级别的复杂度的能力。然而,从我们在符号回归分析和预测方面的研究结果[4-6]可以看出,虽然基本的GEP算法可以得到较好的结果,但是仍然存在早熟收敛的问题。本文在基本GEP的基础上,将分布评估学习算法(EDA)的思想引入GEP,以增加其学习能力,提高其收敛速度;同时采用多种群协同进化策略和总最优个体的迁移策略[7,8],在MPI环境下实现了同步和异步并行GEGEP算法,以此来提高算法的均衡搜索能力,进一步优化GEP的性能。实验表明,该并行算法提高了运行速度和解质量,具有线性加速比。最后对同步和异步分布式并行算法进行了对比实验分析。

2 相关工作

GEP是一种基于基因型和表现型的新型演化算法。它采用独特的线性编码方式实现了编码长度的固定和统一,且能表示复杂结构,这种特点使得GEP的性能比传统GP更好[9,10]。通常,为了提高GEP的性能,许多智能方法被引入GEP,彭京[11]、段磊[12]等将生物启发技术引入GEP中。为提

到稿日期:2009-03-19 返修日期:2009-05-25 本文受高等学校博士点基金(No. 20070486081),湖北省杰出青年基金(No. 2005ABB017)资助。
杜 欣(1979—),女,博士生,CCF会员,主要研究领域为智能计算,E-mail:xindu79@126.com;丁立新(1967—),男,教授,博士生导师,主要研究领域为智能计算、智能信息处理;谢承旺(1974—),男,博士生,主要研究领域为智能计算、数据挖掘和分布式计算技术;陈 莉(1976—),女,博士生,主要研究领域为智能计算。

高 GEP 的学习能力，杜欣[6]将学习算法引入 GEP 中。模拟退火算法是一种启发式优化算法，蒋思伟[13]将模拟退火的思想引入 GEP 来提高算法跳出局部最优的能力。GEP 和其他演化算法一样，本身具有并行性，蒋思伟[13]提出了一种基于模拟退火的并行基因表达式程序设计方法。

3 基于 GEGEP 的并行算法

3.1 GEGEP 算法

GEP 的基因型是由头部和尾部(具有一定约束关系)两部分构成的具有固定长度的串，表现型则为树型结构。基因型若要转换成表现型，则按从左到右、从上到下的方式从基因型编码中按各个函数所带参数数目读取字符来构建表现型，最后表现型可按中序遍历的方式转换为表达式，图 1 给出了三者之间的转换关系。

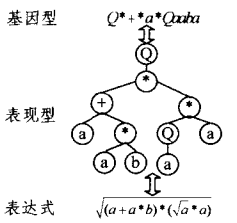

图 1 基因型、表现型与表达式之间的转换关系

QGEP 算法的搜索能力非常强，基因型向表现型转换时可产生丰富的变换结构，每个基因位置的变异都有可能导致树形结构发生很大的变化，每一次杂交或转座操作也会造成结构发生变化，因此在搜索过程中有一些较好的结构无法保留下来，从而降低了收敛速度。为提高标准 GEP 的收敛速度，增加 GEP 的学习能力，对基因结构进行了改进，在原来的“头＋尾”基因结构的基础上，提出了“头＋身＋尾”三段基因结构[14]。此结构有利于引入学习机制。在此结构的基础上，又引入了 EDA 的思想，提出了基因评估基因表达式程序设计方法(GEGEP)[6]，具体算法如下：

Step1 随机生成初始化种群 $P(0)$，$t\leftarrow 0$，计算 $P(0)$中每个个体的适应值，将具有最大适应值的个体保留下来；

Step2 变异：根据分布评估概率表对 $P(t)$中每个个体以变异概率 $p_m(0<p_m<1)$执行变异操作；

Step3 重组：对 $P(t)$中每个个体以重组概率 $p_c(0<p_c<1)$执行重组操作；

Step4 转座：对 $P(t)$中每个个体以转座概率 $p_t(0<p_t<1)$执行转座操作；

Step5 计算 $P(t)$中每个个体的适应值；

Step6 选择：基于保留最优个体策略从 $P(t)$中选择生成下一代种群 $P(t+1)$；

Step7 $t\leftarrow t+1$；

Step8 若满足停止准则，则结束；否则，转 Step2。

EDA[15]属于连锁学习算法中的一种，其学习功能是该算法一个十分重要的特点。EDA 主要有 3 种模型，分别为变量无关、两变量相关和多变量相关。由于引入统计学习会给算法带来新的时间和空间开销，因此本算法采用的是变量无关 EDA 模型。对普通基因和同源基因的头部和身部的每一位上的运算符进行统计，求出其统计概率。变异时根据其分布评估概率表中的概率进行变异，即当 $k\in H$，G_i 的第 k 位为算子 ω，则以概率 p_k^{ω} 进行变异，否则进行随机变异。每演化一代更新一次分布评估概率表，概率表如表 1 所列。

