

文章编号: 1000-6788(2002) 02-0015-09

并行遗传算法的新进展

郭彤城, 慕春棣

(清华大学自动化系, 北京 100084)

摘要: 并行遗传算法将并行计算机的高速并行性和遗传算法固有的并行性相结合, 极大地提升了遗传算法的求解速度和质量. 在主从式、细粒度和粗粒度这三类遗传算法并行化模型中, 粗粒度模型以其较小的通讯开销和对种群多样化, 获得了最广泛的应用. 本文概括了基于模式定理和有限状态马尔可夫链的遗传算法理论, 总结了前人在粗粒度模型下开展的理论分析和实践应用, 并指出并行遗传算法的研究将向异步化、理论化和模型化的方向发展, 而有限状态马尔可夫链是构建并行遗传算法可执行模型的有力工具.

关键词: 遗传算法; 并行计算; 粗粒度; 有限状态马尔可夫模型

中图分类号: TP18

文献标识码: A (20)

The Parallel Drifts of Genetic Algorithms

GUO Tong-cheng, MU Chun-di

(Department of Automation, Tsinghua University, Beijing 100084, China)

Abstract: The parallel GAs (PGAs) combine the high-speed parallel-ability of supercomputers with the inherent parallelity of GAs, and improve greatly the efficiency and accuracy of GAs. Among the master-slave, fine-grained and coarse grained parallel avenues, the coarse-grained model is most widely used for its little communication overhead and its diversifying of the population. In this paper, the schema theory and the model based on the limit Markov chain are generalized, the previous analysis and implementation on the coarse grained model are reviewed. It is shown that researches of PGAs will focus on asynchronization, theorization and modelization. Furthermore, the theory of limit Markov chain provides powerful tools to construct executable models of PGAs.

Keywords: genetic algorithms; parallel computation; coarse grain; limit Markov chain

1 引言

组合优化和函数优化需要在复杂庞大的搜索空间中寻找最优解或次最优解(满意解), 是存在于各学科中的普遍难题. 遗传算法(Genetic Algorithms, GAs)从一组初始可行解出发, 在不需要除目标函数值外的其它信息的条件下实现对可行域的全局高效搜索, 并以概率 1 收敛到全局最优解^[1]. 这种良好的特性使遗传算法成为组合优化和函数优化的有利工具, 并成为计算智能(computational intelligence)领域的研究热点.

随着科学技术的不断发展, 问题的规模不断扩大, 面对复杂程度越来越高的搜索空间遗传算法在优化效率(时间)和求解质量上都显得“力不从心”. 并行遗传算法将并行计算机的高速并行性和遗传算法天然的并行性相结合, 极大地促进了遗传算法的研究与应用. 并行处理的引入不但加速了遗传算法的搜索过

(20) 收稿日期: 2000-06-16

资助项目: 863应用基础研究基金(863-511-943-014)

程,而且由于种群规模的扩大和各子种群的隔离,使种群的多样性得以丰富和保持,减少了未成熟收敛的可能性,提高了求解质量。

2 遗传算法的运行机理

对遗传算法运行机理的解释有两类:一是传统的模式理论;二是1990年以后发展起来的有限状态马尔可夫链模型。

2.1 模式理论

模式理论由Holland创建^[2],主要包括模式定理、隐并行性原理和积木块假说三部分^[3,4]。模式是可行域中某些特定位置取固定值的所有编码的集合。模式理论认为遗传算法实质上是模式的运算,编码的字母表越短,算法处理一代群体时隐含处理的模式就越多。当算法采用二进制编码时,效率最高,处理规模为 N 的一代群体时,可同时处理 $O(N^3)$ 个模式^[5]。遗传算法这种以计算少量编码适应度而处理大量模式的性质称为隐并行性。模式理论还指出,目标函数通常满足积木块假说,即阶数高、长度长、平均适应度高的模式可以由阶数低、长度短、平均适应度高的模式(积木块)在遗传算子的作用下,接合而生成^[6]。而不满足积木块假说的优化问题被称为骗问题(deceptive problem)^[7]。模式理论为遗传算法构造了一条通过在群体中不断积累、拼接积木块以达到全局最优解的寻优之路。

近十多年的研究,特别是实数编码遗传算法的广泛应用^[8-11]表明,上述理论与事实不符。

1) 首先,只有在线性空间,或满足与全局最优解之间的“距离”越小则适应度越高的单极值(one-max)空间才满足阶数低、长度短、平均适应度高的模式(积木块)可拼接成阶数高、长度长、平均适应度高的模式,甚至全局最优解。事实也表明,只有搜索空间为单极值空间(one-max)时,模式理论才能够准确地预测搜索过程^[12]。但事实上线性空间和单极值空间的优化问题可以非常方便地利用梯度算法解决。而需要利用GA搜索的空间几乎全部是非线性、多极值(multiple-optima, multiple-modal)空间,即GA绝大多数情况下处理的是骗问题。可见模式理论用只适合于线性、单极值空间的思想去解释非线性、多极值空间中的现象是不严谨的。

