基于混沌反馈控制理论的资源选择算法研究

余云霞[1] 綦志勇[2]

(荆楚理工学院 荆门 448000)[1] (武汉软件工程职业学院 武汉 430205)[2]

摘 要 当前的服务发现技术主导着对用户的信息获取,因此一个好的服务资源选择成为服务发现的关键问题之一。目前的服务资源选择技术在动态服务资源选择方面,还不能很好地对服务资源的动态性与实时性进行处理。通常使用基于历史资源信息的方法来进行动态服务资源的选择。在服务过程中,由于种种不确定的原因,资源对应的节点可能出现终止服务的可能性,并且实时资源可能不在历史信息列表中等问题的出现,需要一个合适的动态资源选择方法来进一步解决这些问题。针对上述问题,提出了一种基于混沌反馈控制思想的动态服务资源选择方法,以选择一个合适的动态服务资源为目标,该方法改进了基于历史信息动态资源选择方法的不足。

关键词 服务发现,动态性,实时性,反馈控制,动态服务资源选择

中图法分类号 TP311 **文献标识码** A

Research on Resource Selection Algorithm Based on Feedback Control of Chaos Theory

YU Yun-xia[1] QI Zhi-yong[2]

(Jingchu University of Technology, Jingmen 448000, China)[1] (Wuhan College of Software Engineering, Wuhan 430205, China)[2]

Abstract Service discovery technology dominates the information gets to the user, so a good service resources choose to become one of the key issues of service discovery. Current service resource selection techniques in dynamic service selection of resources, not a good dynamic and real-time processing of service resources. Often use historical resources information based method for dynamic service selection of resources. Due to various causes of uncertainty in the service process, resource node that corresponds to the possible termination service possibility and real-time resource may not be appears in historical information list. You need a suitable dynamic resource selection method to further address these issues. In response to these problems, the article presented a dynamic servic resource selection method based on chaos feedback control. To select a suitable dynamic services resources as the goal, which improves the insufficient of dynamic resource selection method based on historical information.

Keywords Service discovery, Dynamic, Real-time, Feedback control, Dynamic services resources select

1 国内外研究现状

目前,国内外文献对服务选择问题提出了许多观点。文献[1]提出一种网格的动态资源服务组成方法,其利用近期的任务执行信息来预测未来的服务选择方式,不足是动态性与不确定性不能控制,且文中动态性参考数据为资源CPU的负载情况,显然不宜。文献[2]仅根据任务的历史执行情况来进行预测,对服务资源的实时性与动态性并未考虑。文献[3]从任务发起的角度考虑了服务选择,主要考虑多智能体的静态服务资源选择问题。文献[4]使用空间知识库的方式来存储服务列表,在资源注册成功后该表即为静态;且资源注册需额外存储,属静态服务选择方法。文献[5-9]均提出了各自的服务资源选择方式,但是这些服务选择大多基于QOS的局部或者全局查找最优解决Web的资源选择问题,仍然不能兼顾动态性与实时性。

国内在服务资源选择方面的研究也有很多。文献[10]提出了一种基于历史信息的动态资源选择方法。其缺点是资源的动态性控制不足,且仅限于局域网。文献[11]的主要目标是寻找Web服务中的全局最优资源。该方法的使用对寻找一个静态全局最优解十分有效,但其未考虑动态性与实时性,且寻优目标数量极大的情况下效率较低。文献[12]使用了蚁群算法来寻求服务组合中最优化动态服务选择问题的解。但未对使用该方法定义的拒绝服务问题做出解释。文献[13]提出了一种基于自适应算法的动态网格服务选择方法。在实际中的服务资源可能是动态与时间相关并且随机的,服从泊松分布。该方法在寻优完成时得到的是一个静态候选的服务资源,难以平衡动态性。

2 基于混沌反馈控制的动态服务资源选择方法

我们希望能够实时监控服务资源的情况,并通过服务资源的当前使用情况来实时地对服务资源选择过程进行动态控制。故首先对服务资源建模。

2.1 服务资源建模

2.1.1 历史资源的建模

历史资源是一个任务曾经选择的资源轨迹。对历史资源

余云霞(1975—),女,硕士,讲师,主要研究方向为普适计算、数据挖掘,E-mail:zhouyuyun@tom.com;綦志勇(1977—),男,硕士,助教,主要研究方向为普适计算、嵌入式系统。