表 1 分布评估概率表

ω	j				
	1	2	3	4	5
+	P_1^+	P_2^+	P_3^+	P_4^+	P_5^+
−	P_1^-	P_2^-	P_3^-	P_4^-	P_5^-
*	P_1^*	P_2^*	P_3^*	P_4^*	P_5^*
/	$P_1^/$	$P_2^/$	$P_3^/$	$P_4^/$	$P_5^/$

下面以单基因为例说明概率表的更新方法。设 $h=2$，$b=3$，则统计部分(基因的头部和身部)的序号 $H=\{1,2,3,4,5\}$，函数集合 $F=\{+,-,*,/\}$，$\omega\in F$，记染色体 G_i 的适应值为 f_i，则有 $p_i=\frac{f_i}{\sum_{j=1}^{n}f_j}$，故 $\sum_{i=1}^{n}p_i=1$。

定义 $z_{ij}^{\omega}=\begin{cases}1, & \text{若 } G_i \text{ 的第 } j \text{ 位为算子 } \omega\\ 0, & \text{否则}\end{cases}$，则

$p_j^{\omega}=\sum_{i=1}^{n}Z_{ij}^{\omega}\cdot p_i$，$j\in H$，$\omega\in F$，显然 $\sum_{i=1}^{n}p_i^{\omega}=1$。

3.2 基于 GEGEP 的异步并行算法

在 GEGEP 算法的基础上，采用粗粒度的异步分布式并行模型，全互连迁移拓扑，以便迁移个体可以同时迁移到其他各区域中，提出了基于 MPI 的 GEGEP 异步分布式并行算法。具体算法如下：

Step1 初始化生成 P 个进程，每个处理器一个进程，进程的 ID 号分别为 $0,1,\cdots,P-1$；

Step2 处理器 0 读入数据文件、控制参数(包括种群规模 n、最大演化代数 *generation*、迁移代频等)，并将这些消息发送到其它各进程；

Step3 各处理器独立地进行演化，包括种群的初始化，更新统计评估表，执行变异、重组、转座遗传操作，计算种群中各个体的适应值；

Step4 每隔 g 代将当前种群中的最好个体发送到其它各个处理器(在原种群中仍保留其副本)；

Step5 每代检查有无其它处理器发送来的个体，若有，则用接收到的个体替换当前种群中的最差个体；若没有，则继续执行下面的操作；

Step6 如果到达最大演化代数，则进程 0 发送终止信息给别的 $P-1$ 个进程，程序终止；否则，转到 Step3。

3.3 基于 GEGEP 的同步并行算法

Step1 初始化生成 P 个进程，每个处理器一个进程，进程的 ID 号分别为 $0,1,\cdots,P-1$；

Step2 处理器 0 读入建模的数据文件、控制参数(包括种群规模 n、最大演化代数 *generation*、迁移代频等)，并将这些消息发送到其它各进程；

Step3 各处理器独立地进行演化，包括种群的初始化，更新统计评估表，执行变异、重组、转座遗传操作，计算种群中各个体的适应值；

Step4 搜索各子种群中的最优个体和最差个体，并记忆当前最优个体；

Step5 每隔 g 代各处理器发送当前最优个体到处理器 0，处理器 0 接收其它各处理器的当前最优个体并比较各节点的最优个体，得到总最优个体所在节点 max；

Step6 对于各个处理器 m，若 $m==\max$，发送处理器

m 的最优个体；否则，接收处理器 max 的最优个体，并替换掉最差个体；

Step7　若满足终止条件，则退出；否则，转 Step3。

4　实验和结果

4.1　实验设置

实验环境是由 20 台 PIV的微机通过 10Mbps 的以太网互联而成的分布式并行环境，操作系统采用 WINXP，用 MPI1.2.4[16] 编程实现。用并行加速比(S_p)、平均拟合误差(AFE)、平均适应值(AF)来评估并行算法的性能，其中解的好坏主要是通过 AFE 和 AF 来评价，AFE 的计算为：