2) 由于骗问题的广泛存在,利用模式理论无法证明GA的全局收敛性。

2.2 有限状态马尔可夫链模型

由于模式理论的种种缺陷,从80年代末开始,研究者开始尝试利用有限状态马尔可夫链模型研究遗传算法的运行过程。尤其是Gunter Rudolph^[1]在1994利用有限状态马尔可夫链模型证明了只要对标准遗传算法稍作改进,在循环过程中保存搜索到的适应度最高的染色体,那么无论搜索空间多么复杂,在循环次数趋于无穷时,保证遗传算法以概率1收敛到全局最优解。

对于遗传算法可以解决的优化问题,问题的可行域都是由有限个点组成的。即便是参数可以连续取值的问题,实际上搜索空间也是以要求精度为单位的离散空间。因此遗传算法的实际运行过程可以用有限状态马尔可夫链的状态转移过程建模和描述。

对于有 m 个可行解的目标函数和群体规模为 N 的遗传算法, N 个个体共有 $\begin{Bmatrix} N+m-1 \\ m-1 \end{Bmatrix}$ 种组合,相应的马尔可夫模型也有 $\begin{Bmatrix} N+m-1 \\ m-1 \end{Bmatrix}$ 个状态^[1]。实际优化问题的可行解数量 m 和群体规模 N 都十分可观,马尔可夫模型的状态数几乎为天文数字,因此利用精确的马尔可夫模型计算群体的状态分布是不可能的。

为了换取模型的可执行性,必须对实际模型采取近似简化,保持算法的实际形态,通过对目标函数建模,简化目标函数结构实现模型的可执行性。UIUC的Mahfoud博士提出了一种基于可行域等价划分的目标函数模型^[13]。优化问题的可行域可划分成若干个等价类,每个等价类包含且只包含一个局部极值点,并且该极值点是等价类中函数值中最大(最小)的点,利用这些极值点代表其所属等价类中的所有点(候选解),即各可行域中所有点的函数值均近似为所属等价类所对应的局部极值,并假定算法总能正确判定各点所属的等价类。北京理工大学的邹燕明博士对上述模型进行了改进^[14],他把任意两点间距离近似为其

所属等价类所对应的局部极值点的距离,以进一步讨论算法判断和选择不正确时的性能,拓宽了模型的适用范围.这样,群体中个体间的竞争被近似抽象成以邻域结构为依据划分成的等价类之间的竞争.

遗传算法优化的过程,可以看作算法在循环过程中不断对可行域进行随机抽样,利用前面抽样的结果对目标点的概率分布进行估计,然后根据估计出的分布推算下一次的抽样点.模式理论将包含相同模式的个体集合视为等价类,并进一步认为,对模式的抽样结果提供了对整个等价类的信息,可用于估计等价类的适应度;相反,没有处理任何不包含该编码的模式,也就绝对没有获得相应模式的任何信息.但非常明显的是对一个个体的抽样结果,可用于对个体的邻域进行较精确的估计,但提供给与抽样点“距离”较大的个体的信息很小,尤其是在多极值空间中,即抽样结果对等价类(模式理论意义下)中个体的参考价值是不同的,对距离近的个体参考价值大,距离远的个体参考价值小.对于处于不同极值点邻域内的个体,甚至提供的是噪声,这个矛盾很难利用模式理论作出解释.而马尔可夫模型认为遗传算法是通过搜索空间不同区域的抽样,来估计不同区域的适应度,进而估计最优解存在于不同区域的概率,以调整算法对不同区域的抽样密度和搜索力度,进而不断提高对最优解估计的准确程度.

可见,以邻域结构为依据划分等价类的马尔可夫模型更符合实际,对问题的抽象更能体现优化问题的本质.

3 粗粒度 (coarse-grain)并行遗传算法

遗传算法的并行模型可分为三类^[15]:主从式模型、细粒度模型和粗粒度模型.主从式模型将选择、交叉、变异等全局操作交由主处理器(master)串行执行,而将适应度评估等局部操作交予从处理器网络(slaves)并行执行.由于未对串行遗传算法的框架进行改动,所以主从式模型不可避免的存在着主、从节点负荷忙闲不均衡的问题.而且一次局部操作完成后,从节点都要向主节点发送结果,造成瓶颈和大的通信延迟,所以运行效率不高,一般只用于适应度评估的工作量很大的情况^[16-17].

当并行计算机系统的规模很大,处理器多到可以与染色体一一对应,即每一处理器上驻扎一个染色体时,可以采用细粒度模型的并行遗传算法.对于每个染色体,选择操作和交叉操作都只在其所处的处理器及其邻域中进行,这样的并行模型无需或只需很少的全局控制,在最大的限度上发挥了遗传算法的并行潜力^[18-20].由于对处理器数量的要求很高,所以细粒度模型的应用范围不广,一般只运行于大规模的SIM D(single instruction multiple datastream)并行计算机.