主要考虑如下几个方面。历史资源表达为：

$$RSEhis=(Po,Re,St,Nd,Lh) \tag{1}$$

式中，Po 标示了历史资源的网络位置。为通用性起见实际中应该采用 URL 来表示网络资源的位置。资源的平均历史可靠性 Re，由公式[10]

$$Re(X)=\frac{1}{n}\sum_{i=1}^{n}succ_i^p \tag{2}$$

确定。其中 $succ_i^p=\begin{cases}1, & \text{服务 } p \text{ 执行成功}\\ 0, & \text{服务 } p \text{ 执行失败}\end{cases}$，该参数决定资源 X 运行服务 p 是否成功。平均历史服务执行时间 St，由公式

$$St=\frac{1}{n}\sum_{i=1}^{n}T_i^p \tag{3}$$

确定。其中 T_i^p 表示资源 X 运行服务 p 的单次任务执行成功的执行时间。资源的历史平均网络延时 Nd，由公式

$$Nd=\frac{1}{n}\sum_{i=1}^{n}Nt_i^p \tag{4}$$

确定。其中 Nt_i^p 表示资源 X 运行服务 p 的单次任务执行成功的网络通讯时间。

资源的历史平均负载情况 Lh，由公式

$$Lh=\frac{1}{n}\sum_{i=1}^{n}Ll_i^p \tag{5}$$

确定。其中 Ll_i^p 表示资源 X 运行服务 p 的单次任务执行成功的负载情况，通常使用 CPU 的使用率来表达。

2.1.2 实时资源的建模

实时资源是一个任务等待选择的服务资源。对实时资源主要考虑如下几个方面。实时资源表达为：

$$RSEret=(rPo,rRe,rSt,rNd,rLh) \tag{6}$$

式中，rPo 标示了当前资源的网络位置，实际中使用 URL 来表示。资源的可用性 rRe，资源可用性由公式

$$rRe(X)=\frac{1}{n}\sum_{i=1}^{n}succ_i^p \tag{7}$$

确定。其中，$succ_i^p=\begin{cases}1, & \text{允许服务 } p \text{ 执行}\\ 0, & \text{服务 } p \text{ 未授权}\end{cases}$。该参数决定资源 X 是否允许运行服务 p。资源的预测服务执行时间 rSt，由公式

$$rSt(X)=Nd+St \tag{8}$$

决定。其中，St 表示平均历史服务执行时间，Nd 表示平均历史网络延时。如果是首次使用该资源，则 St 设置为预定时间（例如可以设定为在本机上执行的可能时间）；Nd 为实际网络延时 rNd，即 $Nd=rNd$。资源网络延时 rNd，由公式

$$rNd(X)=|Ts-Tr| \tag{9}$$

决定，其中，Ts 表示发送请求时刻，Tr 表示接收响应时刻。资源的负载情况 rLh，由公式

$$rLh(X)=\begin{cases}M, & Lcpu\geqslant M\\ Lcpu, & Lcpu<M\end{cases} \tag{10}$$

决定。$Lcpu$ 为实际负载情况。其中 $M>0$ 为某一经验正数，由实际情况确定。$Lcpu\geqslant M$ 表示对方负荷较重，反之可以正常接收任务。根据 rLh 的值可以把该资源作为参考备用资源。若 rLh 超过 M 则该资源负载过大暂时无法完成任务。

2.2 动态服务资源选择数学描述

一个动态服务资源选择系统可以描述为：

$$\bar{X}=f(X,U) \tag{11}$$

式中，$\bar{X}$ 为输出，与输入 X 相关。

X 为资源向量，$X=[X_1,X_2,\cdots,X_n,\cdots]^T$，其中，$X_i$ 表示一个实时资源描述。

$$X_i=RSEret=(rPo,rRe,rSt,rNd,rLh) \tag{12}$$

式中，U 为控制向量，$U=[y_1,y_2,y_3]^T$，控制分量 y_i 主要有 3 个，y_1 表示对资源的选择，y_2 表示对实时资源的申请，y_3 表示对历史资源情况的修改。

f 为决策函数，$f=[f_1,f_2,f_3]^T$，f_1 表示对资源选择的控制，f_2 表示对资源申请的控制，f_3 表示对历史资源修改的控制。

考虑反馈控制的方法对上式加入以时间为间隔的微扰，由于我们主要关心如何选择最佳资源，从而对动态资源选择系统进行描述。该系统可以描述为如下方程：

$$\begin{cases}\bar{y}_1=f_1(x\text{资源},x\text{历史资源},y\text{当前控制})+F(t)\\ \bar{y}_2=f_2(x\text{资源},x\text{结果},t\text{间隔时间})\\ \bar{y}_3=f_3(x\text{结果},x\text{历史资源})\\ \bar{x}=Q(x,y)\end{cases} \tag{13}$$