$$AFE=\sum_{i=1}^{m}\sqrt{\sum_{j=1}^{n}(\bar{y}_j-y_j)^2}/m \tag{1}$$

其中，m 为独立运行次数，n 为数据对个数，显然由式(1)知，拟合误差越小，解越好；适应值越大，解越好。

为便于比较，选取文献[6]的瓦斯涌出量建模数据作为测试集，参数设置如表 2 所列。通过 100 次运行来比较 GEGEP、异步并行 GEGEP 和同步并行 GEGEP 算法的性能。本文所采用的适应值函数为基于相对误差的适应值函数：

$$f_i=\max(1000-\sum_{j=1}^{n}(|\frac{C_{ij}-T_j}{T_j}|\cdot factor),0) \tag{2}$$

其中，n 为样本数，M 用来控制适应值的范围，C_{ij} 表示第 i 个个体对应的函数表达式针对第 j 个样本的计算值，T_j 是第 j 个样本的目标函数值，$factor$ 为比例因子常数，设为 M/记录数，M=1000。显然，如果适应值越大，则表示个体越好。

表 2　参数设置

种群大小	40
进程数	20
基因头长	1
基因身长	10
同源基因头长	6
同源基因身长	20
基因数	8
单点重组概率	0.3
两点重组概率	0.3
变异概率	0.044
RIS 变换概率	0.1
IS 变换概率	0.1

4.2　串行结果与并行结果比较

我们比较了运行 100 次后的串行算法和并行算法的运行结果。其中并行算法得到的最好模型为 Y=Exp((Exp(((Sin(Ln(c))*(Exp(b)−Sqrt(c*f)))/((((d/Tan(f*e/b))/Cos(f−Tan(e^2*d−e+a)))+f−Tan(d))))/Cos(Sin(((((((d/Tan(f*e/b))/Cos((Tan(Ln(c))−Ln(c))))−f+Tan(d))*Ln(c))/(f−Tan(e^2*d−e+a)))))))，适应值=987.5432，FE=0.226017，运行时间为 114.2s(约 2min)，并行参数设置为 s=100，$generation$=5000，进程数=20，其拟合与预测曲线如图 2 所示。串行算法得到的最好模型为 Y=Ln(Exp((Exp(Cos((Cos(((Cos(Sqrt(Exp(Sin(Tan(Sqrt(a+c)+Sqrt$f$)))))−Exp$b$))/Sqrt(Sqrt(Sqrt(Ln(((Exp$c$−Tan$d$)/Sqrt(Exp(Sin(Tan(Sqrt(a+c)+Sqrt$f$)))))))))))+Ln((((Cos(Exp(Tan(d−b)))+Exp$b$)−Sqrt(Exp(Sin(Tan(Sqrt(a+c)+Sqrt$f$)))))/Tan(LnLnd))))))，适应值=987.1156，FE=0.218814，运行时间为 1124s(约 20min)，其拟合与预测曲线如图 3 所示。

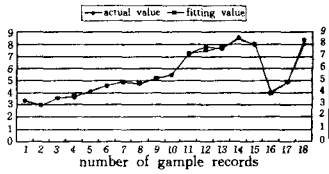

图 2　并行算法得到的最好模型的拟合与预测曲线

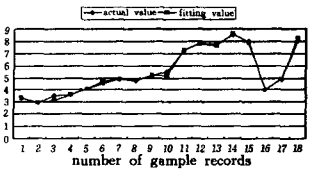

图 3　串行算法得到的最好模型的拟合与预测曲线

由以上结果可知，采用并行算法后运行时间大大减少(减少了 9/10)，搜索到的模型精确度与串行模型相差无几，其拟合与预测曲线和实际数据的仿真曲线几乎重合。

4.3　同步并行和异步并行算法比较

通过并行加速比(S_p)、平均拟合误差(AFE)、平均适应值(AF)来比较我们设计的基于 GEGEP 的同步并行算法和异步并行算法的性能。

(1) 并行加速比的比较

S_p 为串行平均运行时间 $\bar{T}_1$ 与并行平均运行时间 $\bar{T}_p$ 之比，即 $S_p=\frac{\bar{T}_1}{\bar{T}_p}$。每组实验均独立运行 100 次，计算 $\bar{T}_1$ 和 $\bar{T}_p$ 求 S_p。其参数设置如表 3 所列。