粗粒度模型又称分布式模型^[21](distributed style)或孤岛模型^[22,23](island-based model),是适应性最强和应用最广的遗传算法并行化模型.粗粒度模型是将随机生成的初始群体依处理器个数分割成若干个子群体.各个子群体在不同的处理器上相互独立的并发执行进化操作,每经过一定的进化代,各子群体间会交换若干的个体以引入其它子群体的优秀基因,丰富各子群体的多样性,防止未成熟收敛的发生^[24].

粗粒度模型的通信开销较小,可获得接近线性的加速比,而且非常适合运行在通信带宽较低的集群系统^[25]上.

3.1 子群体的初始化

各子群体的初始分布可以同为随机分布在可行域中,不论是在一个处理器(主节点 host)上产生一个大的群体以后,再随机分配到各个节点上^[26];还是各个节点独立的随机生成自己的初始群^[27-29].另一种情况,各子群体的初始群体分布在可行域的不同区域中^[30-32].比较而言,后者的效果较好.

3.2 连接拓扑

各子群体间的连接拓扑包括完全隔离(无迁移)^[33,34],单向环^[35],双向环^[29],超立方体^[36],网格^[37]等等.文献[32]利用大量的实验比较了各种拓扑连接对解质量的影响,发现单向环拓扑既保证了优良基因在群体间的扩散,又较好地隔离了子群体,保护了群体间的多样性,虽然收敛速度较慢,但解的质量较高.

为了克服收敛速度和解质量之间的矛盾,文献[32]提出了一种圆锥形的连接拓扑(图1):每间隔一段时期,圆锥中处于同一层次中的小群体间迁移随机挑选的个体;同时,向上层传递最佳个体.不但保持层内的多样性,而且加强了层间的进化压力,效果良好.

3.3 粗粒度模型的迁移策略

在粗粒度模型的各种实现方法中,有的选择子群体中的最优个体向外“移民”^[22,29,38];有的“移民”则是随机选出的^[39];有的算法用迁入者取代群体中的最差个体^[28,35];有的被取代者则是随机选出的^[38,39]。利用最优个体迁出/最差个体被取代(本文称其为最优/最差准则)是最常用的方法^[28,35],但对特定问题,有时随机选择迁出个体或随机被取代个体的效果可能更好^[34]。

大多数情况下,迁出者只是被复制到其它的子群体,但有的算法^[27]利用迁入者取代迁出者,即将迁出者在本地删除。

1) 确定迁移间隔(同步迁移)

在最初的粗粒度遗传算法中,个体迁移的间隔是固定的。这时,利用消息传递迁移个体前,各进程间要进行同步,以确保各子群体进化了相同的代数。实验结果表明,间隔过大,会未成熟收敛(收敛到次优解),虽然收敛快,但解的质量不高;间隔过小,使子群体之间的多样性被破坏,在可接受的时间内依然得不到高质量的解^[22,33,40]。所以迁移间隔要视具体问题而定。

2) 非确定迁移间隔(异步迁移)

自然界中存在着这样的现象,当外部个体迁入稳定的局部环境时,会导致环境中个体的飞跃式发展。于是异步迁移的粗粒度并行遗传算法模型被提出来:当一个子群体的发展(经过一定的进化代)一直停滞不前,则向其它子群体发出申请——申请个体迁入;而当一个子群体的最佳个体的适应度提高后,向其邻居子群体发送这个最佳个体。每个处理器都为收发个体分配了缓冲区,开辟了通讯线程管理个体的收发。

文献[41]提出的 injection island GA(iiGA)和文献[42]提出的 Random Island Model(RIM)都属于异步迁移方式,它们都得到了良好的收敛速度和解质量,文献[43]利用 iiGA进行复合结构梁的优化设计,取得了比同步迁移粗粒度并行遗传算法更好的效果。

文献[44]注意到同样的自然现象,设计了 Hypergamous Parallel GA(图2),各子群体成树状的层次关系,各子群体的遗传操作中没有变异,这致使各子群体很快进入未成熟收敛状态,这时各子群体将各自的最佳个体汇集到上层的父亲节点中,如此循环直至树的根节点的进化也完全收敛。

