

协同进化遗传算法及其应用

张运凯^{1,2}, 王方伟¹, 张玉清³, 马建峰²

(1. 河北师范大学网络信息中心, 石家庄 050016; 2. 西安电子科技大学计算机学院, 西安 710071;

3. 中科院研究生院国家计算机网络入侵防范中心, 北京 100039)

摘要: 介绍了协同进化遗传算法及其实际应用, 大量的实验数据表明, 它的性能明显优于传统的遗传算法, 而且对不同的协同进化方法进行了对比。由此本文提出用协同进化遗传算法来解决入侵响应问题, 以提高响应的速度和效率, 最后预测了协同进化遗传算法的发展方向和

应用。

关键词: 协同进化遗传算法; 适应度; 收敛

Co-evolutionary Genetic Algorithm and Its Applications

ZHANG Yunkai^{1,2}, WANG Fangwei¹, ZHANG Yuqing³, MA Jianfeng¹

(1. Network Center of Hebei Normal University, Shijiazhuang 050016; 2. School of Computer Science and Engineering, Xidian University,

Xi'an 710071; 3. National Computer Network Intrusion Protection Center, GSCAS, Beijing 100039)

【Abstract】 In the paper, a genetic algorithm which coevolves individuals is introduced. Its operation is illustrated on a lot of different applications. Then, it compares the differences between coevolutionary genetic algorithm and traditional genetic algorithms, furthermore different co-evolutionary approaches are compared. Whereas CGA itself has many good performances, a new method is presented to solve the problem of intrusion response. Finally, according to the trends of coevolution development, the paper predicts the future trends of its development and application.

【Key words】 Co-evolutionary genetic algorithm (CGA); Fitness; Convergence

1 协同进化遗传算法

1.1 原理

CGA用两个种群来模拟生物间的“军备竞赛”, 种群间的相互作用是用适应度函数来评价的。CGA把问题的解表示成“染色体”, 在算法中是以二进制表示的编码串。并且, 在执行CGA之前, 产生初始的解种群和测试种群, 然后, 把这些初始的种群置于实际的问题“环境”中。初始解个体的适应度按随机从测试种群中挑选的一定数目的测试个体里面符合的个数计算, 相反, 测试个体的适应度按随机挑选的一定数目的解个体违反的个数计算。按相克相生的原则, 从解种群中选择较适应环境的个体进行复制, 再通过交叉, 变异等遗传操作, 产生更适应环境的新一代解种群。这样, 经过一代一代地进化, 最后就会收敛到最适应环境的一个个体上, 它就是问题的最优解。

1.2 协同进化遗传算法

标准的协同进化遗传算法可以描述如下:

- (1) 初始化, 产生初始解种群和测试种群。
- (2) 初始解个体的适应度按随机挑选的 n 个测试个体中符合的个数计算, 相反, 测试个体的适应度按随机挑选的 n 个解种群中违反的个数计算。
- (3) 从两个种群中分别随机选择 n 个个体进行配对, 并安排遭遇。在每个遭遇中, 如解个体属于符合的情况, 它的适应度就增加 1, 否则为 0。而测试个体的适应度计算正好相反。
- (4) 对解种群进行选择、较差、变异等遗传操作, 产生下一代解种群。
- (5) 如果满足停止准则, 就结束, 输出最佳个体。否则, 转(3)。

为解决更复杂的问题, Potter和De Jong^[9]提出了一个改进的CGA算法——互操作的协同进化遗传算法 (Co-Operative Coevolutionary Genetic Algorithm, CCGA), 采用分而治之的思想把复杂的问题分解成若干较简单的子问题, 对每个子问题进行优化, 最后把子问题和并形成完整的解。

在具体应用CGA时需要面临几个关键问题, 这几个问题处理得好坏, 直接影响到后面解的质量和算法的性能。这些问题包括编码、种群的产生、个体适应度的评价、操作算子、停止准则等。在本文中, 主要介绍个体适应度的评价、操作算子。