表 3　参数设置

S-1	generation=1000，s=50
S-2	generation=1000，s=100
S-3	generation=1000，s=200
S-4	generation=5000，s=100
S-5	generation=5000，s=200
S-6	generation=5000，s=500

表 4 为基于 GEGEP 的同步和异步并行算法在不同的参数设置下的加速比测试结果。由表 4 可知，基于 GEGEP 的同步并行算法的加速比优于基于 GEGEP 的异步并行算法。由于基于 GEGEP 的异步并行算法采用的是任意两进程间均进行通信的方式，因此需花费大量的通信时间。

表 4　加速比测试结果

	基于 GEGEP 的同步分布式并行算法	基于 GEGEP 的异步分布式并行算法
S-1	12.34	9.04
S-2	17.5	16.1
S-3	19.87	17.22
S-4	12.64	6.78
S-5	14.46	12.78
S-6	16.02	14.25

(2)AFE 和 AF 的比较

由图 4、图 5 可知，异步并行算法的平均适应值优于同步并行算法，且具有更好的运行结果。然而从花费的时间来看，同步并行算法花费时间更少。

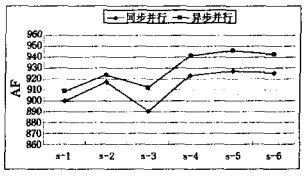

图 4　基于 GEGEP 的同步并行与异步并行算法的 AF 的比较

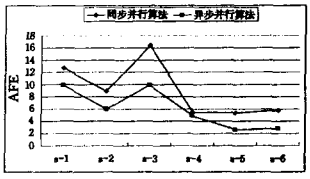

图 5　基于 GEGEP 的同步并行与异步并行算法的 AFE 的比较

4.4　并行控制参数算法的影响

这部分实验主要考察迁移代频对并行算法的影响，变量 s 表示迁移代频，变量 *generation* 表示演化代数，这里 *generation*=5000，参数设置如表 5 所列。

表 5　参数设置

S-1	s=50
S-2	s=100
S-3	s=200
S-4	s=500

由图 6、图 7 知，迁移代频越大，即每次迁移相隔的代数越长，则无论对同步并行算法还是异步并行算法来说，花费的运行时间都越少；对于 AF，同步并行算法和异步并行算法都是在迁移代频 s=200 时达到最好，因此迁移代频不宜设置太频繁。

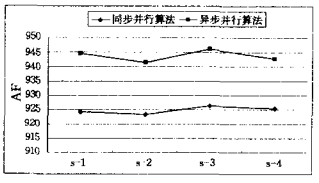

图 6　迁移代频对平均适应值的影响

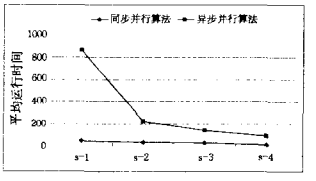

图 7　迁移代频对平均运行时间的影响

结束语　本文将 EDA 和多种群策略引入 GEP 中，提出了基于 GEGEP 的同步及异步并行算法。此并行算法提高了搜索能力，减少了计算时间，取得了较好的效果。

(1)EDA 是一种很好的学习算法，有利于提高收敛速度，多种群策略有利于跳出局部最优，个体迁移策略有利于增加种群多样性；

(2)并行算法与串行算法相比，并行算法获得了较好的解；

(3)同步并行算法和异步并行算法相比，同步并行算法获得了线性加速比，具有较好的解。异步并行算法虽没有获得线性加速比，但平均适应值、最好适应值和拟合误差的结果都优于同步并行算法。然而，同步并行算法的运行时间明显优于异步并行算法。

参 考 文 献

[1] Ferreira C. Gene Expression Programming: A New Adaptive Algorithm for Solving Problems[J]. Complex Systems, 2001, 13(2): 87-129

[2] Holland J H. Adaptation in Natural and Artificial Systems[M]. Cambridge: MIT Press, 1992

[3] Koza J R. Genetic Programming: On the Programming of Computers by Natural Selection[M]. Cambridge: MIT Press, 1992

[4] 杜欣，刘坤起，康立山，等. M-GEP-GA：基于多层染色体基因表达式程序设计的混合遗传进化算法[J]. 计算机应用，2007，27(4)：956-959