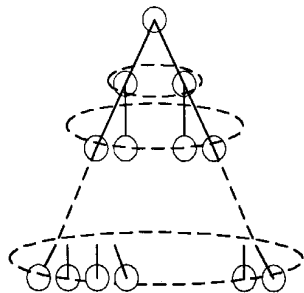


图1 圆锥形拓扑

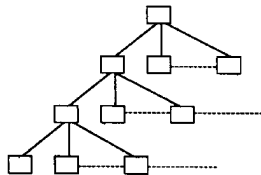


图2 Hypergamous Parallel GA

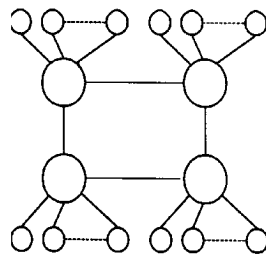


图3 粗粒度主从式

异步执行方式还使在 Internet上实现并行遗传算法成为可能,文献[45]完成了这样的工作:构成系统的处理节点限制在一定范围(如某个网关)以内。整个处理系统依据并行随机发动算法(parall random launch algorithm, PRLA)来组织,当一个处理节点收到启动请求和并行处理系统的状态信息后,将创建一个计算进程,然后随机的选择一个未启动的节点并发送启动请求;同时,计算进程忽略此后到达的启动请求。由于 Internet上各节点能提供的计算资源不统一,数据传输的时延也无法准确估计,所以为保证一个纪元(epoch,即一定次数的循环)结束后,新的纪元立刻开始,采用异步迁移策略。每个节点维护一个接收缓冲区,当节点完成一个纪元之后,将一部分个体发送到随机选出的一个接收缓冲区空的节点。然后用接收缓冲区的个体取代发送出的个体,开始下一个纪元;如果节点的接收缓冲区为空,则省略取代步骤,直接进入下一个纪元。当迁移目的地的接收缓冲区已满时,迁移的个体(数据包)将转向其它节点,算法随机

选择迁移的目的节点,避免了固定的拓扑连接关系.这样,无论是一次迁移失败(数据包在网上丢失),还是一个节点中途退出,都不会使系统瘫痪,极大地提高了系统的容错性能.此外为避免死锁和拥塞,算法还进行了特殊的优化.

Jens Liening 博士^[46]通过大量的实验发现:

1) 过多以及过频繁的迁移会破坏子群体的多样性,致使多个搜索进程集中到相同的区域,不利于提高解的质量;过少的迁移以及迁移频率过低,使各子群体不能充分利用其它子群体的信息,同样不利于提高解的质量.

2) 选择适应度最高的个体迁移,在短期内收敛快,解的质量提高快;但利用随机挑选的个体迁移,在一段时间后能得到质量更高的解.

3) 保持子群体的规模,增加子群体的数量,能够明显地提高解的质量.而且在保持总群体规模不变的前提下,一定范围内减小子群体规模,同时增加子群体的数量,也能提高解的质量.文献[22]也观察到了类似的结果.

3.4 混合模型并行遗传算法

将上述三种模型混合形成层次结构.在下层的并行模型中,子群体的规模是真实的,即为一个处理进程所处理的个体数量.而对于上层模型,将每个下层的并行结构都视为一个“集合子群体”,而“集合子群体”之间按上层的并行模型协调运行.无论是从下层还是从上层角度看,都是子群体(集合子群体)内部信息交互量大,子群体(集合子群体)之间信息交互量小.

上、下层模型的组合关系主要有三种^[12]:粗粒度-细粒度,粗粒度-粗粒度,粗粒度-主从式.

实际中应用较多是粗粒度-主从式模型^[47,48](图3).由于并行资源有限(处理器的个数有限),文献[48]下层的主从式依靠分布在不同处理器上的线程实现,即每个从节点上运行着为不同主节点服务的多个工人线程,但这种作法对提高搜索效率和质量并无明显的效果.

3.5 结合局部搜索的并行遗传算法

虽然遗传算法具有大范围的优化搜索能力,但对搜索空间中的局部邻域结构并不敏感.而很多其它的优化算子,如人工神经网络、快速下降法等等都属于梯度算法,能够保证收敛到局部极值点.通过引入交叉和变异以外的局部优化算子,将遗传算法大范围搜索与局部搜索相结合,能够有效的提高解的质量.

虽然局部优化搜索和并行机制的相互影响不大,但由于二者的结合效果能够获得更好的优化效果,所以被广泛用于主从式模型^[49,50]和粗粒度模型^[15,28,30].与并行遗传算法结合的局部优化方法多为针对问题特点设计的局部爬山(climbing hill)算法^[15,28,30,49,50].

局部优化的使用方法也有区别,文献[28]随机选择子群体的40%进行局部优化;文献[50]选择子群体中最好的20%进行局部优化;文献[15]只在子群体停止进化一段时期后才使用局部优化方法;文献[30,49]则对整个子群体进行局部优化.

但是局部优化不能过于“精细”,虽然文献[30]获得的解质量略有提高,但进化时间提高了近10倍,效率很低.

3.6 并行小生境遗传算法

在遗传算法中,小生境是指包含且只包含一个目标函数极值点的可行子区域.能够在一次搜索过程中得到目标函数多个极值点的遗传算法被称为小生境方法(niching methods).当前小生境方法实现上述能力的途径主要有两条:适应度调整和直接竞争限制.基于适应度调整的小生境方法还可进一步分为共享机制(Sharing)和清除机制(Clearing);而基于直接竞争限制的小生境方法,又称排挤机制^[13].