(1) 个体适应度的评价

用来度量种群中个体优劣 (符合条件的程度) 的指标值, 它通常表现为数值形式, 是由用户定义的。Angeline^[10]提出了相对适应度的概念。竞争适应度是相对适应度的一个类型。它计算个体的适应度值是通过同其它的个体竞争“决斗”得到的。主要用在博弈论中, 如下棋、迭代囚徒困境 (Iterated Prisoners Dilemma, IPD)。常用的竞争模式有 4 种: 全部-全部、随机、锦标赛、所有-最优个体。可以用于两个或两个以上种群的CGA中。文献[10]提出了一个竞争适应度共享的评价方法, 它可以很好地维持种群的多样性。其基本思想是: 当一个个体 α 打败其对手 β 时, α 收到的增益为 $1/n$, 其中, n 为能够打败 β 的个体的总数。文献[3]给出了一个针对寄生虫个体的适应度评价方法, 在寄主个体中, 对寄生虫个体用重叠方法操作 n 次, 实验数据表明, 此方法有效。针对实际问题存在复杂的因果关系, 并且在评价个体时有很多的组合方式, DONG提出了一个新的评价算法 DCF (Derivative Contribution Feedback), 它的个体不是单独测试适应度, 而是根据它所参与的当前决策从系统中收到反馈。如果在某个阶段, 一个个体的数目增多, 它将有更多的机会参与进化。系统也能通过 Zero-order 和 First-order 反馈影响所有个体的适应度, 当解达到稳态时, First-order 将发挥更大的作用, 能克服问题的高可变性。

基金项目: 国家信息攻防与网络安全保障持续发展计划; 国家“863”计划 (2002AA142151); 中国科学院知识创新工程方向性项目 (KGCX2-106); 北京市科技计划项目 (H020120090530)

作者简介: 张运凯 (1965—), 男, 副教授、博士生, 主研方向为网络安全; 王方伟, 硕士; 张玉清, 副研究员、博士后; 马建峰, 教授、博导

收稿日期: 2004-01-06

E-mail: zhyk@mail.hebtu.edu.cn

(2) 操作算子

操作算子包括个体的选择、交叉、变异等，对算法性能影响最大，直接影响算法的收敛性和解的质量。Aickelin等人对当前的选择策略进行了深入的研究，经过比较，随机/随机（Random/Random, RR）策略性能最好，其做法是：经过两次随机的组合，把其中较好的适应度值保留下来，详见文献[6]。Handa^[1]提出了一个新的交叉操作—Superposition，由两个父代个体只产生一个子代个体，而单点交叉、 n 点交叉都是两个父代个体产生两个子代个体。其思想是：先从两个种群中选择两个父代个体，接着检查它们的相容性，最后合并父代个体的某些特殊位产生子代染色体。下面举例说明：

1	1	0	1	0	1	1	0	0	0	1	1	0	寄主个体	
合 并	*	*	1	0	0	*	*	1	*	0	0	*	1	寄生虫个体
1	1	1	0	0	1	1	1	1	0	0	0	1	1	子代个体