式中，$\bar{y}_i$ 描述系统控制信号，为系统的输出变量，$\bar{x}$ 描述难以预测的变量，$F(t)$ 定义如下：

$$F(t)=KD(t)=K[y(t-\partial)-y(t)] \tag{14}$$

式中，$y(t)$ 作为系统的一个正常输出，$y(t-\partial)$ 作为系统延迟之后的反馈输出，表达了系统在反馈外部情况之后对系统进行输入对比。其中，∂ 为延迟时间，该时间定义为在发送两个服务请求数据包的时间间隔不引起网络拥塞情况下的最小延迟。K 为控制因子，通过控制 K 与 $D(t)$ 来控制系统的输出。

2.3 动态服务资源选择方法

通过上述对动态服务资源选择系统进行建模之后，则需要讨论如何对系统进行控制。

(1)系统的输入

由参考文献[14]知，系统的输入为资源列表，它是历史参考资源表。事实上，如果要实现反馈控制还需要回传结果，将其作为输入之一。因此，系统的主要输入形式化描述如下：

X_{rel} 实时资源向量：

$$X_{rel}=[X_{r1},X_{r2},\cdots,X_m,\cdots]^T \tag{15}$$

式中，X_{ri} 表示一个实时资源描述。

X_{his} 历史资源向量：

$$X_{his}=[X_{h1},X_{h2},\cdots,X_{hm}]^T \tag{16}$$

式中，X_{hi} 表示一个历史资源描述。

这里，实时资源向量为无限长度，历史资源向量长度受存储限制。X_{nt} 表示服务结果，T 表示请求列表时间间隔。

(2)系统的输出与控制函数

由系统模型函数知，资源选择控制输出 $\bar{y}_1$ 依赖于当前资源列表，历史资源服务情况对比与当前的控制输出。因此，这里有

$$\bar{y}_1=f_1(X_{rel},X_{his},y) \tag{17}$$

其中，

$$X_{rel}=[X_{r1},X_{r2},\cdots,X_m,\cdots]^T$$

$$X_{his}=[X_{h1},X_{h2},\cdots,X_{hm}]^T$$

对于函数 f_1，其输入为实时资源列表 X_{rel}、历史最优资源

列表 X_{his} 与当前输出 y。初始化函数 $y=0$ 表示没有选择任何资源。当实时资源列表 X_{rel} 输入时，则选择列表中的一部分进行实时查询操作。这里，考虑到 X_{rel} 向量为不可数，故选择列表的一个以时间 β_T（β 与具体计算设备的缓冲区大小相关）为顺序的样本来进行查询。则上式可以改写成：

$$\bar{y}_1 = f_1(X'_{rel}, X_{his}, y) \tag{18}$$

式中，$X'_{rel}=(x_{r1}, x_{r2}, \cdots, x_{rm})^{\mathrm{T}}$。查询的方式与历史资源信息相关。如果 X'_{rel} 与 X_{his} 的相关系数为 0，则表明：

$$X'_{rel} \cap X_{his} = \Phi \tag{19}$$

那么可能当前属于初次服务资源选择。由对应的策略函数作出匹配，并选择一个合适的策略。

对于函数 $\bar{y}_2$，其主要功能为发送资源请求申请，它依赖于当前资源列表、服务返回结果与定时时间。即：

$$\bar{y}_2 = f(X_{rel}, x_{rlt}, t) \tag{20}$$

式中，$X_{rel}=[X_{r1}, X_{r2}, \cdots, X_{rn}, \cdots]^{\mathrm{T}}$。

结果与定时时间 T 起到了决定作用，如果存在结果回传，则无需发送资源申请。故此，找到了系统的不动点 $f(\bar{X}, \bar{U})=0$，控制过程结束。否则根据当前资源列表与定时时间决定是否发送资源请求。

