分布式遗传算法在智能组卷中的 Web services 实现 *

李广水[1], 马青霞[1], 陈爱萍[1], 郑 滔[2]
(1. 金陵科技学院, 南京 211169; 2. 南京大学 软件学院, 南京 210093)

摘 要: 研究了分布式并行遗传算法的智能组卷问题。设计了在校园网范围内的分布式 Web 服务的组卷系统,通过多线程的软件结构实现了 Web 服务之间的相互调用及反馈机制,采用基于 Base64 的预先基因压缩方案来提高 SOAP 性能。仿真试验证明了分布式 Web 服务的组卷系统不仅可以极大地减少组卷时间、具有良好的容错能力,同时也能一次生成多套试卷,为构建 B/S 模式下的机器考试系统提供了依据。

关键词: 智能组卷; 遗传算法; Web 服务
中图分类号: TP391 **文献标志码**: A **文章编号**: 1001-3695(2010)11-4185-04
doi:10.3969/j.issn.1001-3695.2010.11.050

Intelligent test paper with distributed genetic algorithm realizing on Web services

LI Guang-shui[1], MA Qing-xia[1], CHEN Ai-ping[1], ZHENG Tao[2]
(1. *Jinling Institute of Technology*, *Nanjing* 211169, *China*; 2. *Software Institute*, *Nanjing University*, *Nanjing* 210093, *China*)

Abstract: To solve the problem of intelligent test paper with distributed paralleling genetic algorithm, this paper designed the Web service-based test paper auto-generating system running in campus network with multithreading software structure. It realized procedure calling and possessing feed-back mechanism between Web services, improved the SOAP performance with the Base64 encoding compress method. Experimental results validate this system not only having the power of fault-tolerant and reducing the time in working out, but also being able to generate more than one test papers once. This study can provide the reference to developing machine test system running in B/S mode.

Key words: intelligent test paper; genetic algorithm; Web services

智能组卷系统得到了广泛的重视,但是当前的热点主要集中在组卷算法的研究。由于遗传算法是一种通过模拟自然进化过程搜索最优解的方法,成为了解决搜索问题的一种通用算法,非常适合组卷过程中多目标的逼近,在自动组卷算法中得到了普遍的采用[1,2]。

面向服务的体系结构(service-oriented architecture, SOA)是为了解决在网络环境下业务集成的需要,通过连接能完成特定任务的独立功能模块实现的一种软件系统架构,经过多年的发展,Web 服务已经成为了 SOA 事实上的标准。

从组卷系统的实际可操作性来看,如果以分散于校园内的相关服务器(如各院系部门的网站服务器及数据库服务器)作为保存不同课程试题的网络数据库及组卷算法的 Web 服务提供者;以实验室的教学机或其中的某台固定机器作为本实验室考试网站,网站基于对相关服务的绑定实现考试课程的选择和考试题目的抽取,系统经过对服务的组合以达到组卷算法的分布式实现,这在现有硬件基础上就可以大大提高组卷效率,满足自动组卷实时性的要求,同时也使得系统的可扩展性及松耦合提高到一个新的水平。因此,研究基于 Web 服务的组卷系统是有其现实意义的。

1 相关研究

在机器考试系统中,一般将题库中的试题依据知识点、难度等级、题型、区分度、分值、答题耗时等属性进行预先设定,在线考试过程中依据用户设定的考试总分值、考题知识点覆盖、考试时间、期望难度等整套试卷指标进行智能组卷,这样组卷问题就是求解 P 矩阵。

$$P=\begin{vmatrix} p_{11} & p_{12} & \cdots & p_{1m} \\ p_{21} & p_{22} & \cdots & p_{2m} \\ \acute{u} & \acute{u} & & \acute{u} \\ p_{n1} & p_{n2} & \cdots & p_{nm} \end{vmatrix}$$

使得 P 的每一列满足约束条件。其中:m 代表题库中每一道题目的预设属性,n 代表整套试卷的题目数。而适应度函数定义为 $f=\sum_{i}^{m} f_i w_i$。其中:f_i 代表第 i 个属性指标与用户指定指标的误差绝对值,w_i 代表不同属性在组卷中重要程度的权值。

基于遗传算法的智能组卷需要进行染色体编码,一般采用二进制或十进制编码方案,其长度等于题库中考题条数,过长的编码直接导致算法的时空复杂性,因此对遗传组卷算法的改进成为了研究重点。主要思想包括改进遗传操作或采用分段

