改进的混合蛙跳算法在传感器配置优化中的应用

刘晓芹[1,2] 黄考利[1] 安幼林[1] 吕晓明[1]
(军械工程学院 石家庄 050003)[1] (陆军航空兵学院 北京 101123)[2]

摘 要 传感器配置优化是可测性设计的重要研究内容,将混合蛙跳算法应用于传感器配置优化是一种新的尝试。针对传感器配置优化属于离散问题求解,提出离散的混合蛙跳算法,设计了一种离散化的更新方式。为克服蛙跳算法的早熟收敛问题,在改进的离散蛙跳算法中采用混沌优化算法以概率的形式对全局极值进行了优化。最后通过具体系统实例验证了该方法的正确性和有效性。

关键词 传感器配置优化,混合蛙跳算法,混沌优化算法
中图法分类号 TP301 **文献标识码** A

Application of Improved Shuffled Frog Leaping Algorithm in Optimum of Sensor Location

LIU Xiao-qin[1,2] HUANG Kao-li[1] AN You-lin[1] LU Xiao-ming[1]
(Ordnance Engineering College, Shijiazhuang 050003, China)[1] (Army Aviation Institute of PLA, Beijing 101123, China)[2]

Abstract Optimum of sensor location is an important research field in testability design, and it is a new attempt to use shuffled frog leaping algorithm for optimum of sensor location. Considering the optimal problem of sensor location is set in a space featuring discrete, a discrete shuffled frog leaping algorithm was proposed, and the change in position was redefined discretely. To avoid converging too fast, the algorithm was improved. Chaos optimization algorithm was used to optimize the best solution in the form of probability. An example and simulation results were provided to verify the effectiveness and practicability of this approach.

Keywords Optimum of sensor location, Shuffled frog leaping algorithm, Chaos optimization algorithm

传感器配置是测试性设计的重要内容之一。随着系统集成规模的增加,传感器配置开销越来越大,如何在满足系统可测性要求下尽可能地降低传感器开销这一问题也越来越突出[1]。因此,有必要对传感器配置进行优化设计。

传感器配置优化是一个NP难问题[2],多采用启发式搜索方法优化求解[3,4]。混合蛙跳算法(Shuffled Frog Leaping Algorithm, SFLA)结合了遗传算法和粒子群优化算法的优点[5],具有概念简单、参数少、计算速度快、全局寻优能力强、易于实现等特点,并成功地解决了NP完全问题[6]。本文尝试将蛙跳算法应用于传感器配置模型的优化问题中。

依据配置开销最小的原则,建立了一个代价最优的传感器配置模型(即目标函数),将IEEE Std1522标准中的诊断性能指标具体化,得到配置模型的约束条件。传感器配置模型的优化求解属于离散优化问题,针对SFLA不适合离散问题求解的缺陷,以及其自身收敛速度慢和易陷入局部最优解等缺陷对SFLA加以改进,提出基于混沌优化算法(Chaos Optimization Algorithm, COA)的离散蛙跳算法。通过polybox系统实例和结果对比分析,验证了改进算法的有效性。

1 传感器配置的形式化描述及模型

1.1 传感器配置的形式化描述

传感器配置问题的形式化描述如下:

(1)$S=\{s_1,s_2,\cdots,s_m\}$是可供选择的传感器集合,m是备选传感器总个数。

(2)$V=\{v_1,v_2,\cdots,v_n\}$表示n个故障源。

(3)$D=[d_{ij}]$是$m\times n$的二进制矩阵,它描述了故障源集V和传感器集S之间的关系,其中,如果传感器s_j可以检测故障源v_i,则$d_{ij}=1$,否则$d_{ij}=0$。

(4)$P(V)=\{p(v_i)\}$,$i=1,2,\cdots,n$,表示与故障源关联的先验故障概率。

(5)$Sf=\{P(s_j)\}$,$j=1,2,\cdots,m$,$P(s_j)$表示传感器s_j发生故障的概率,用Sf_j表示。

(6)$Pd=\{Pd_{ij}\}$,$i=1,\cdots,n$,$j=1,2,\cdots,m$,Pd_{ij}表示s_j能够检测故障源v_i故障的概率。

1.2 传感器配置模型

系统的可测性设计一般要考虑3个诊断性能指标约束:

到稿日期:2010-03-31 返修日期:2010-06-23 本文受军队保障科研项目"新型地空导弹装备测试性分析与验证技术研究"资助。
刘晓芹(1982—),女,博士生,主要研究方向为武器系统的故障诊断与预测技术,E-mail:liuxiaoqin1121@163.com;黄考利(1958—),男,教授,博士生导师,主要研究方向为装备测试性设计与分析;安幼林(1980—),博士,主要研究方向为武器装备自动测试技术与系统;吕晓明(1983—),男,博士生,主要研究方向为测试性设计与分析。

检测率约束 $FDR \geqslant D_{\text{request}}$，分辨率约束 $FIR \geqslant I_{\text{request}}$，虚警率约束 $FAR \leqslant A_{\text{request}}$，其中 D_{request}，I_{request}，A_{request} 分别为系统应满足的故障检测率、故障隔离率和虚警率。因此，传感器配置必须满足这些指标约束。首先对这 3 个诊断性能指标进行分析。

在对诊断性能指标进行定量定义之前，首先定义一个二进制向量 $X=\{x_j\}$，$j=1,\cdots,m$，表示传感器的选择情况。若 $x_j=1$，表示传感器 s_j 被选择使用；否则 $x_j=0$，表示不被选择。

(1)故障检测率约束

每个故障的故障检测率为

$$Pd_i=\frac{\sum_{j=1}^{m}x_j d_{ij}Pd_{ij}(1-Sf_j)}{\sum_{j=1}^{m}x_j d_{ij}} \tag{1}$$

对于任意的 i，式(1)中分母都不能为 0，否则表示故障 i 不能被检测。

根据 IEEE Std1522 标准，故障检测率应满足

$$\frac{\sum_{i=1}^{n}p(v_i)Pd_i}{\sum_{i=1}^{n}p(v_i)} \geqslant D_{\text{request}} \tag{2}$$

(2)故障分辨率约束

在测试性设计时，不仅要检测故障，而且要对检测到的故障进行隔离和诊断。对于任意两个故障源 v_i 和 v_k，假设它们在 D 矩阵中对应的行向量分别为 d_i 和 d_k，那么单故障分辨率条件可描述为

$$\begin{cases} d_i \cdot d_k \leqslant \text{Max}(|d_i|_2,|d_k|_2)-1 \cdot \forall v_i,v_k \in V, v_i \neq v_k \\ \dfrac{\sum_{i=1}^{n}[p(v_i)\prod_{j=1}^{m}(1-x_j d_{ij}Sf_j)]}{\sum_{i=1}^{n}[p(v_i)(1-\prod_{j=1}^{m}Sf_j^{x_j d_{ij}})]} \geqslant I_{\text{request}} \end{cases} \tag{3}$$

式中，函数 $\| \cdot \|_2$ 是求向量的 2 范数。

由于存在一个传感器可以检测多个故障，一个故障也可被多个传感器检测的情况，因此故障隔离问题远比故障检测问题复杂。大多数文献只考虑两个故障的故障隔离问题，本文考虑多故障的分辨率在单故障可分辨的基础上增加如下条件

$$d_i \cdot d_k \neq \text{Min}(\| d_i \|_2, \| d_k \|_2) \tag{4}$$

(3)虚警率约束

根据 IEEE Std1522 标准，传感器虚警率约束为

$$\frac{\sum_{i=1}^{n}[(1-p(v_i))\prod_{j=1}^{m}Sf_j d_{ij}x_j]}{\sum_{i=1}^{n}[(1-p(v_i))\prod_{j=1}^{m}(1-Sf_j d_{ij}x_j)]} \leqslant A_{\text{request}} \tag{5}$$

在测试性设计过程中，应使所选的传感器在满足上述 3 个诊断性能指标的前提下，价格和失效概率最低，同时要保证指定的必须检测的故障被检测，指定的必须隔离的故障被隔离。因此，传感器配置模型可描述为