对于函数 $\bar{y}_3$，其主要功能为替换历史记录表中的结果。它依赖于服务返回结果与历史记录表。即：

$$\bar{y}_3 = f(x_{rlt}, X_{his}) \tag{21}$$

式中，$X_{his}=[X_{h1}, X_{h2}, \cdots, X_{hm}]^{\mathrm{T}}$。如果存在结果回传，则与最优历史记录表进行比对，并根据情况修改记录。

(3)基于混沌反馈控制的动态服务资源选择方法

前述我们对动态服务资源的选择方法进行了描述，并定义了函数的外部功能。这里提出基于混沌反馈控制的动态服务资源选择模型，其基本框架如图 1 所示。

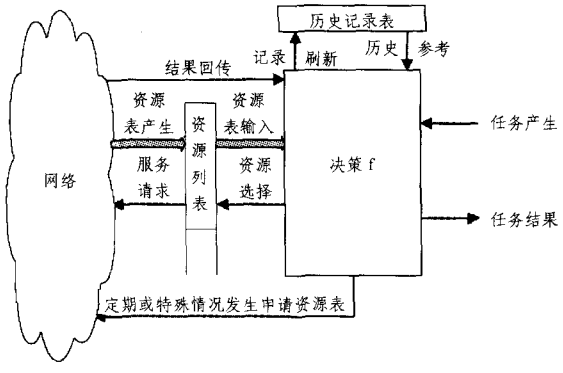

图 1　反馈控制资源选择模型

从图 1 可以看到，该图体现了反馈控制资源选择模型的主要流程。这里对模型中的各个部分进行介绍。

历史记录表：其中包含资源的网络位置 Po，资源的历史可靠性 Re，资源的平均历史服务执行时间 St，资源平均网络延时 Nd，资源的平均历史负载情况 Lh。该表表达了历史资源的详细情况。

决策函数：决策函数是该模型的核心。决策函数的主要功能表达为 3 个方面的决策与控制。其首要功能是实时资源的选择。如果实时资源出现断开连接的状况这个函数必须自动再次连接并重新选择一个合适的服务资源。并且在选择一个合适资源时应该就历史最优表的情况进行对比判断。考虑到系统执行时间，这里的判断算法应在 β_T 时间内完成。第二个方面是定期发起资源列表的申请。这里的列表申请作为函数的微扰，属于反馈结果。定期申请的资源列表项目众多，考虑算法的优化，这里只对被控的资源进行定期查询式申请。第三个方面是当资源成功完成结果的回传时，应该根据资源的执行情况来修改历史资源最优表，用于下次资源选择的参考。

资源列表：实时资源列表在理论上是一个不可数集合。事实上，如果申请了一个实时资源列表，其中的表项目也是数量巨大。因此，使用滑动窗口的方式来进行输入列表的移动选择。由决策模块决定需要滑动几个窗口，在每一个窗口中仅仅选择一个合适的资源。由决策再次对几个局部最优资源选择一个最优。

反馈控制资源选择模型的主要工作流程如下：

第一步　任务产生。

第二步　由决策模块发送服务资源请求：决策函数 f 的初始条件为 $X=0T$，表示启动系统，尚无资源可选。并调出历史可用资源列表。

第三步　网络返回可用资源列表：由于资源的理论无限个数，决策应在接收缓冲大于限制值 λ（该值根据具体的系统缓冲区大小进行经验选择）之前作出决策，选择合适的资源来取得服务或者是递交任务。

第四步　选择合适的资源：首先对列表选择一个包含合适列表项的列表窗口 $X_i \subset X$，其中，$X_i=[X_{r1}, X_{r2}, \cdots, X_{rm}]^{\mathrm{T}}$。并且根据历史最优资源表的情况对窗口之内的 n 个值按某种条件进行查找，如果找到合适的资源则使其执行，如果没有符合条件的资源，则至少选择一个最优资源。

第五步　等待结果回传：如果在规定时间 T 之内有结果回传，则传送结果到上层，计算结束。如果没有结果回传，说明计算资源正在执行中或者是网络故障。此时应该等待 T 时间，当时间到则返回第二步继续执行。