现有的结合并行技术与小生境技术的研究工作,都是在主从式(Master-Slave)的框架下展开的.文献[51]利用了主从式模型并行化共享机制,把适应度计算环节并行执行.文献[50]结合了局部优化(local optimization)和排挤(crowding)机制,在利用排挤机制进化完成一代以后,从中挑出一定比例的个体利用局部爬山(climbing hill)进行微调,然后开始下一代进化.由于局部爬山与其它个体无关,所以可以同适应度计算环节一起交予从节点(Slave)执行.由于文献[50]的初衷是寻找全局最优解而非全部极值点,所以

局部爬山对低适应度,小规模物种的维持性能不必关心,引入排挤的目的是维持群体中的多样性,防止未成熟收敛。

4 并行遗传算法的评价模型

有的文献^[12, 22, 32]利用 Amdahl定律^[53]评价并行遗传算法,即绝对加速比(speedup) S : $S = T_s / T_p$,其中, T_s 为串行遗传算法(单个处理器)的执行时间; T_p 为并行遗传算法的执行时间。

Amdahl定律适用于负载固定的情况,具体到并行遗传算法,适用于总群体规模不变的情况。所以对于主从式和细粒度模型 Amdahl定律是适用的,在适应度评价计算量较大时,主从式模型可以得到接近线性的加速比^[12]。由于细粒度模型的应用较少,适用的 SIMD并行机的可扩展性也不突出,所以很少有人评价细粒度模型的加速比。利用 Amdahl定律评价粗粒度模型时,需保持总的群体规模,即子群体数量和子群体规模成反比。这种情况下粗粒度模型的加速比接近线性,这是由于粗粒度模型的通信开销和同步开销都不大。

文献[47]还从实用的角度给出了两条定性评价并行遗传算法的指标:

1) 确定一个适应度指标,利用串行遗传算法最短的搜索时间(找到一个个体适应度高于适应度指标)除以并行遗传算法的搜索时间,作为并行遗传算法的加速比指标。在此指标下,并行遗传算法能够达到超线性加速比^[22, 47]。

2) 给定一段时间,利用并行遗传算法进行优化,能够得到的最高适应度比利用串行遗传算法搜索到的最高适应度高出多少。

第二条定性评价指标非常有实用价值,因为遗传算法的应用者需要的是在一定的时间内获得最好的优化效果^[47]。

5 并行遗传算法的可执行模型

在上世纪 90年代中期以前,虽然并行遗传算法被广泛应用于各种领域,但主要限于实践,为得到好的优化效果,研究者要进行大量的实验,试用各种策略和参数。从 1994年开始,UIUC 的 Goldberg 教授和他的学生 Cantu-Paz 博士着手设计了一种基于模式理论(及积木块假设)的并行遗传算法的可执行模型^[12]。

他们以一个由 20个 4位基因串联而成的染色体表示单极值(One-max)目标函数为例(函数的适应度等于染色体中被置为 1的比特数),建立了以基因为描述对象的模型:他们在分析过程中忽略了交叉和变异对积木块的构造和破坏作用,并将选择视为最优模式与其它模式的竞争。假设每个纪元结束时,子群体都收敛,即子群体中的个体已无法再进化后,子群体之间依照最优/最差准则进行同步迁移。Cantu-Paz 博士利用可靠性理论对模型进行了详细的分析(分析搜索不到全局最优点的概率),得到了一系列的结论:

- 1) 总群体的规模(各个分群体规模之和)越大,搜索的错误概率越小,需要进化的代数就越少。
- 2) 群体间的迁移率越大,搜索的效果越好。
- 3) 连接拓扑的度(子群体直接邻居的数量)越大,搜索的效果越好。

总之,并行遗传算法中子群体之间的隔离越弱,越象单一群体遗传算法(Simple GA with a aggregate population)性能越好。

文献[54]还利用实验验证了这个以基因为描述对象的马尔可夫模型可以较好地预测加性可分解函数(additively-decomposable function),即染色体由交迭子串组成,染色体的适应度等于交迭子串适应度的和,而子串的适应度函数可以是多极值的。

与前面文献[32, 34, 46]的实验结果相对比,可以发现 Cantu-Paz的理论分析结果^[12]与实际观察有着不小的偏差,这是由于 Cantu-Paz的研究分析是基于单极值函数的,而在实用中遗传算法几乎都是用于多极值的复杂空间搜索。即使是加性可分解函数,也与实际情况相差甚远。所以 Cantu-Paz关于迁移率越大,连接拓扑的度越大,则搜索的效果越好的结论与事实严重不符。可见依据 Cantu-Paz的并行遗传算法可执行模型很难准确地设置遗传参数和迁移策略。

6 结论

从上面的分析可以看出,并行遗传算法的发展呈现出三个特点:

1) 系统化和模型化,建立并行遗传算法可执行模型,利用可执行模型分析遗传算法的优化过程,指导并行方式和参数的设置.