求解域：$x_j \in \{0,1\}$，$j=1,2,\cdots,m$

目标函数：$\text{Min}\sum_j C_j x_j$

约束条件：

$$\begin{cases} \sum_{j=1}^{m}x_j d_{ij} \geqslant 1, \forall v_i \in V_s \\ \dfrac{\sum_{i=1}^{n}p(v_i)Pd_i}{\sum_{i=1}^{n}p(v_i)} \geqslant D_{\text{request}} \\ \dfrac{1}{m}\sum_{j=1}^{m}d_{ij}Pd_{ij}>0 \\ d_i \cdot d_k \leqslant \text{Max}(\| d_i \|_2, \| d_k \|_2)-1 \\ d_i \cdot d_k \neq \text{Min}(\| d_i \|_2, \| d_k \|_2) \\ \dfrac{\sum_{i=1}^{n}[p(v_i)\prod_{j=1}^{m}(1-x_j d_{ij}Sf_j)]}{\sum_{i=1}^{n}[p(v_i)(1-\prod_{j=1}^{m}Sf_j^{x_j d_{ij}})]} \geqslant I_{\text{request}} \\ \dfrac{\sum_{i=1}^{n}[(1-p(v_i))\prod_{j=1}^{m}Sf_j d_{ij}x_j]}{\sum_{i=1}^{n}[(1-p(v_i))\prod_{j=1}^{m}(1-Sf_j d_{ij}x_j)]} \leqslant A_{\text{request}} \end{cases} \tag{6}$$

2 混合蛙跳算法的改进

2.1 混合蛙跳算法

2000 年，Eusuff 和 Lansey 通过类比青蛙的觅食行为与优化问题求解的相似性，提出了一种新的求解 NP 难题的智能优化方法——混合蛙跳算法。SFLA 结合了遗传算法和粒子群优化算法的优点，具有概念简单、参数少、计算速度快、全局寻优能力强、易于实现等特点，并成功地解决了 NP 完全问题。

SFLA 中，种群位置更新方式为

$$D_k^i = r_1(x_k^g - x_k^w) \tag{7}$$

$$x_k^{w'} = x_k^w + D_k^i (-D_{\max} \leqslant D^i \leqslant D_{\max}) \tag{8}$$

式中，$r_1 \in U(0,1)$，$D_{\max}$ 表示青蛙的最大移动步长。x_k^g，x_k^w 分别表示第 k 个子群中适应度函数值最好和最差的蛙的位置；$x_k^{w'}$ 表示第 k 个子群中适应度函数值最差的蛙更新后的位置。

SFLA 的优化步骤如下：

1. 参数赋初值并初始化种群：种群的规模 L、重新组成的子群数目 K、最大迭代次数 T、算法终止条件等等，随机选取 F 只蛙。

2. 计算每个个体的适应度函数值。

3. 将 F 只蛙按适应度函数值降序排列分为 K 个子群，分配原则为：第一只蛙在第一个子群中，直到第 K 只蛙进入第 K 个子群，然后第 $K+1$ 个蛙又进入第一个子群，循环分配，到所有蛙分配完毕。

4. 对每一个子群中的蛙找出其中的最优个体和最差个体，找出群体最优个体，在指定迭代次数内按式(7)和式(8)提高最差个体的目标函数值。

5. 对更新后的子群进行混合，作为一个群体，取代原来的群体。

6. 判断迭代次数是否满足。如果满足，结束迭代，输出最优目标函数值的相关信息；否则转向步骤 2。

2.2 离散混合蛙跳算法

在 SFLA 中，种群的更新方式仅适合连续问题的求解。而传感器配置模型的优化属于离散优化问题，需要设计具体的更新算子。更新过程更改如下：比较 x_k^g 和 x_k^w 编码中不同二进制位的位数 p。若 p 大于 $D_{\max}$，则更新为式(9)；若 p 小