3　实验

(1)实验方法

实验程序允许同时发送服务资源申请与接收服务资源请求。该程序模拟服务资源申请端，并且可以同时模拟服务资源响应申请端。在本地计算机上启动一个程序，在局域网计算机上各自启动至少一个程序。此时由其中一个程序向其他程序发起申请，其他程序对这个申请同时进行响应，我们对每一台计算机上的相应程序各自安排一个资源优先级编号。启动动态资源搜索，从候选实时资源列表中选择一个合适的资源并提交资源申请。候选资源接收到申请之后由接收程序延时两分钟左右，作为资源执行时间的模拟。此时我们强制结束当前连接的服务程序，通过强制结束模拟资源的不确定性。若在动态扫描该资源过程中发现该资源断开连接，则重新自动刷新资源列表，再次选择一个合适资源进行连接。直到正常返回结果或者用户终止任务。实验使用到的数据如表 1 所列。

表 1　实时与历史资源数据列表

全部实时资源表				
资源标识	可用性	预测执行时间	网络延时	资源负载
pc-1	1	0.32	0.002	0.3
pc-2	1	0.72	0.22	0.5
pc-3	1	0.62	0.022	0.9
pc-4	1	0.76	0.98	0.78
pc-5	1	0.23	0.102	0.1
pc-6	1	0.53	0.978	0.56
pc-7	1	0.44	0.034	0.4
pc-8	1	0.35	0.011	0.34
pc-9	1	0.28	0.089	0.2
pc-10	1	0.95	0.45	0.6

全部历史资源表				
资源标识	历史可用性	平均服务时间	平均网络延时	平均负载
pc-1	0.9	0.25	0.04	0.35
pc-2	0.9	0.22	0.08	0.33
pc-3	0.8	0.35	0.05	0.76
pc-4	0.7	0.32	0.09	0.67
pc-5	0.7	0.21	0.12	0.56
pc-6	0.6	0.13	0.08	0.34
pc-7	0.5	0.27	0.09	0.22
pc-8	0.5	0.23	0.07	0.34
pc-9	0.4	0.24	0.08	0.78
pc-10	0.3	0.33	0.06	0.45

(1)仿真结果曲线图

我们根据两种实验的结果来绘制其资源计算结果曲线图。该图比较了 10 个实时资源的资源选择参数。

①正常结果返回试验择优曲线图

从图 2 可见，PC-5 的曲线对应点最低，表示该资源目前比较优的性能作为任务完成的服务资源。

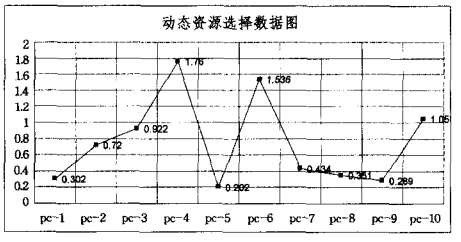

图 2　动态资源选择曲线

②不确定资源情况下的试验数据择优图

由图 3 可见，当节点 PC-5 出现问题被去掉之后最好的资源选择是 PC-9。

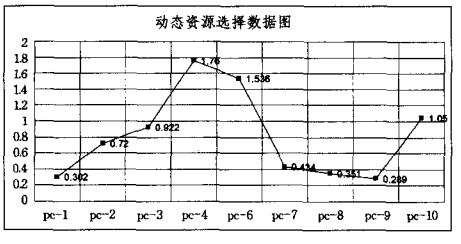

图 3　去掉一个节点的动态资源选择曲线

4　动态服务资源选择方法比较

通过仿真实验的情况与得出的数据，下面就对本文使用的动态服务资源选择方法与其他方法进行比较。具体从如下几个方面对其进行比较。

(1)有历史资源表时的较优资源选择

基于历史资源信息的动态资源选择方法是历史表存在情况下的资源选择方法，本文采用的方法可以不使用历史表。若当历史表动态填写完成之后，两种方法在正常情况下基本一致。区别是在计算资源的最优值时我们采用直接评估。即当资源可用的前提下，将负载情况 rLh 与网络延时 rNd 直接相加，选择最小值作为资源最优。而基于历史信息的方法选择百分比最大的值作为最优。则可以将基于历史信息的方法评估值使用如下公式转化：

$$ResBest=1-Eval \tag{22}$$

表示将其最优值变为最小值。对于相同数据，基于历史信息的方法较优值仿真曲线图如图 4 所示。

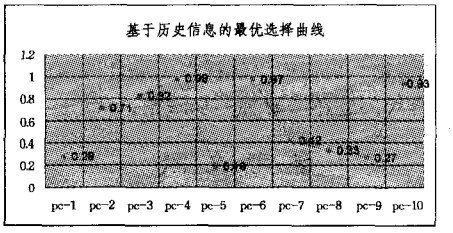

图 4　基于历史方法的较优曲线图

图 4 与图 2 比较最优也是 PC-5 资源，且曲线走向一致。因此，在有完善历史参考的情况下，基于混沌反馈控制的动态服务资源选择方法与基于历史资源信息的方法具有近似的性能。