2) 异步化,各处理器之间以异步方式进行通信,既节省了同步开销,又为在 Internet 的分布计算资源上实现并行遗传算法打下了理论基础.

3) 与局部优化算法相结合,有效地提高求解质量.

而设计能够较准确描述多极值的复杂搜索空间,但又能够较方便地进行演绎分析的模型成为了研究并行遗传算法的关键问题.

通过对遗传算法发展动态的了解,以及理论上和实践中尚存欠缺的分析,可以相信建立基于有限状态马尔可夫链的并行遗传算法的定量分析模型具有重要的理论意义.在这个依据目标函数空间邻域结构划分等价类,能够较精确地反映目标函数分布的模型基础上研究子群体的初始化策略和迁移策略以指导实践,具有良好的应用前景.

参考文献:

- [1] Rudolph G. Convergence analysis of canonical genetic algorithm [J]. IEEE Trans, 1994, 5(1): 96 - 101.
- [2] Holland J H. Adaptation in Natural and Artificial Systems [M]. Ann Arbor: Univ. Of Michigan Press, 1975.
- [3] Goldberg D E. Genetic algorithms in search, optimization and machine learning [M]. Reading, MA, Addison-Wesley, 1989.
- [4] 史忠植. 高级人工智能 [M]. 北京: 科学出版社, 1998.
- [5] Bertoni A, Dorigo M. Implicit parallelism in genetic algorithm [J]. Artificial Intelligence, 1993, 61: 307- 314.
- [6] Thierens D, Goldberg D E. Mixing in genetic algorithms [A]. Proc. 5th. Int. Conf. Genetic Algorithms[C]. Morgan Kaufmann, San Mateo, 1993. 38- 45.
- [7] Vose M D. Generalizing the notion of schema in genetic algorithms [J]. Artificial Intelligence, 1991, 50: 385- 396.
- [8] Hong J S. Genetic approach to bearing estimation with sensor location uncertainties [J]. Electronics Letters, 1993, 29(23): 2013- 2014.
- [9] McClurkin G D, Sharman K C, Durrani T S. Genetic algorithms for spatial spectral estimation [A]. 4th Annual ASSP Workshop on Spectrum Estimation and Modeling[C]. IEEE, NY, USA, 1988. 318- 322.
- [10] Yoshioka M, Omatu S. Signal separation method using genetic algorithm [A]. IEEE Int. Joint Conf. on NN [C]. 1998, 2: 909- 912.
- [11] Zou, Y M, Ke Y. A genetic algorithm with sharing and its application in MUSIC method [A]. Proc. ICSP 98[C], 393- 396.
- [12] Canú-Paz E. Designing efficient and accurate parallel genetic algorithms [EB/OL]. IlliGAL Report No. 99017 (http://www-illigal.ge.uiuc.edu).
- [13] Mahfoud S W. Niching Methods for Genetic Algorithms [EB/OL]. IlliGAL Report No. 95001. (http://www-illigal.ge.uiuc.edu) 1995.
- [14] 邹燕明. 小生境遗传算法的研究与应用 [D]. 北京: 北京理工大学, 1999.
- [15] Muhlenbein H, Schomisch M, Born J. The parallel genetic algorithm as function optimizer [J].