于等于 D_{max}，则这 p 个不同的二进制位根据随机数 r_i（$i=1,2,\cdots,p$）更新，若 $r_i \leqslant 0.5$ 则不更新，若 $r_i > 0.5$ 则更新，设第 i 个不同二进制位为 $x_k^w(i)$，则更新方式为式(10)

$$x_k^{w'} = x_k^b \tag{9}$$

$$x_k^w(i) = (x_k^w(i)+1) \bmod 2, i=1,2,\cdots,p \tag{10}$$

采用更新方式(9)、式(10)的混合蛙跳算法称为离散蛙跳算法(Discreet SFLA)，简称 D-SFLA。

2.3 离散混合蛙跳算法的改进

种群的更新方式使 x_k^w 的空间位置在更新前后发生较大的变化。随着变量维数的增加，这种变化也就越大，能够扩大解空间的搜索范围。其缺点是：①搜索范围的扩大容易跳过全局最优解，减缓算法收敛速度，从而一定程度上增加了为提高 SFLA 的收敛效果而对混合迭代次数的需求，也就大大增加了计算时间的长度。对于复杂问题甚至可能搜索不到最优解。②当蛙向当前最佳蛙的位置靠拢到非常接近零时，群体的多样性就会慢慢丧失，所有的蛙就会停止移动，导致不能在其解空间内继续搜索，陷入局部最优解，即所谓的“早熟”收敛。为此，本文引入混沌优化算法[7]对 SFLA 算法加以改进。

2.3.1 混沌优化算法

COA 算法利用类似载波的方法将混沌状态引入到优化变量中，并把混沌运动的遍历范围放大到优化变量的取值范围，然后利用混沌变量进行搜索。用于载波的混沌变量通常选用 Logistic 映射

$$z_{l+1} = \mu \cdot z_l(1-z_l), l=0,1,2,\cdots \tag{11}$$

式中，$0<z_l<1$，μ 为控制参量，当 $\mu=4$ 时，系统没有稳定解，是[0,1]区间的满映射，呈现完全的混沌状态。

对于一类连续对象的优化问题

$$\begin{aligned} &\min\ f(x_i), i=1,2,\cdots,n \\ &st.\ a_i \leqslant x_i \leqslant b_i \end{aligned} \tag{12}$$

算法的基本步骤如下：

1. 利用混沌状态对初值敏感的特性，对式(11)中的 z_l 赋予 n 个（n 为优化问题中自变量的个数）具有微小差异的初值，可获得 n 个不同轨迹的混沌变量 $z_{i,l+1}$。

2. 利用式(13)的载波表达式，将混沌变量映射到优化变量的取值范围中。

$$x_{i,l+1} = c_i + d_i \times z_{i,l+1} \tag{13}$$

式中，$c_i = a_i$，$d_i = b_i - a_i$。

3. 进行混沌搜索。令 $x_i(k) = x_{i,l+1}$，求 $f(x_i(k))$。当 $k=1$ 时，设函数的最优值为 $f^b = f(x_i(1))$，然后进行如下搜索：

if $f(x_i(k)) \leqslant f^b$

$f^b = f(x_i(k))$;

$x^b = x_i(k)$

end

$k = k+1$;

4. 若经过若干步迭代后 f^b 保持不变，其值则可能是最优解。

2.3.2 改进的离散蛙跳算法的基本思想

基于 COA 的改进的蛙跳算法的基本思想是：以 SFLA 算法为主流程算法，将 COA 嵌入到 SFLA 中，对每次迭代的群体极值 x_g 以概率 Pg 进行混沌优化，指导种群向最优解方向搜索。同时，由于对 x_g 进行了混沌优化，也避免了算法的早熟现象，使 x_g 在解空间内重新搜索，引导 x_g 迅速跳出局部最优解。由于算法在后期搜索中搜索速度变慢，因此搜索初期以小概率对 x_g 进行混沌优化；搜索后期，以接近 1 的概率对 x_g 进行混沌优化。调用混沌优化的概率 Pg 按下式自适应变化

$$Pg = 1 - \frac{1}{1+\ln t} \quad (t \text{ 为迭代次数}) \tag{14}$$

需要注意的是，COA 算法嵌入到蛙跳算法中，应进行离散化处理，过程如下：1. 随机生成一个[0,1]之间的数作为混沌变量初值，并利用 Logistic 映射生成一条混沌轨迹；2. 在混沌轨迹上取 M 个值，使这 M 个值均匀分布在[0,1]区间内；3. 将步骤 2 中得到的值转化为相应的二进制数作为蛙的位置。具体转化方法如下：对于 $z_i \in [0,1]$，利用式(15)将 z_i 放大到 Z，然后对 Z 取整，将 Z 转化为二进制数，这就是 z_i 对应的蛙。