收稿日期: 2010-06-12; **修回日期**: 2010-07-18 **基金项目**: 国家"863"计划资助项目(2007AA01Z448);江苏省现代教育技术研究重点课题(2010-R-15267)

作者简介:李广水(1965-),男,江苏扬州人,高级工程师,副教授,博士,主要研究方向为系统集成、数据挖掘(yz_lgs@126.com);马青霞(1976-),女,江苏盐城人,讲师,硕士,主要研究方向为数据挖掘;陈爱萍(1973-),女,江苏盐城人,讲师,硕士,主要研究方向为数据挖掘;郑滔(1966-),男,教授,主要研究方向为软件工程、系统集成.

的编码策略[1,2]，以达到优化求解效率。

Web 服务基于标准的 Internet 协议来访问信息，通过 WSDL 契约，使用 SOAP 封装格式进行远程方法的消息传递，从而规范了基于 Internet 上的远程方法发现和绑定。因为 Web 服务本身的无状态，相互调用的服务或者本身提供状态反馈机制，或者基于工作流实现应用集成[3]。

遗传算法最为耗时的部分在遗传演化过程中，主要包括选择、交叉、变异等步骤。相比于串行遗传算法仅在单一集群中进行遗传优化，并行遗传算法首先让多个子种群单独进化数代后，再进行种群之间的个体交互，这符合自然种族的进化，防止了近亲繁殖而带来的子种群退化问题，可以获得更好的优化结果。并行遗传算法在独立种群演化过程中，彼此可以仅有少量通信，通过比较子种群的最优解就能得到最终结果[4]，这也为分布式遗传算法的设计提供了依据。

利用分布式遗传算法求解实际问题得到了研究，而当前基于的分布式环境一般是网格服务平台，重点探讨的是遗传算法在现有网格集群中的实现及具体应用[5,6]。相比而言，Web 服务是更低粒度上的服务提供者，因此，基于 Web 服务的组合将使得系统更具柔性，且基于具体问题进行的系统设计比采用通用的网格集群管理将发挥更高的效率，这也是本研究的主要目的。

2　系统设计

为了有效实现基于 Web 服务的分布式智能组卷系统，基于并行遗传算法的思想，采用多个子种群在不同的 Web 服务上独立进化，经过一定的进化代数之后进行种群之间的交互操作，并对交换后的种群再次独立进化，重复该过程直至满足结束条件。

图 1 给出了基于 Web 服务的分布式遗传组卷算法的体系结构。其中，M-Web 首先形成初始多个子种群并将参数传送给不同的 S-Web 进行节点服务调用；此后由节点的每个处理器在约束条件下独立地对染色体进行选择、交叉、变异操作，并计算操作后的染色体适应度；当染色体进化指定代数后，S-Web 提交适应度高的部分染色体至 M-Web，M-Web 将来自不同S-Web提交的染色体彼此进行交换形成新的种群提交给不同的 S-Web，当某一个 S-Web 进化到符合终止条件时，提交适应度最高的染色体给 M-Web 并终止该 S-Web 服务；M-Web 比较所有提交的染色体适应度，选取最好的染色体为求解结果。

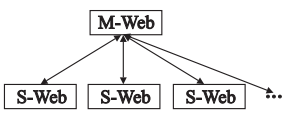

图1　基于Web服务的分布式遗传算法的体系结构

系统需要开发两种 Web 服务，即 M-Web 和 S-Web。其中 M-Web 担负协调、集成等任务。从系统容错性、实时性出发，Web 服务需要符合以下要求：

a) S-Web 需要有被成功调用的反馈机制。

b) M-Web 以多线程方式实现。具体而言，除分发给每一个 S-Web 子种群求解任务，还在本地开辟一个该子种群的遗传求解线程 S-thread$_i$，并且当某一个 S-Web 不能被成功调用时，将对应的 S-thread$_i$ 优先级设为最高。

c) Web 服务在求解过程中，若某个 S-Web 完成该子种群的进化指定代数后，将提交适应度高的部分染色体至 M-Web 以便种群之间的染色体交换，同时中断 M-Web 中对应的进程 S-thread$_i$，更新相关数据后再重启 M-Web 中的该线程。

d) 在得到一个 S-Web 最终返回结果以后，如果 M-Web 中还有 S-thread$_i$ 在运行，除了结束对应的 S-thread$_i$ 线程，还分配某个在本机正运行的另一个 S-thread$_i$ 及对应数据给该 S-Web；相应地，如果 M-Web 中某一 S-thread$_i$ 线程得到最终结果，除了中止对应的 S-Web，将以其他运行中的 S-thread$_i$ 及相应参数重启该 S-Web。