从操作过程可以看出，产生子代时，以寄生虫个体的基因为主，如果某个基因位有值（或0获1），则子代遗传寄生虫个体的基因，否则从寄主个体得到遗传基因。

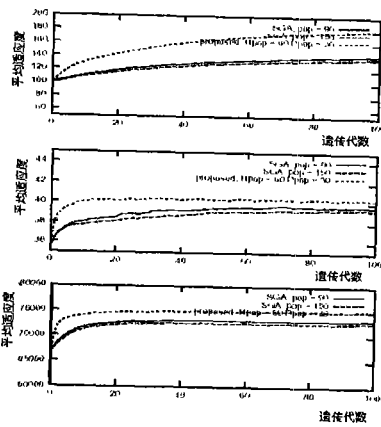
1.3 和常规遗传算法的比较

遗传算法虽然实现简单，操作方便，但是存在很多的缺陷：(1) 很容易导致“早熟”，陷入局部最优。(2) 随着问题规模的增大，其计算复杂度明显增加，收敛性显著降低，搜索问题空间能力也下降。(3) 依靠简单的交叉、变异操作，很容易产生不可行解。(4) 交叉产生的子代可能一个适应度很高，另一个很低，低的个体虽然含有比较好的基因，但是会被淘汰。为此国内外很多外学者进行了深入的研究，提出了一些改进的方法，如采用佳点集交叉、二元变异等操作算子，在性能上有了一些改进，但还不能适应将来大型的复杂问题。为此，一些新的方法应运而生，来克服上述缺点。下面主要用实验数据来说明协同进化遗传算法的有效性。

表1、图1给出了两种算法的比较结果，很明显就可以看出两种算法的优劣：CGA要明显优于GA：计算时间短，收敛速度快，而且收敛精度也比GA高。从表1可以看出CGA在求解分类神经网络训练问题时计算工作量大大减少，同样达到90%的分类精度，CGA的遗传代数只有GA的1/3。在求解Manipulator Path Planning问题时CGA占用CPU的时间只有GA的1/9。

表1 GA和CGA在求解问题时性能的对比

		GA		CGA		参考文献
Manipulator Path Planning 问题	遗传代数	540		750		[4]
	CPU时间 (分)	183		20		
	直线活动距离	79		58		
3D 装箱问题	箱子的数目	4		3		[5]
	空的体积 适应度值	0.55428Vmax 0.855428		0.31181Vmax 0.931181		
分类神经网络 训练问题	>75%	520K		162K		[7]
	>80%	920K		342K		
	>85%	1320K		502K		
	>90%	2420K		762K		
	95%	? 93%		1902K		
	平均收敛分类精度			96.45%		
Approximation 问题	种群规模	Traning	Test	Trainin	Test	[8]
	20	0.317 8	0.450 0	g	0.397 7	
	10	0.301 0	0.430 6	0.156 2	0.389 4	
	6	0.326 7	0.419 9	0.147 7	0.324 1	
	4	0.333 5	0.513 5	0.129 2	0.501 0	
				0.150 4		



(上图为最大值问题,中间为Six Hump Camel Buck函数,下图为Rosenbrock函数)

图1 标准的GA和CGA在函数优化方面性能的比较^[3]

2 遗传算法在入侵响应中的应用

目前的入侵响应系统只能对入侵提供很有限的响应机制，而且绝大部分是人工操作。从入侵开始到响应间存在太长的时间，时间越长，入侵成功的概率就越大，文献[11]用仿真的方法研究了响应时间和和入侵成功率之间的关系，试验表明：如果响应时间为10h，入侵成功的概率为80%；如果响应时间为20h，入侵成功的概率为95%，这就迫切要求提出一种切实可行的方法，来缩短入侵和响应之间的时间差，以便能有效的阻止入侵活动的继续。

文献[12]提出了一种用基于智能决策支持系统的入侵检测与响应方法，其核心是规则发现，用遗传算法完成的。在有入侵时能够检测和自动响应，能够监视网络上不同的活动，如故障、异常、误用、入侵等；能在不同层(如用户层、系统层、过程层、包层)同时监视网络上计算机的活动，根据监视参数偏移正常值的程度来决定入侵的类型，以便产生响应的方法。图2为基于分类器的入侵检测系统的模型。