$$Z = (2^N - 1) \cdot z_i \ (N \text{ 为蛙的位置编码的长度}) \tag{15}$$

COA 算法的逆离散化过程为

$$z_i = Z/(2^N - 1) \tag{16}$$

2.3.3 改进的离散蛙跳算法的流程

改进的 D-SFLA 算法称为 CD-SFLA，其具体流程如图 1 所示。

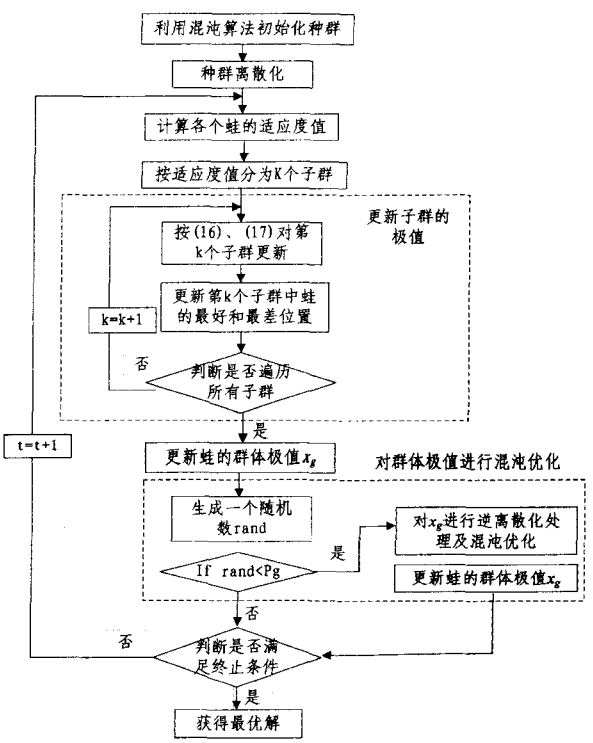

图 1 改进的离散蛙跳算法的流程

3 改进的蛙跳算法在传感器配置中的应用

在传感器配置优化模型中应用改进的蛙跳算法，应解决的问题有：

(1)青蛙个体规模的大小为备选传感器的个数。种群中蛙的位置由长度为 M 的二进制码构成，每个蛙对应一个求解问题。设第 i 个蛙 I_i 的二进制编码为 $I_i = [b_{1i}, b_{2i}, \cdots, b_{Mi}]$，

其中 $b_{ji}=1$(或 0)表示备选传感器集 S 中测试 i 被选中(或不被选中),$i\in\{1,2,\cdots,L\}$,$j\in\{1,2,\cdots,M\}$,L 为粒子群中的粒子规模。

(2)更新方式(10)中 $D_{\max}$ 的选取非常重要,过小会倾向于在局部范围内搜索,过大会影响到算法的效率。由于其值极大地依赖于粒子的规模,如果粒子的规模较大,一般 $D_{\max}$ 的值也要稍大一些。对于本文第 4 节的验证系统,粒子的规模为 7。在实际求解过程中,通过多次实验发现,值取为 3 的结果最为理想。图 2 描述了 $D_{\max}$ 值对求解结果的影响。

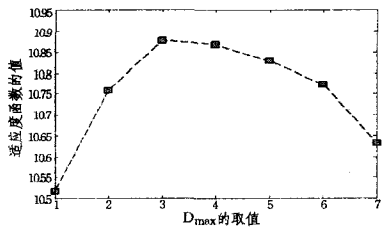

图 2 $D_{\max}$ 值对求解结果的影响

(3)由于目标函数中存在不等式约束,求解时需要构造有效的适应度函数(式(17))。引入权重因子和惩罚函数进行无约束处理,通过调整权重因子和惩罚函数,可增加算法的收敛性。

$$f(I_i,Q_i)=\lambda_1\cdot\left(\frac{FDR(I_i)}{D_{\text{request}}}\right)^2+\lambda_2\cdot\left(\frac{FIR(I_i)}{I_{\text{request}}}\right)^2-\lambda_3\cdot\left(\frac{FAR(I_i)}{A_{\text{request}}}\right)^2-\mu\cdot\frac{\sum_j C_j x_j}{\sum_j C_j}+Q_i\cdot\max\left(0,\min_k\left(\sum_{j=1}^{m}x_j d_{kj}\right)\right) \tag{17}$$