图 2 表述了 M-Web 和 S-Web 在协同遗传进化过程中的进程态势。一个考试网站在实际调用该服务时，只是绑定 M-Web 服务，即仅发布 M-Web 供 Internet 集成。

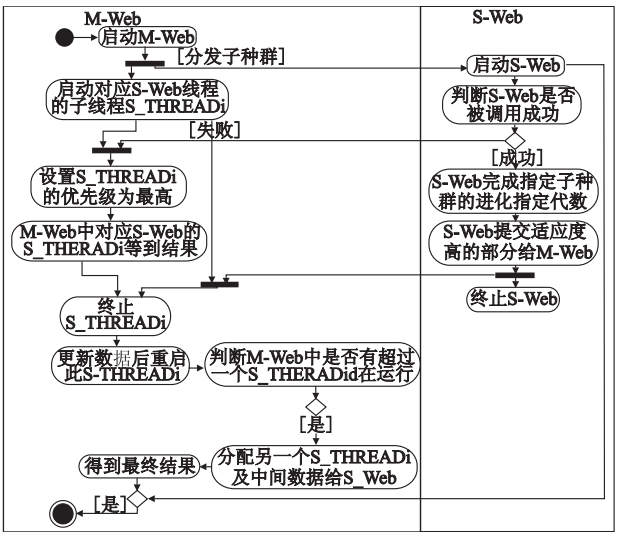

图2　Web服务之间协同遗传进化活动图

基于以上原则设计的分布式服务，既能充分利用不同节点的运算能力，提高系统的运行效率，同时也保证了某一节点出错的情况下依然可以进行结果求解，最大限度地达到系统的容错性。当然，该系统的设计对 M-Web 服务器具有较高的要求，不仅需要担负相关 S-Web 服务的调用管理、种群进化等任务，且即使所有期望的 S-Web 服务器都能正常运行，为防止运行期错误，在 M-Web 中依然存有对应的线程在进行不同子种群的遗传求解，这也是容错系统的代价。

3　服务性能改进

相比于其他分布式架构（如 RMI、CORBA），Web 服务在提供松耦合的同时也存在较大的在线弱势[7~10]，主要原因是 SOAP 需要额外生成和解析 XML 文档，其过程包括解析 XML 文档、将 XML 数据映射为对应高级语言格式的序列化、服务调用之后反映射为 XML 数据格式的反序列化。进一步研究发现[8~11]，在解析 SOAP 过程中，SOAP 封装的不同数据格式及数据大小对服务性能的影响也存在差异。花磊等人[9]研究表明，随着 SOAP 载荷增大，三个过程的耗时都逐渐增大，但序列化和反序列化时间增幅超过 XML 解析时间；Anupam 等人[10]在对基于 SOAP 的 Web 服务性能测试过程中发现，SOAP 中绑定的不同数据类型对 Web 服务存有重要的影响，就基本数据类型而言，浮点型在映射过程中耗时最多，而字符型表现最好。针对 SOAP 的性能特点，W3C 也多次提出 SOAP 的优化建议[11]，包括改进 SOAP 的传输机制（MTOM）、增加基于二进制的优化包（XOP）。

基于 Web 服务的遗传算法在进化过程中需要在不同的服务之间来回传输遗传种群,直接传输二进制或十进制编码的基因种群会因为 SOAP 绑定的数据量大及不适当的数据格式导致服务性能降低。为此,提出基于 Base64 的基因预先压缩方案。

采用 Base64 编码方案是 W3C 推荐标准[12],其形成的字符串可以直接在 HTTP 中传输,在不同的编程语言中不会形成转义而得到通用。

以二进制编码为例,M-Web 形成的初始子种群中的基因无论是以整型还是字符型表示,每个基因需要占用一个字节,参照 Base64 编码方案,可以将连续 6 个基因编码为一个对应的字符,这样在基因编码较长的情况下,压缩后的 SOAP 所占字节数将较原先大大减少。服务调用过程中,在序列化以前首先进行 Base64 编码压缩,服务器端获取 SOAP 进行反序列化之后再进行解压,从而得到原始的基因编码。压缩和解压缩是基于本地高级语言实现,因此时耗将会很少。