- Parallel Computing, 1991, 17(6-7): 619-632.
- [16] Garrett C A, Junqi Huang, Goltz M N, et al. Parallel real-valued genetic algorithms for bioremediation optimization of TCE-contaminated groundwater[A]. Proc. CEC99. IEEE[C]. Piscataway, NJ, 1999, Vol. 3, 2183-2189.
 - [17] Lohn J D, Colombano S P. A circuit representation technique for automated circuit design[J], IEEE Trans. on EC, 1999, 3(3): 205-219.
 - [18] Folino G, Pizzuti C, Spezzano G. Combining cellular genetic algorithms and local search for solving satisfiability problems[A]. Proc. 10th IEEE Int. Conf. on Tools with Artificial Intelligence [C], Piscataway, NJ, 1998. 192-198.
 - [19] Han M M, Tatsumi S, Kitamura Y, et al. Parallel genetic algorithms based on a multiprocessor system FIN and its application[J]. IEICE Trans. Fundam. Electron. Commun. Comput. Sci., 1995, E78A(11): 1595-1605.
 - [20] Shapiro B A, Navetta J. A massively-parallel genetic algorithm for RNA secondary structure prediction[J]. J Supercomput, 1994, 8(3): 195-207.
 - [21] Prahlada Rao B B, Hansdah R C. Extended distributed genetic algorithm for channel routing[A]. Proc. of 5th IEEE Symp. on Parallel and Distributed Processing[C]. Los Alamitos, CA, 1993. 726-733.
 - [22] Calegari P, Guidic F, Kuonen, P, et al. Parallel island-based genetic algorithm for radio network design[J]. J Parallel Distrib. Comput, 1997, 47(1): 86-90.
 - [23] Topping B H, Sziveri J, Bahreinejad A, et al. Parallel processing, neural networks and genetic algorithms[J]. Advances in Engineering Software, 1998, 29(10): 763-786.
 - [24] 周远晖, 陆玉昌, 石纯一. 基于克服过早收敛的自适应并行遗传算法[J]. 清华大学学报, 1998, 38(3): 93-95.
 - [25] Johnston W E. Rationale and strategy for a 21st century scientific computing architecture the case for using commercial symmetric multiprocessors as supercomputers[J]. Int. J. of high speed computing, 1997, 9(3): 191-222.
 - [26] da Fonseca Neto J V, Bottura CP. Parallel genetic algorithm fitness function team for eigenstructure assignment via LQR designs[A]. Proc. CEC99. IEEE[C]. Piscataway, NJ, 1999, Vol. 2, 1035-1042.
 - [27] Andre D, Koza J R. A parallel implementation of genetic programming that achieves super-linear performance[J]. Information Sciences, 1998, 106(3-4): 201-218.
 - [28] Easton F F, Mansour N. A distributed genetic algorithm for deterministic and stochastic labor scheduling problems[J]. European J. of Operational Research, 1999, 118(3): 505-523.
 - [29] Yang H T, Yang P C, Huang C L. A parallel genetic algorithm approach to solving the unit commitment problem implementation on the transputer networks[J]. IEEE Trans. Power Syst. 1997, 12(2): 661-668.
 - [30] Hidrobo F, Aguilar J. Toward a parallel genetic algorithm approach based on collective intelligence for combinatorial optimization problems[A]. Proc. CEC98[C]. IEEE, 1998. 715-720.
 - [31] Lee C Y, Kim S J. Parallel genetic algorithms for the earliness tardiness job scheduling problem with general penalty weights[J]. Comput. Ind. Eng, 1995, 28(2): 231-243.
 - [32] Matsumura T, Nakamura M, Okech J, et al. A parallel and distributed genetic algorithm on loosely-coupled multiprocessor system[J]. IEICE Trans. Fundam. Electron. Commun. Comput. Sci., 1998, E81A(4), 540-546.
 - [33] Matsumura T, Nakamura M, Miyazato D, et al. Effects of chromosome migration on a parallel

- and distributed genetic algorithm [A]. Proc. I- SPAN '97[C]. Los Alamitos, CA, 1997, 357-61.
- [34] Wang L, Maciejewski AA, and Siegel HJ. A comparative study of five parallel genetic algorithms using the traveling salesman problem[A]. Proc. of the 1st Merged Int. Parallel Processing Symp. and Symp. on Parallel and Distributed Processing[C]. Los Alamitos, CA, 1998, 345-349.
- [35] Beckers M L M, Derks EPPA, Melssen W J, et al. Using genetic algorithms for conformational analysis of biomacromolecules[J]. Comput. Chem., 1996, 20(4): 449-457.
- [36] Xu DXJ, Daley M L. Design of optimal digital-filter using a parallel genetic algorithm [J]. IEEE Trans. Circuits Syst. II- Analog Digit. Signal Process, 1995, 42(10): 673-675.
- [37] Fukuyama Y, Chiang H D. A parallel genetic algorithm for generation expansion planning [J]. IEEE Trans Power Syst, 1996, 11(2): 955-961.
- [38] Pan J S, McInnes F R, Jack M A. Application of parallel genetic algorithm and property of multiple global optima to VQ codevector index assignment for noisy channels[J]. Electron Lett, 1996, 32(4): 296-297.
- [39] Mayer M K. A network parallel genetic algorithm for the one machine sequencing problem [J]. Computers & mathematics with applications, 1999, 37(3), 71-78.
- [40] Alba E, Cotta C, Troya J M. Numerical and real time analysis of parallel distributed GAs with structured and panmictic populations[A]. Proc CEC99. IEEE Vol. 2[C]. Piscataway, NJ 1999. 1019-1026.
- [41] Shyh-Chang Lin, Punch W F III, Goodman E D. Coarse-grain parallel genetic algorithms categorization and new approach[A]. Proc. 6th IEEE Symp. on Parallel and Distributed Processing[C]. Los Alamitos, CA, 1994. 28-37.
- [42] Salhi A, Glaser H, De Roure D. Parallel implementation of a genetic-programming based tool for symbolic regression[J]. Information Processing Letters, 1998, 66(6): 299-307.
- [43] Punch W F, Averill R C, Goodman E D, et al. Optimal-design of laminated composite structures using coarse-grain parallel genetic algorithms [J]. Comput. Syst. Eng., 1994, 5(4-6): 415-423.
- [44] Evans I K. Embracing premature convergence the hypergamous parallel genetic algorithm [A]. Proc CEC98 IEEE[C]. NY, 1998, 621-626.
- [45] Joseph D, Kinsner W. Design of a parallel genetic algorithm for the internet [A]. IEEE WES-CAN EX 97 Communications, Power and Computing Conf Proc IEEE[C]. New York, NY, 1997. 333-343.
- [46] Lienig J. Institution Tanner Res. Inc, Pasadena, CA, USA. A parallel genetic algorithm for performance-driven VLSI routing [J]. IEEE Trans. on EC, 1997, 1(1): 29-39.
- [47] Han Yang Foo, Esbensen H, Ianjian Song, et al. Performance improvement of a genetic algorithm for floorplanning with parallel computing technology [A]. Proc. ISCAS '97, IEEE Vol. 3[C]. NY, 1997. 1544-1547.
- [48] Oussaidene M, Chopard B, Pictet O V, Tomassini M. Parallel genetic programming and its application to trading model induction[J]. Parallel Computing, 1997, 23(8): 1183-1198.
- [49] Chemishkian S, Arabyan A. Intelligent algorithms for optimal placement of actuators and sensors in structural control[A]. Proc American Control Conference IEEE Vol. 3[C]. Piscataway, NJ 1999. 1812-1816.