(2)无历史资源表时的较优资源选择

当无历史资源表或历史资源表项目很少时，通常情况下表中元素个数小于 3。这种情况发生在首次启动资源表，或者多次选择的最优资源均为相同的几个资源。基于历史资源信息的方法显然不能选出较好的结果。这是由于没有历史资源信息作为参考，故资源的可用性 $Re(X)=0$，则计算的资源较优评估值公式变为：

$$Eval=Re(X)\times\frac{Mp}{Test}=0 \tag{23}$$

则对于每个资源 Res_i 而言，所有的估价值均为 0。则基于历史资源信息的方法变为：无论其可用性、负载情况、网络情况均不考虑，直接使用第一个值为 0 的资源，明显不能达到资源选择的基本要求。本文方法在这种历史资源表不足的情况下使用如下公式来计算资源的评估值：

$$Eval=\begin{cases} rLh+rNd, & rRe=1 \\ 0, & rRe=0 \end{cases} \tag{24}$$

式(24)表示，当资源可用时，资源的评估值为网络延时与资源负载之和；当资源不可用时，评估值为 0。这样仍然可以在实时列表中以窗口为单位选择出一个合适的动态资源。

综上所述，在无历史资源表时的较优资源选择方法上，使用基于混沌反馈控制的动态资源方法显然优于基于历史信息的方法。

(3)资源出现特殊情况下的选择

网络中，服务资源的状态不能确定，典型的情况就是当任务递交到服务资源方时，正在执行任务的服务资源由于断电等原因造成的服务资源失效。此时不能由于这个失效导致全部任务的失败，或者不能由任务发起方不限制等待的时间其重启继续执行任务。那么从资源选择的角度可以实时监控并

重新选择合适的资源。其他方法没有考虑这个问题。因此，该方法对资源的不确定性不加控制。本文方法使用实时监控方式，如果在规定次数 λ 内发送的申请无响应，则返回重新刷新列表，重新选择一个合适的资源执行任务。那么此时本方法对资源的不确定性进行控制，保证了任务总能找到一个合适的服务资源来完成计算。

结束语 本文针对动态资源的几个典型特性，设计并实现了一种基于动态服务资源选择的方法。该方法弥补了基于历史资源信息方法的不足，能够动态可控地选择一个合理服务资源来执行特定任务，加强了服务发现技术的可靠性。但是一些问题仍然存在，典型的情况是历史表的大小按照何种方式来排序才能在查找资源的时候达到最佳的效果；当历史表记录不能作为参考时，窗口方式的较优资源选择对资源选择有很大影响，那么窗口应该选择多少个合适，窗口的合适大小为多少最佳等问题还需要进一步研究。

参 考 文 献

[1] Cheung W K, Liu Ji-ming, Tsang K H, et al. Dynamic Resource Selection For Service Composition in The Grid[C] // IEEE/WIC/ACM International Coference on Web Intelligence (WI'04). IEEE Press, 2004: 412-418

[2] Foste I, Kesselmn C, Nick J. Gird services for distrubted system integration[J]. IEEE Press, 2002, 35(6): 37-46

[3] Maximilien E M, Singh M P. Multiagent System for Dynamic Web Services Selection[C] // Workshop on Service-Oriented Computing and Agent-Based Engineering (SOCABE 2005). Utrecht, The Netherlands, July 2005: 25-29

[4] Roman M, Hess C K, Cerqueira R. et al. Gaia: A Middleware Infrastrueture to Enable Actives Paces[J]. IEEE Pervasive Computing Magazine, 2002, 1(1)

[5] Zeng L Z, Benatallah B, Dumas M. Quality driven Web service composition. In: Proc. of the WWW 2003[C] // Budapest: ACM. 2003: 411-421