假设有以下的原始基因序列:10010111,如果以字符型格式存储,则相应的二进制编码将会是:

00110001 00110000 00110000 00110001

00110000 00110001 00110001 00110001

对应的 Base64 基数分别为 DEwMDEwMTEx。其时,基于 Base64 编码方案的 SOAP 形如以下格式:

```
〈? xml version = '1.0' ?〉
〈soap:Envelope xmlns:soap = "http://schemas. xmlsoap. org/soap/envelope/" xmlns:xsi = "http://www. w3. org/2001/XMLSchema-instance"〉
〈soap:Body〉
〈group xsi:type = "xsi:base64Binary"〉 DEwMDEwMTEx =〈/group〉
〈/soap:Body〉
〈/soap:Envelope〉
```

如果预先对基因进行压缩存储,则 10010111 将被压缩为“CX”两个字符,其保存的两个字节转换为三个六进制的 Base64 基数为“ENY”, 对应的 SOAP 将形如以下格式:

```
…
〈soap:Body〉
〈group xsi:type = "xsi:base64Binary"〉 ENY =〈/group〉
〈/soap:Body〉
…
```

显然,在有较长编码的种群之间进行分布式遗传进化过程中,基于 Base64 的基因预先压缩方案将缩减 SOAP 的数据量,并且以字符串方式进行传输,从而提高服务性能。

4　仿真实验

4.1　原型系统开发

仿真实验利用. NET 工具和 SQL Server 2005 进行了 Web 服务开发,调用实际已被使用的试题库管理系统中 VFP 程序设计课程试题近 2 000 条。因为本研究的重点在于分布式组卷算法的有效性,对题库中试题的一些缺少属性进行了临时设定,最终的测试素材包括 500 条选择题、500 条填空题、300 条程序完善题、380 条程序改错题四种题型,每道试题的预设属性分别为题型、知识点、难度等级、区分度、分值、答题耗时六个部分。

染色体编码采用二进制方式,其中 1,0 分别代表某一题目是否被选择。考虑到每一题型中试题条数不能变化,在遗传交叉过程中以题型编码段为单位进行;在变异过程中也是某一题型内保证 1 的个数不变,其中交叉概率选 0.8,变异概率选择 0.01,为方便起见,适应度函数的权值都取 1。

实际布置了三个 S-Web 服务,设定每一个子种群数目分别为 10、20、30,这样整个种群进化分别在 30、60、90 个染色体上进行。分别设定各个独立的 S-Web 服务每一次进化代数达到 100 次时,将选出子种群中适应度最高的前一半染色体作为候选染色体子群。S-Web 首先提交候选子群给 M-Web,M-Web 进行彼此交换,并将交换后的染色体返给对应的 S-Web。每一个 S-Web 将交换来的染色体与原先的候选染色体共同构成了新一代种群并再次进化,当 S-Web 与 M-Web 交换五次后完成全部遗传进化。

各个 Web 服务上的子种群保存在. NET 的 DataSet 控件中的表里, DateSet 对象可以动态提取远程数据库中的数据并构造易于数据处理的本地内存数据库,大大减少数据库访问次数,还可以参数读入数据,这为异地数据传输、处理以及应用软件的可伸缩性提供了便利。本系统每一个 Web 服务的 DateSet 实例中存有两张数据表:试题库中每一试题的题型、知识点、难度等级、区分度、分值、答题耗时等属性数据表和染色体编码表。由于表的字段长度限制,因此每一个染色体占据表的固定数行,相关遗传操作和种群更新直接作用在数据表上,这保证了数据操作的简单方便。

4.2　实验及结果分析

仿真在实验室的四台 PC 机上进行,配置为 Intel Pentium Dual Core E5200,2 GB 内存,Web 调用直接在 M-Web 主机内,测试结果也存放在 M-Web 主机内的 SQL Server 之中。组卷要求假设为 15 道选择题、30 道填空题、2 道程序完善题和 3 道程序改错题,总分 100 分,考试时间 120 分钟,要求覆盖 12 个知识点,难度等级为 2,区分度为 5。测试主要针对基于 Web service 分布式组卷算法的有效性进行,具体包括测试内容:a)不同数目的子种群进化过程及结果比较;b)系统的容错能力以及分布式系统的优势;c)基于 Base64 基因预压缩方案改进效果。