式中,$u,\lambda_1,\lambda_2,\lambda_3$ 为权重系数,调节各因子的权重;Q_i 为惩罚函数,满足 $Q_{i+1}\geqslant Q_i$(本文取 $Q_{i+1}=4Q_i$),当前解不能在检测所有故障时给予惩罚。

4 实例验证

利用本文的方法,对著名的 polybox 系统(见图 3)进行验证。系统由 7 个组元构成,对应的变量分别为 $V=\{x,y,z,t,f,g,h\}$。假定每个组元对应一个故障,故障概率为 $P(V)=\{0.2,0.03,0.02,0.01,0.01,0.001,0.001\}$,其对应的传感器集为 $S=\{s_1,s_2,s_3,s_4,s_5,s_6,s_7\}$。传感器的测试代价为 $C=\{400,600,500,400,600,300,200\}$,传感器的故障概率为 $Pf=\{0.01,0.1,0.01,0.001,0.002,0.003,0.001\}$。表 1 为传感器的检测概率。故障源与备选传感器的相关性矩阵为

$$D=\begin{bmatrix}1&0&0&0&0&0&0\\0&1&0&0&0&0&0\\0&0&1&0&0&0&0\\0&0&0&1&0&0&0\\1&1&0&0&1&0&0\\0&1&1&0&0&1&0\\0&0&1&0&0&0&1\end{bmatrix}$$

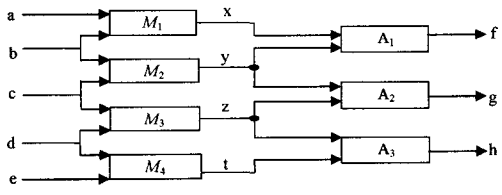

图 3 polybox 系统的混合运算电路

表 1 传感器的检测概率

传感器 \ 故障源	M_1	M_2	M_3	M_4	A_1	A_2	A_3
s1	0.96	0	0.06	0	0.08	0.16	0
s2	0	0.95	0.09	0.15	0	0	0.08
s3	0.14	0.91	0.98	0.045	0	0.05	0
s4	0	0	0.09	0.85	0	0	0.14
s5	0.94	0.81	0	0.045	0.90	0.05	0
s6	0.04	0.99	0.93	0.2	0	0.94	0
s7	0	0.1	0.92	0	0.18	0	0.92

设定测试性设计指标如下:检测率 FDR:95%;隔离率 FIR:94%;虚警率 FAR:2%;实验参数设置:$M=7$,$L=10$,$T=50$,$u=1$,$\lambda_1=5$,$\lambda_2=3$,$\lambda_3=0.5$,$Q_1=0.05$。利用本文的 CD-SFLA 算法求解,同时利用 D-SFLA 算法进行求解。

表 2 记录了 CD-SFLA 和 D-SFLA 在迭代过程中得到的优化结果的对比。可以看出,CD-SFLA 在第 30 次迭代时就达到了全局最优解,而 D-SFLA 在第 40 次迭代时才接近全局最优解,说明 CD-SFLA 算法的运算时间要明显少于 D-SFLA。另外,本文还利用文献[4]中粒子群优化算法进行优化求解,适应度函数仍采用式(17),其运算时间与 D-SFLA 方法的运算时间差不多(见表 2),进一步说明了改进算法的有效性,也表明混合蛙跳算法是适用于求解传感器配置问题的。

表 2 优化性能比较

算法	迭代次数	平均适应度函数值
D-SFLA	10	5.29
	20	7.74
	30	9.57
	40	10.42
	50	10.65
CD-SFLA	10	6.98
	20	9.15
	30	10.67
	40	10.88
	50	10.88
粒子群优化算法	10	5.30
	20	7.73
	30	9.57
	40	10.50
	50	10.67

选用的传感器集合为$\{s_1\ s_4\ s_5\ s_6\ s_7\}$,测试总代价为 1900。由求解结果可以看出,优化后的系统均满足测试性指标要求,选用传感器的个数明显少于备选传感器的个数,减少了测试费用,降低了系统可测性设计的复杂度,证明了改进的蛙跳算法应用于传感器配置模型优化中的有效性。