[6] Benatallah B, Dumas M, Sheng Q Z, et al. Declarative composition and peer-to-peer provisioning of dynamic Web services[C] // Proc. of the 18th International Conference on Data Engineering. San Jose: IEEE Computer Society, 2002: 297-308

[7] Casati F, Ilnicki S, Jin L J, et al. eFlow: A platform for developing and managing composition e-services[R]. HPL-2000-36. HP Laboratories Palo Alto, 2000

[8] Liu Y T, Anne H H, Zeng L Z. QoS computation and policing in dynamic Web service selection[C] // Proc. of the WWW 2004. New York: ACM, 2004: 66-73

[9] Zhao J F, Xie B, Zhang L, et al. A Web services composition method supporting domain feature[J]. Chinese Journal of Computers, 2005, 28(4): 731-738

[10] 许兰，朱巧明，李培峰，等. 一种基于历史信息反馈的动态服务资源选择模型[J]. 计算机应用，2007，27(9)：2283-2286

[11] 刘书雷，刘云翔，张帆，等. 一种服务聚合中 Qos 全局最优服务动态选择算法[J]. 软件学报，2007，18(3)：646-656

[12] 王勇，代桂平，侯亚荣. 信任感知的组合服务动态选择方法[J]. 计算机学报，2009，32(8)：132-138

[13] 李清，李志蜀，朱明放，等. 基于自适应算法的动态网格服务选择方法[J]. 计算机工程，2007，22(13)：23-25

[14] 丁永生. 智能计算(理论技术与应用)[M]. 北京：科学出版社，2004

(上接第 431 页)

97%，95.8%与 94.3%。并且在相同噪声水平下，约简属性集输入时，PNN 结构故障识别结果明显好于所有 40 个条件属性都输入的情况。

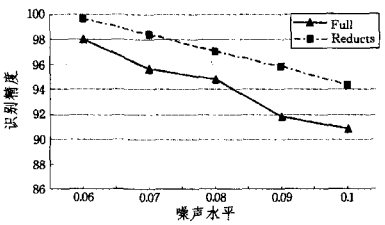

图 3 不同噪声水平下粗糙集识别结果

结束语 本文利用粗糙集约简技术对特征属性进行约简，求得系统的最简特征集合，最后通过概率神经网络进行了结构故障的识别，通过仿真算例分析并对比属性约简前后直升机结构故障识别结果，可以得出下列结论：

1)粗糙集属性约简可以显著减少 PNN 输入层神经元的数目，从而降低网络结构的复杂性，减少样本训练时间。

2)经过粗糙集属性约简，可以获得比全部特征属性输入时更高的识别精度。

3)粗糙集约简技术用于结构故障识别具有很好的抗噪声能力，即便噪声水平为 0.1 时，属性约减后的总识别精度仍为 94.3%，识别技术抗干扰能力增强。

参 考 文 献

[1] Pawlak Z. Rough Sets. Decision Algorithms and Bayes's Theorem[J]. European Journal of Operational Research, 2002, 136: 181-189

[2] 薛定宇，陈阳泉. 高等应用数学问题的 MATLAB 求解[M]. 北京：清华大学出版社，2008

[3] 姜绍飞，姚娟. 基于粗集和数据融合的结构故障识别方法[J]. 工程力学，2009，26(4)：207-213

[4] 韩晓红，胡玉. K-means 聚类算法的研究[J]. 太原理工大学学报，2009，40(3)：236-239

[5] Specht D F. Probabilistic neural networks [J]. International Journal of Neural Networks, 1990, 3: 109-118

[6] 贾清泉，杨奇逊，杨以涵. 基于故障测度概念与证据理论的配电网单相接地故障多判据融合[J]. 中国电机工程学报，2003，23(12)：6-11

[7] Hao L N, Xu X H. Applications of rough set theory in intelligent fault diagnosis[J]. China machine engineering, 2002, 13(21): 1856-1858

[8] 张文修，吴伟志，梁吉业，等. 粗糙集理论与方法[M]. 北京：科学出版社，2003

[9] 王前震，蔡瑞英，等. 概率粗糙集模型在机械故障诊断中的应用[J]. 计算机工程与应用，2009，23(4)：146-151

[10] 何明，冯博琴，马兆丰，等. 一种基于粗糙集的粗糙神经网络构造方法[J]. 西安交通大学学报，2004，18(12)：426-430