不同子种群在五次迭代中的最小适应度值如表 1 ~ 3 所示。

表 1　子种群数目为 10 的遗传进化在每次迭代中各 S-Web 的最小适应度值

迭代次数	最小适应度值		
	S-Web1	S-Web2	S-Web3
1	195	173	203
2	85	106	92
3	38	58	42
4	26	28	30
5	15	18	13

表 2　子种群数目为 20 的遗传进化在每次迭代中各 S-Web 的最小适应度值

迭代次数	最小适应度值		
	S-Web1	S-Web2	S-Web3
1	155	178	160
2	72	80	78
3	29	32	32
4	16	21	20
5	7	8	7

为了检验系统的容错能力以及不同数目的服务参与情况下的求解效果,分别设置系统在只有一个服务(M-Web),有两个服务(M-Web 及一个 S-Web)、三个服务(M-Web 及两个 S-Web)、所有服务正常的情况下求解不同种群数五次迭代过程中的累计耗时情况。表 4 是种群数为 90 时的相关数据。

表 3　子种群数目为 30 的遗传进化在每次迭代中各 S-Web 的最小适应度值

迭代次数	最小适应度值		
	S-Web3	S-Web1	S-Web2
1	148	138	150
2	41	40	39
3	17	18	20
4	7	5	5
5	5	5	5

表 4　种群数目为 90 时不同数目的 S-Web 服务参与下的运行时间　s

迭代次数	参与的 S-Web 数			
	a	b	c	d
1	36	28	21	9
2	70	46	38	15
3	112	66	49	21
4	146	83	58	38
5	176	93	65	43

注：a ~ d 分别代表 1、2、3 个及全部参与运算的 Web 服务数目

图 3 是不同种群数目下，基于 Web 服务的分布式遗传出卷算法在不同 Web 服务个数参与情况下最终求解时间的折线图。由具体实验可以得到以下结论：

a)种群数越多最优解的逼近速度越快。就组卷系统而言，更多的种群数在较少的进化代数之后就可以获得结果。

b)分布式遗传算法随着参与运算节点的增加其求解时间快速减少。种群数越大，采用分布式遗传算法的效果越好。

c)当设定一个适应度阈值之后，可以通过选取不同服务返回的满足条件的多个解，达到同时生成多套试卷的目的。

为验证本文提出的基因预压缩在性能提高方面的有效性，笔者实际比较了两种系统在不同子种群下的进化时间，图 4 显示了其存在的差异。很显然，在实际存有更大的试题数量时，基因编码将会更长，基于 Base64 预压缩的 SOAP 改进效果将会更为明显。

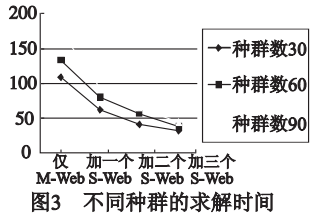

图3　不同种群的求解时间

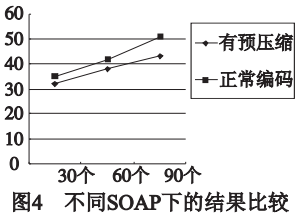

图4　不同SOAP下的结果比较

5　结束语

本文研究的主要目的是探讨基于 Web 服务的分布式遗传组卷算法，相比于其他分布式结构，Web 服务可以实现 Internet 下的调用绑定，是 B/S 模式下信息系统集成的首选服务。

由于 Web 服务本身的无状态，如再进行远程调用及大数据集传输，其在线性将受到一定的影响。本研究采用服务之间相互调用的反馈机制并进行编码预压缩，在一个局域范围内实现，达到了较好的效果，从组卷的实际应用环境而言是完全切实可行的。而从系统的易于扩展性出发，能否动态地添加 S-Web 显然是问题的一个关键；另一方面，节点间染色体交换的比例、某次迭代过程中能否动态地实现 S-Web 之间染色体交换，这些都值得进一步研究。

参考文献：

[1] ZHOU Yan-cong, LI Yuan-yuan, FENG Chao. Research on intelligent test paper auto-generating algorithm based on improved GA [C]//Proc of the 21st Chinese Control and Decision Conference. Piscataway: IEEE Computer Society Press, 2009:3974-3978.

[2] OUYANG Yong, LUO Hong-fang. Design of personalized test paper generating system of educational telenet based on genetic algorithm [C]//Proc of the 4th International Conference on Computer Science & Education. [S.l]: IEEE Computer Society Press, 2009:170-173.