结束语 本文提出将混合蛙跳算法应用于传感器配置优化中,对更新方式进行了改进,对参数的选取进行了研究,并将混沌优化算法嵌入到蛙跳算法的寻优过程中。系统实例验证表明,改进的蛙跳算法提高了算法的收敛速度,可以有效地应用于传感器配置优化问题中。

参 考 文 献

[1] 安幼林. 面向综合诊断的装备诊断设计关键技术研究[D]. 石家庄:军械工程学院,2009:91-92

[2] Jiang Shengbing, Kumar R. Optimal Sensor Selection for Discrete-event Systems with Partial Observation [J]. IEEE Transactions on Automatic Control, 2003, 48(3): 369-381

(下转第 81 页)

目的节点 $L_{x,y}$ 丢弃，不会把 RREP 返回给源节点 $L_{i,j}$。当然，这条最优路由路径不一定是距离最短的。

(4)路由维护

当一条路由在其生命周期内没有出现业务，则该路由在路由表中被标为无效路由。无效路由上收到数据，将引起卫星节点产生错误消息 ERR。该 ERR 消息将沿着反向路径发往源卫星节点。

当某个有效路由上的链路由于卫星节点移动出其逻辑位置而损坏，利用错误消息 ERR 进行报告。所有 ERR 消息均要求报告节点进行签名。

(5)证书吊销

当某个卫星节点由于出现故障或遭受恶意攻击而需要吊销一个证书时，MEO 卫星向注册在其下的 LEO 组网络发送广播消息，通知此吊销信息。假设被吊销的证书为 $CertL_{m,n}$，则吊销广播消息为：

$M_{i,j}\rightarrow Broadcast:[Revoke,CertL_{m,n}]KM_{i,j}^-$

收到此消息的任何节点向它的近邻广播此消息，并将此吊销通知存储下来，直到吊销证书过期为止。

4 路由协议安全性证明

在本文协议中，处于任何配置(config)下，对于任意的现实世界攻击者 A，令理想世界攻击者 $A'=A$。下面将证明 C' 读取的路由回复消息中不可模糊路由的数目可以忽略。

假定存在一条不可模糊路由回复信息 $(l_1,l_2,\cdots,l_n)$ 被 C' 读取，并且 $v_1,v_2,\cdots,v_k$ 是卫星网络节点序列，使得①$j_1+j_2+\cdots+j_k=n$；②标识符序列可以按照顺序分隔为 k 份，第 i 份包含于 $L(v_i)$，其中 $1\leqslant i\leqslant k$。因为该路由消息是不可模糊路由，那么以下两种情况至少有一种发生：

ⅰ)存在两个顶点 v_{i-1} 和 v_i，其中 $2\leqslant i\leqslant k$，使得 $\{l_{j-1}\}=L(v_{i-1})$，$\{l_j\}=L(v_i)$，并且 l_{j-1} 和 l_j 是两个不相邻的诚实卫星节点的标识符。

ⅱ)存在一个划分 $\{l_j\},\{l_{j+1},l_{j+2},\cdots,l_{j+q}\},\{l_{j+q+1}\}$，其中 $1\leqslant j<j+q\leqslant n-1$，使得 l_j 和 l_{j+q+1} 是诚实卫星节点的标识符，$\{l_{j+1},l_{j+2},\cdots,l_{j+q}\}$ 是攻击者控制的节点标识符，并且 l_j 和 l_{j+q+1} 至少有一方没有攻击者控制的邻居节点。

(1)在第ⅰ)种情况中，因为 l_{j-1} 和 l_j 都是诚实的卫星网络节点，根据协议，当 l_j 发现收到的路由请求信息的最后一个签名是与自己不相邻的卫星节点 l_{j-1} 的签名时，l_j 应该抛弃这条路由信息，而不是在该信息后加上自己的签名，继续转发该消息，所以如果存在这样的路由信息，必定是攻击者伪造了 l_j 的签名。