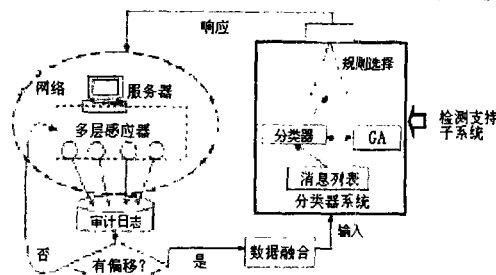


图2 基于分类器的入侵检测系统

在图2中，数据融合模块的作用是整合监视的各参数值，然后作为分类器系统的模板。整个模型的工作原理如下：用多层感应器收集网络状态信息，存入审计日志中，如果监视数据与正常值有偏移，就经过数据融合模块进入分类器系统模块中；否则不做处理，继续监视。分类器系统是一个动态系统，用遗传算法来进化响应规则，它从监视环境接收二进制数据，经过处理产生一个响应动作作为输出，同时更新规则库。分类器把从外部收到消息先放到消息列表中，然后搜索分类器列表，以便能找到一条规则可以匹配当前信息，如果存在多条，就根据规则的强度（或概率）选择一条；否则就利用遗传算法来进化产生一条。分类器的规则采用下面的格式：

<条件>: <动作>

条件部分是一个包含“0”、“1”、“#”字符串,表示监视参数的偏移度,有4个等级;动作部分表示根据攻击类型采取的相应对策,有8个等级。如下例:

#100111#001#1#111#110:5

就是根据条件“#100111#001#1#111#110”采用第五个响应对策。

在文献[12]中,遗传算法主要用来实现规则发现,用遗传算法求解需要解决4个难题:染色体的编码;染色体的解码;适应度的计算;遗传操作算子,主要是交叉和变异。其中染色体的解码和操作算子在此不详述,这里主要介绍染色体的表示和适应度的计算。

2.1 染色体的编码

染色体包含系统要监视的4个层面的参数,每层有5个参数,每个参数用两个二进制位表示,用40位二进制位表示一个完整的个体。如表2所示。

表2 染色体的编码

用户层	系统层	过程层	包层
U ₁ U ₂ U ₃ U ₄ U ₅	S ₁ S ₂ S ₃ S ₄ S ₅	P ₁ P ₂ P ₃ P ₄ P ₅	N ₁ N ₂ N ₃ N ₄ N ₅
00 11 10 11 11	10 01 00 11 10	00 01 10 11 11	10 11 10 01 00

在表2中,第二行表示各层各参数,第三行表示一个染色体的例子。

2.2 适应度的计算

适应度函数的目标就是用来衡量染色体的优劣程度,在本问题中,染色体表示为

$$C = \langle c_1, c_2, \dots, c_{40} \rangle, \quad c_i \in \{0, 1, \# \}$$

为了计算适应度方便,把染色体C分解成四部分,如表3所示。

其中, P^i 表示相对应的层 i ($i=1$ 用户层; $i=2$ 系统层; $i=3$ 过程层; $i=4$ 包层); P_j^i 表示层 i 的第 j 个参数。还做了如下的定义:

$Min(P_j^i)$: P_j^i 可以匹配的最小值,如表4。

表3 染色体的一般形式

P^1	P^2	P^3	P^4
$P_1^1 \dots P_5^1$	$P_1^2 \dots P_5^2$	$P_1^3 \dots P_5^3$	$P_1^4 \dots P_5^4$
$c_1 \dots c_{10}$	$c_{11} \dots c_{20}$	$c_{21} \dots c_{30}$	$c_{31} \dots c_{40}$

表4 $Min(P_j^i)$ 的定义

P_j^i	$Min(P_j^i)$
00	0
01	1
10	2
11	3
0#	0
#0	0
1#	2
#1	1
##	0

相应的可以得到

$$MinVal(P^i) = \frac{1}{5} \sum_{j=1}^5 \frac{Min(P_j^i)}{3} \quad (1)$$

$KnowDist$ 函数用来定义从当前规则各层最小值到某一指

定知识库的距离, $k_i \in [0.0, 1.0]$, 如式(2),

$$KnowDist(C) = \frac{1}{4} \sum_{i=1}^4 |k_i - MinVal(P^i)| \quad (2)$$

Gen 函数用来度量规则中各层通配符(“#”) 在数量上同优化值($OptNum$)的差异程度,其中 $GenCf \in [0.0, 1.0]$, 表明评估适应度时的重要性,如式(3)。