[3] 范贵生，刘冬梅，陈丽琼，等. 可靠服务组合的协调策略与分析[J]. 计算机学报, 2008, 31(8):1445-1457.

[4] 曾孝平，陈燕飞，李勇明. 实数自适应并行遗传算法的研究[J]. 计算机应用研究, 2008, 25(6):1687-1689,1735

[5] HERRERA J, HUEDO E, MONTERO R, *et al*. A grid-oriented genetic algorithm [C]//Proc of European Gride Conference. Berlin: Springer, 2005 :315-322.

[6] LIM D, ONG Y S, JIN Yao-chu, *et al*. Efficient hierarchical parallel genetic algorithms using grid computing [J]. Future Generation Computer Systems, 2007, 23(4):658-670.

[7] COHEN F. 发现 SOAP 编码对 Web 服务性能的影响[EB/OL]. (2003-03-01)[2010-05-20]. http://www.ibm.com/developerworks/cn/Webservices/ws-soapenc/.

[8] 李磊，牛春雷，陈宁江，等. 一种高效的 Web 服务性能优化策略[J]. 计算机研究与发展, 2007, 44(7):1191-1198.

[9] 花磊，魏峻，牛春雷，等. 动态模板驱动的高性能 SOAP 处理[J]. 计算机学报, 2006, 29(7):1145-1156.

[10] NATH A K, SINGH R. Evaluating the performance and quality of Web services in electronic marketplace[C]//Proc of the 14th Americas Conference on Information Systems. 2009:1-11.

[11] W3C SOAP current status [EB/OL]. [2010-06-06]. http://www.w3.org/standards/techs/soap#w3c_all.

[12] W3C SOAP version 1.2 part 0: primer. 2nd ed[EB/OL]. (2007-04-27)[2010-06-06]. http://www.w3.org/TR/soap12-partol.

(上接第 4174 页)

[3] HE Yi-gang, SUN Yi-chuang. Neural network-based L1-norm optimization approach for fault diagnosis of nonlinear circuits with tolerance [J]. IEE Proceeding: Circuit Devices Systems, 2001, 148(4): 223-228.

[4] TADEUSIEWICZ M, HALGAS S, KORZYBSKI M. An algorithm for soft-fault diagnosis of linear and nonlinear circuits[J]. IEEE Trans on Circuits Systems I: Fundamental Theory and Applications, 2002, 49(11):1648-1653.

[5] ZHOU Long-fu, SHI Yi-bing. Soft fault diagnosis in analog circuit based on fuzzy and direction vector[J]. Metrology and Measurement Systems, 2009, 16(1):61-75.

[6] 金瑜，陈光禹，刘红. 基于非张量积小波网络的模拟电路故障诊断[J]. 仪器仪表学报, 2008, 29(8):1613-1616.

[7] EBERHART R, KENNEDY J. A new optimizer using particle swarm theory[C]//Proc of the 6th International Symposium on Micro Machine and Human Science. 1995: 39-43.

[8] KENNEDY J. The particle swarm: social adaptation of knowledge [C]//Proc of International Conference on Evolutionary Computation. Piscataway: IEEE Press, 1997:303-308.

[9] OZCAN E, MOHAN C K. Particle swarm optimization: surfing the waves[C]//Proc of Congress on Evolutionary Computation. Washington DC: IEEE Computer Society, 1999:1939-1944.

[10] 李宁，孙德宝，邹彤，等. 基于差分方程的 PSO 算法粒子运动轨迹分析[J]. 计算机学报, 2006, 29(11): 2052-2061.

[11] 周龙甫，师奕兵，李焱骏. 容差条件下 PSO 算法诊断模拟电路单软故障方法[J]. 计算机辅助设计与图形学学报, 2009, 21(9): 1270-1274.

[12] 周龙甫，师奕兵. PSO 算法粒子运动轨迹稳定收敛条件分析[J]. 控制与决策, 2009, 24(10):1499-1503.

[13] COUZIN I D, KRAUSE J, FRANKS N R, *et al*. Effective leadership and decision making in animal groups on the move [J]. Nature, 2005, 433(7025):513-516.

[14] RATNAWEERA A, HALGAMUGE S K, WATSON H C. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients[J]. IEEE Trans on Evolutionary Computation, 2004, 8(3):240-255.

[15] 彭祖赠，孙韫玉. 模糊(Fuzzy)数学及其应用[M]. 2nd ed. 武汉：武汉大学出版社, 2007.