(2)现在分析第ⅱ)种情况。我们先假定攻击者不能伪造任何诚实节点的签名。诚实节点不会伪造其他参与方的签名，所以 l_{j+q} 的签名一定是攻击者生成的。由于 l_{j+q+1} 为诚实节点，当收到的路由请求消息的最后一个签名是 l_{j+q} 签名时，除非 l_{j+q+1} 和攻击者控制的节点相邻，否则 l_{j+q+1} 应该抛弃该路由请求信息，而不是加上自己的签名继续转发。另一方面，因为不能伪造 l_j 的签名，为了生成发给的 l_{j+q+1} 路由请求信息，攻击者必然要收到最后由 l_j 签名的消息，这样就导致 l_j 和 l_{j+q+1} 同时都是攻击者控制节点的邻居节点。所以假定不成立。如果存在这样的路由信息，攻击者必须伪造 l_j 的签名。

综上可知，如果理想模型中出现可模糊路由，攻击者必然能够伪造诚实卫星节点的签名。因为本协议中采用的签名算法是不可伪造的，所以本协议是统计安全的。

结束语 空间信息网络环境下的卫星网络路由消息的安全性日益重要。本文在分析了卫星网络路由的研究现状、卫星网络拓扑结构和卫星网络安全路由需求的基础上，研究了卫星网络路由协议，提出了一种基于 MEO/LEO 双层卫星网络的认证路由协议。该认证路由协议经过扩展还可以应用于 GEO/MEO/LEO 三层卫星网络，其在一定程度上提高了卫星网络路由消息的安全性。

参 考 文 献

[1] Werner M, Delucchi C, Vogel H-J, et al. ATM-Based Routing in LEO/MEO Satellite Networks with Intersatellite Links[J]. IEEE Journal on Selected Areas in Communications, 1997, 15(1): 69-82

[2] Chang Hong Seong, Kim Byoung Wan. FSA-based link assignment and routing in low earth orbit satellite networks[J]. IEEE Transactions on Vehicular Technology, 1998, 47(3): 1037-1048

[3] Gounder VV, Prakash R, Abu-Amara H. Routing in LEO-based satellite networks[C]// Proceedings of IEEE Emerging Technologies Symposiumon Wireless Communications and Systems. Richardson, 1999: 91-96

[4] Hashimoto Y. Design of IP-based routing in a LEO satellite network[C]// Proceedings of the 3rd International Workshop on Satellite-based Information Services. Dallas, 1998: 81-88

[5] Ekici E, Akyildiz I F, Bender M D. A distributed routing algorithm for datagram traffic in LEO satellite networks[J]. IEEE/ACM Transactions on Networking, 2001, 9(2): 137-147

[6] Akyildiz I F, Ekici E, Bender M D. MLSR: A Novel Routing Algorithm for Multi-layered Satellite IP Networks[J]. IEEE/ACM Transactions on Networking, 2002, 10(3): 411-424

[7] 孙利民，卢泽新，吴志美. LEO 卫星网络的路由技术[J]. 计算机学报，2004，27(5)：659-667

[8] Sanzgiri K, Dahill B, Levine B N, et al. Belding-Royer, E. M.. A secure routing protocol for ad hoc networks[C]// Proceedings of 2002 IEEE International Conference on Network Protocols (ICNP). Paris, France, 2002: 78-86

[9] 李喆，李冬妮，王光兴. LEO/MEO 卫星网络中运用自组网思想的动态路由算法[J]. 通信学报，2005，26(5)：50-62

(上接第 75 页)

[3] 姚钦，史仪凯，夏锐. 多目标交互式遗传算法在测试点确定问题中的应用[J]. 系统仿真学报，2006，18(6)：1469-1472

[4] Zhang Guangfan. Optimum Sensor Localization/Selection in A Diagnostic/Prognostic Architecture [D]. Georgia: Georgia Institute of Technology, January 2005: 39-40

[5] Elbeltagi E, Hegazy T, Grierson D. Comparison among five evolutionary-based optimization algorithms [J]. Advanced Engineering Informatics, 2005, 19(1): 43-53

[6] Eusuff M M, Lansey K E. Optimization of water distribution network design using shuffled frog leaping algorithm[J]. Journal of Water Resources Planning and Management, 2003, 129(3): 210-225

[7] 李兵，蒋慰孙. 混沌优化方法及其应用[J]. 控制理论与应用，1997，14(4)：613-615