$$Gen(C) = GenCf * \frac{1}{4} \sum_{i=1}^4 |OptNum - Num(P^i)| \quad (3)$$

式(4)就可以用来评价个体的适应度了,很明显可以看出,个体C的适应度值最大为2。

$$Fitness(C) = 2 - KnowDist(C) - gen(C) \quad (4)$$

仿真实验验证了该方法的有效性,能检测内部误用和外部攻击,但也存在一定的不足:分类不够明确,尤其在边界周围,这给制定决策带来了一定的难度。

3 协同进化GA在入侵响应中的应用

通过CGA算法和常规GA算法的比较,可以看出CGA的收敛速度更快,且求解的质量更高,显示了其特有的魅力。从目前查阅的资料来看,还没有发现CGA在自动入侵响应方面的应用。我们打算以文献[12]的模型为基础,通过对CGA进行改进,提出一个新的算法,来比较两者的实验结果,验证在此领域的高效性,以便能提高检测入侵的成功率和响应的准确度。

下一步要做的工作如下:(1)染色体的表达和适应度值的计算;(2)保持种群的多样性;(3)确定有用信息:高适应度或未发现的小生境;(4)竞争和协作是否可共用;(5)开发层次CGA,提高收敛速度;(6)算法的终止条件研究。

通过实验选择合适的算法参数:变异概率、交叉概率,种群规模等。

4 发展方向

综上所述,协同进化遗传算法作为一种新的全局优化搜索算法,搜索过程作用于编码了的个体。所以它没有求导和函数连续性的限制,具有鲁棒性、随机性、全局性以及适于并行处理的优点,在分类、约束满足、优化调度、聚类等领域的应用上展现了它特有的优势,也展示了它潜力和宽广前景;但同时,它在理论和应用技术上也存在着一些不足和缺陷,比如在进化策略方面,数学基础很薄弱,缺乏深刻且具有普遍意义的理论分析;需要CGA的通用编程和形式等。随着理论研究的深入,可以肯定,协同进化遗传算法以其特有的特点在许多实际问题中的应用会越来越广泛,以帮助我们及时、精确地解决实际的复杂难题;同时,随着广泛的数学方法和强大的计算机模拟工具的出现,必将推动它的研究取得长足的进展。

参考文献

- 1 Allaby M. The Oxford Dictionary of Natural History. Oxford, English: Oxford University Press, 1985: 150
- 2 Paredis J. Coevolutionary Life-time Learning. Parallel Problem Solving From Nature IV. Proceedings of the International Conference of Evolutionary Computation, Voigt H M(ed), Spring Verlag, 1996
- 3 Handa H, Baba N, Katai O, et al. Coevolutionary Genetic Algorithm with Effective Exploration and Exploitation of Useful Schemata. In: Proceedings of the 1997 International Conference on Neural Information Processing and Intelligent Information Systems, 1997, (1): 424
- 4 Ali M, Babu N, Varghese K. Offline Path Planning of Cooperative Manipulators Using Co-evolutionary Genetic Algorithms. Indian

(下转第43页)