(下转第 41 页)

$$K = [318.9572, \quad 90.6578]$$

以初始条件 x_0 分别为 $(50^\circ, 0)$, $(60^\circ, 0)$, $(70^\circ, 0)$, $k = \sin(2^\circ t)$, 进行仿真, 仿真结果如图 1 所示.

5 结论

基于鲁棒 H_∞ 稳定性理论, 获得了一类渐近稳定的模糊系统模型. 并用一级倒立摆模型进行仿真验证, 结果表明控制方案简洁有效. T-S 模糊动态模型较容易和其他先进控制方案结合, 是很有价值的研究方向. 本文对应用 H_∞ 控制理论解决非线性问题进行了有益的尝试.

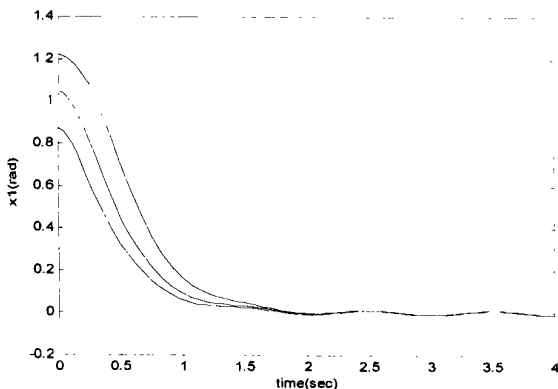


图 1 系统仿真曲线

参考文献:

- [1] Zames G. Feedback and optimal sensitivity: model reference transfer functions, multiplicative semi-norms, and approximate inverses[J]. IEEE Trans Autom Control, 1981, AC-26(2): 301-320.
- [2] Van der Schaft A J. On a state space approach to nonlinear H_∞ control[J]. Syst Control Lett., 1991, 16(1): 1-8.
- [3] Van der Schaft A J. L_2 -gain analysis of nonlinear systems and nonlinear state feedback H_∞ control[J]. IEEE Trans Autom Control, 1992, AC-37(3): 770-784.
- [4] 申铁龙. H_∞ 控制理论与应用 [M]. 北京: 清华大学出版社, 1996.
- [5] Takagi T, Sugeno. Fuzzy identification of systems and its applications to modeling and control[J]. IEEE Trans Syst Man Cybern, 1985, 15(1): 116-132.
- [6] Wang H O, Tanaka K, Griffin M F. An approach to fuzzy control of nonlinear systems: stability and design issues[J]. IEEE Trans Fuzzy Syst, 1996, 4(1): 14-23.

(上接第 23 页)

- [50] Huntley C L, Brown D E. Parallel genetic algorithms with local search[J]. Comput Oper Res, 1996, 23(6): 559-571.
- [51] Makinen RAE, Periaux J, Toivanen J. Multidisciplinary shape optimization in aerodynamics and electromagnetics using genetic algorithms[J]. Intl. J. for Numerical Methods in Fluids, 1999, 30(2): 149-159.
- [52] Hart W E, Baden S, Belew R K, et al. Analysis of the numerical effects of parallelism on a parallel genetic algorithm[A]. Proc. IPPS 96[C]. Los Alamitos, CA, 1996. 606-612.
- [53] Amdahl G W. Validity of single-processor approach to achieving large-scale computing capability[A]. Proc. AFIPS Conf[C]. Reston VA, 1967. 483-485.
- [54] Harik G R, Cantu-Paz E, Goldberg D, et al. The gambler's ruin problem, genetic algorithm, and the sizing of populations[J]. Evolutionary Computation, 1999, 7(3): 231-253.