[5] 谢大同，康立山，李悦乔，等. 符号回归的一种新算法[J]. 系统仿真学报，2007，8(19)：1667-1671

[6] 杜欣，康立山，李悦乔，等. 基因评估基因表达式程序设计方法[J]. 小型微型计算机系统，2007，5(28)：834-840

[7] 曹宏庆，康立山，陈毓屏，等. 常微分方程组并行演化建模的实验研究[J]. 软件学报，2003，3(14)：443-450

[8] 管宇，徐宝文. 基于模式迁移策略的并行遗传算法[J]. 计算机学报，2003，26(3)：294-301

[9] Ferreira C. Gene expression programming: A new adaptive algorithm for solving problems[J]. Complex Systems, 2001, 13(2): 87-129

[10] Ferreira C. Mutation, transposition, and recombination: An analysis of the evolutionary dynamics[C]// Proceedings of the 6th Joint Conference on Information Sciences. USA, 2002

[11] 彭京，唐常杰，李川，等. M-GEP：基于多层染色体基因表达式编程的遗传进化算法[J]. 计算机学报，2005，9(28)：1459-1466

[12] 段磊，唐常杰，左劼，等. 基于基因表达式编程的抗噪声数据的函数挖掘方法[J]. 计算机研究与发展，2004，41(10)：1684-1689

[13] 蒋思伟，蔡之华，曾丹，等. 基于模拟退火的并行基因表达式算法研究[J]. 电子学报，2005，33(11)：2017-2021

[14] 杜欣，刘坤起，陈玉军. 改进的基因表达式程序设计实现复杂函数的自动建模[J]. 微计算机信息，2006(9)

[15] Larranaga P, Lozano J A. Estimation of distribution algorithms: A new tool for evolutionary computation[M]. Kluwer Academic Publishers, 2001

[16] 都志辉. 高性能计算并行编程技术-MPI 并行程序设计[M]. 北京：清华大学出版社，2001

(上接第 182 页)

表 1　模型同步时间表

节点个数		同步事件数			
		2	5	10	20
12	8	<1ms	<1ms	—	—
20	30	34ms	46ms	62ms	—
50	50	453ms	469ms	281ms	297ms
80	120	488ms	503ms	329ms	312ms

结束语　本文以 Petri 网为建模工具，对离散事件系统进行诊断，提出同步扩展模型的方法，解决了建模规模和诊断完备之间的冲突。通过实验分别得出了同步事件数与诊断节点数对同步过程的影响。

参 考 文 献

[1] Yannick P. Diagnosability analysis of distributed discrete event systems[C]// European Conference on Artificial Intelligence. Valencia, Spain, 2002

[2] Fabre E, Benveniste A, Haar S, et al. Distributed Monitoring of Concurrent and Asynchronous Systems[J]. Discrete Event Dynamic Systems, 2005, 15(1): 33

[3] Anika S, Yannick P. Scalable Diagnosability Checking of Event-driven Systems[C]// JCAI. Hyderabad, India, 2007

[4] Jussi R. Diagnosers and Diagnosability of Succinct Transition Systems[C]// IJCAI. Hyderabad, India, 2007

[5] Console Luca, Picardi Claudia, Dupre D T. A Framework for Decentralized Qualitative Model-Based Diagnosis[C]// IJCAI. Hyderabad, India

[6] Jiroveanu G, Boel K R. Petri Net model-based distributed diagnosis for large interacting systems [C] // 16th International Workshop on Principles of Diagnosis. Pacific Grove, CA, USA, 2005

[7] Petri C A. Fundamentals of a theory of asynchronous information flow[C]// IFIP Congress. Amsterdam, North Holland, 1963

[8] Cassez F, Roux O H. From Time Petri Nets to Timed Automata [J]. Journal of Systems and Software, 2006, 79(10): 1456

[9] 袁崇义. 佩特里网(1982 年第一版)[M]. 南京：东南大学出版社，1989：239

[10] Cordier M O, Grastien A. Exploring independence in a decentralized Approach of Diagnosis[C]// IJCAI. Hyderabad, India, 2007

[11] Sampath M, Sengupta R, Lafortune S, et al. Diagnosability of Discrete-event Systems[J]. Automatic Control, 1995, 40(9): 1555

[12] Sampath M, Sengupta R, Lafortune S, et al. Failure Diagnosis Using Discrete-event Models[J]. Control Systems Technology, 1996, 4(2): 105, 2007