3.2 通信软件设计

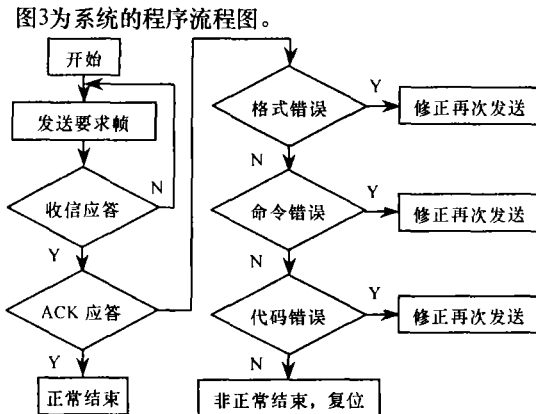


图3 程序流程图

```
//获取系统所有的通讯端口列表
portList = CommPortIdentifier.getPortIdentifiers();
while (portList.hasMoreElements()) {
    portId=(CommPortIdentifier) portList.nextElement();
    //选定COM1端口作为通讯口
    if (portId.getName().equals("COM1")) {
        //打开COM1口
        serialPort=(SerialPort)portId.open("InterVerApplet", 2000);
        //设置端口事件监听器
        serialPort.addEventListener(this);
        serialPort.setSerialPortParams(9600,
        SerialPort.DATABITS_8,SerialPort.STOPBITS_1,
        SerialPort.PARITY_NONE); //设定通讯参数
        ...
    }
}
```

☆☆

(上接第40页)

- Institute of Technology and Arizona State university. IEEE Transactions on Computers, 2002: 415-424
- 5 Pimpawat C, Chaiyaratana N. Using a Co operative Co-evolutionary Genetic Algorithm to Solve a Three-dimensional Container Loading Problem. The 2001 Congr. on Evolutionary Computation, Seoul, Korea, 2001.5(2): 1197-1204
- 6 Aickelin U, Bull L. On the Application of Hierarchical Coevolutionary Genetic Algorithms: Recombination and Evaluation Partners. Journal of Applied System Sciences (JASS), Cambridge Press, ba (tba), 2002: 1-15
- 7 王小平, 曹立明. 遗传算法——理论、应用与软件实现. 西安交通大学出版社, 2002-06: 156
- 8 Wang W, Yi W, He Y. Cooperative Coevolutionary Fuzzy Systems.

```
public class InterverApplet extends Applet implements
    SerialPortEventListener, ActionListener{...
```

```
outputStream = serialPort.getOutputStream();
outputStream.write(messageString.getBytes());
```

```
public void serialEvent( SerialPortEvent event) {
switch(event.getEventType()) {
...

//有新数据到达
case SerialPortEvent.DATA_AVAILABLE:
//设置输入缓冲区
byte[] readBuffer = new byte[29];
//接收数据
while (inputStream.available() > 0) {
int nB=inputStream.read(readBuffer);
}
...
break;
}
}
```

4 结束语

用Java语言编写的串行通信程序可以制作成独立的应用程序,还可以是Applet(小应用程序),其应用前景十分可观。

要实现基于Web的远程控制,需要采用Java Applet程序。需要注意的是Java Applet程序涉及到签名认证等安全性问题,这在后续论文中将会进行深入探讨。

参考文献

- 1 刘其成, 张民, 夏方遒等. Java语言基础教程. 东营: 石油大学出版社, 2001
- 2 刘其成, 王培进. 面向对象柔性设计与控制理论(II). 计算机工程与应用, 2002, 38(18): 64
- 3 刘其成, 肖明, 高坚. Java编程中的中文问题及其解决方法. 计算机系统应用, 2003, 6: 68
- 4 三菱电机株式会社编. 变频器原理与应用教程. 北京: 国防工业出版社, 1998

International Conference on Neural Information Processing 2001,
Shanghai, 2001-11:1-6

- 9 Potter M A, De Jong K A. A Cooperative Coevolutionary Approach to Function Optimization. In: Proceedings of the Third International Conference on Parallel Problem Solving from Nature, Jerusalem, Israel, 1994: 249-257
- 10 Ficij S G, Pollack J B. Game Theory and the Simple Coevolutionary Algorithm: Some Preliminary Results on Fitness Sharing. *Literature Search on Evolutionary Computation*, Conference paper, 2001-07-2-7
- 11 Cohen F B. Simulating Cyber Attacks, Defense, and Consequences. <http://all.net/journal/ntb/simulate/simulate.html>, 1999-05-13
- 12 Dasgupta D, Gonzalez F A. An Intelligent Decision Support System for Intrusion Detection and Response. *MMM-ACNS 2001*: 1-14